Trecento : in silico network-driven identification of target combinations for combination therapy

Chua, Huey Eng

2015

Chua, H. E. (2015). Trecento : in silico network-driven identification of target combinations for combination therapy. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/64995>

<https://doi.org/10.32657/10356/64995>

**TRECENTO: *IN SILICO* NETWORK-DRIVEN
IDENTIFICATION OF TARGET COMBINATIONS FOR
COMBINATION THERAPY**

CHUA HUEY ENG

NANYANG TECHNOLOGICAL UNIVERSITY

2015

**TRECENTO: *IN SILICO* NETWORK-DRIVEN
IDENTIFICATION OF TARGET
COMBINATIONS FOR COMBINATION
THERAPY**

CHUA HUEY ENG
(*B.Eng. (Hons), NTU*)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTATIONAL AND SYSTEMS BIOLOGY (CSB)
SINGAPORE-MIT ALLIANCE
NANYANG TECHNOLOGICAL UNIVERSITY

2015

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Chua Huey Eng
20 March 2015

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, **Associate Professor Sourav S. Bhowmick**, for his continuous support and invaluable advice throughout these years. His enthusiasm and passion in doing good research has inspired me to do the same. From him, I learnt how to identify good research problem, write better technical papers, and give more interesting presentations. I would like to thank him for encouraging me to explore research that interest me and allowing me to pursue it. I am also gratified to have the opportunity to work with and learn from **Professor C. Forbes Dewey, Jr., Dr. Lisa Tucker-Kellogg** and **Professor Hanry Yu**. Their constant encouragement, guidance and advice throughout these years have made doing research so much more enriching and fulfilling.

I am grateful to friends and colleagues in the Centre for Advanced Information Systems (CAIS); the BioInformatics Research Centre (BIRC); the Computational and Systems Biology (CSB) program; the Hatsopoulos Microfluids Laboratory; and the Laboratory of Cellular and Tissue Engineering for their lunch company and intriguing conversations. In particular, my heartfelt gratitude goes out to Mr. Lai Chee Keong, Mr. Chua Chiew Song, Mr. Loo Kian Hock and Mr. Tan Sing Yau for their technical support in CAIS and BIRC. I would also like to thank the Singapore-MIT Alliance, the School of Computer Engineering and Nanyang Technological University for the fellowship and financial support.

Last but not the least, I would like to thank my family, especially my husband Jeffery, for their love, support and patience throughout these years, without which I would not have started and completed this work.

Contents

Acknowledgments	i
List of Figures	vi
List of Tables	xii
List of Publications	xvii
Abstract	xix
1 Introduction	1
1.1 Combination Therapy	2
1.2 Data-driven Combination Therapy	3
1.3 Motivation	4
1.4 Overview of Research	5
1.5 Novel Contributions	8
1.6 Organization of the Thesis	10
2 Related Work	12
2.1 The Drug Discovery and Development Pipeline	12
2.2 Research Areas Related to Phases in the Pipeline	13
2.3 Target Characterization	21
2.4 Target Prioritization	23
2.5 Target Combination Identification	27
2.6 Summary	31

3	Architecture Of TRECENTO	34
3.1	Models of Signaling Networks	34
3.2	Challenges	37
3.3	The MAPK-PI3K Signaling Network	39
3.4	The TRECENTO Architecture Framework	40
3.5	Summary	52
4	Topological Features and Signaling Networks	53
4.1	Topological Features	53
4.2	Signaling Networks	62
4.3	Curation of Signaling Network Targets	68
4.4	Summary	75
5	TENET: Structural Feature-Based Target Characterization	76
5.1	Introduction	77
5.2	Motivation	77
5.3	Problem Statement	79
5.4	Support Vector Machine SVM	82
5.5	Target Characterization	91
5.6	Experiments	107
5.7	Summary	138
6	TAPESTRY: A Novel Algorithm for Fast Discovery of Putative Target Nodes	140
6.1	Background	141
6.2	Motivation	142
6.3	Related Work	145
6.4	Unseen Network Example: Ras Activation Network	151

6.5	Problem Statement	155
6.6	Target Prioritization	157
6.7	Experiments	178
6.8	Summary	196
7	STEROID: Towards <i>In Silico</i> Target Combination Identification	197
7.1	Introduction	197
7.2	Motivation	199
7.3	Background	200
7.4	Problem Statement	202
7.5	Identifying Target Combination	207
7.6	Experiments	218
7.7	Summary	240
8	Conclusions and Future Work	242
8.1	Summary	242
8.2	Future Work	244
8.3	Conclusion	248
	References	249
	Appendix A TENET Prioritization	294
	Appendix B Structural Features: Algorithm Analysis	303
B.1	Degree Centrality	303
B.2	Eigenvector Centrality	303
B.3	Closeness and Eccentricity Centralities	304
B.4	Bridging Coefficient	304

B.5	Bridging Centrality	305
B.6	Clustering Coefficient	305
B.7	Proximity Prestige	306
B.8	Target Downstream Effect	306
Appendix C Best SVM Model Parameters		308

List of Figures

2.1	The drug discovery pipeline adapted from [304].	14
2.2	Summary of approaches for identification of target combinations.	15
2.3	The model-free and model-based approach paradigms.	17
3.1	MAPK-PI3K signaling cascade adapted from [143]. The green solid line highlights an example of a hyperedge involving 3 nodes AktPIP3, Akt and PIP3. The red broken line highlights an example of feedback loops involving ERK, ERKP, ERKPP, MEKPP and MKP3. The grey circle highlights an example of a hub node.	35
3.2	The system architecture of TRECENTO.	40
3.3	Details of some modules in TRECENTO.	43
3.4	Steps to identify target combinations in the MAPK-PI3K network. The black box represents a module in the architecture; the wavy-bottom box represents the given signaling network; the white box represents data generated by the modules and the cylinder represents a repository.	49
4.1	Directed triangle graph [107].	59
4.2	Some feedforward and feedback loop motifs commonly found in directed graphs. The coupled loops in the figure contain instances of simple loops.	64

4.3	Glucose-stimulated insulin secretion network in pancreatic β cells adapted from [168]. (A) A schematic diagram of the network. The modules are color coded: blue (glycolysis), purple (NADH shuttles), yellow (pyruvate cycle), grey (TCA cycle) and beige (respiratory chain). The steps linking glucose uptake to insulin secretion are numbered 1 to 4. (B) Table providing a mapping between the abbreviated node name and the molecule the node represents. (C-E) Biochemical reactions occurring in different modules. For ease of drawing, some molecules (marked in red) are repeated multiple times within the same module.	66
4.4	Glucose-stimulated insulin secretion network in pancreatic β cells adapted from [168]. Continuation from Figure 4.3. (F-H) Biochemical reactions occurring in different modules. For ease of drawing, some molecules (marked in red) are repeated multiple times within the same module.	67
4.5	Glucose metabolism network in <i>Escherichia coli</i> adapted from [192]. (A) Biochemical reactions of the glucose metabolism network. For ease of drawing, some molecules (marked in red) are repeated multiple times. (B) Table providing a mapping between the abbreviated node name and the molecule the node represents.	69
4.6	Glucose metabolism network in <i>Escherichia coli</i> adapted from [192]. Continuation from Figure 4.5. (C-D) Biochemical reactions related to production and degradation of the molecules.	70
4.7	The sea urchin Endo16 regulatory pathway. The nodes in this pathway are targets for regulating Endo16.	74
5.1	Support vector classification. The filled shapes lying on the hyperplanes H_1 and H_2 are the support vectors.	84

5.2	Support vector regression. The filled shapes lying on the hyperplanes H_1 and H_2 are the support vectors.	85
5.3	(a) Hypergraph (adapted from Figure 3.1), (b) Bipartite graph representation of (a), and (c) DAG representation of (b).	98
5.4	Indexed directed acyclic graph representation of Figure 3.1 with strongly connected component and non-spanning tree edge table. Solid and dotted arrows correspond to edges in the spanning and non-spanning tree, respectively	99
5.5	Performance of different kernels using the TENET-naïve approach.	117
5.6	Performance of different feature selection approaches.	118
5.7	Boxplots of performance of different feature selection approaches.	119
5.8	Runtime performance of different feature selection approaches.	121
5.9	Performance of TENET variants incorporating feature selection approach and WMC for the <code>MAPK-PI3K</code> network.	123
5.10	Performance of TENET variants incorporating feature selection approach and WMC for the glucose-stimulated insulin secretion network.	124
5.11	Performance of TENET variants incorporating feature selection approach and WMC for the endomesoderm gene regulatory network.	125
5.12	Performance of TENET variants incorporating feature selection approach and WMC for the glucose metabolism network.	126
5.13	Performance of TENET variants incorporating feature selection approach and WMC for C_1	127
5.14	Performance of TENET variants incorporating feature selection approach and WMC for C_2	128
5.15	Performance of TENET variants incorporating feature selection approach and WMC for C_3	129

5.16	Performance of TENET variants incorporating feature selection approach and WMC for C_4	130
5.17	Normalized rank of nodes in test set of I_1	131
5.18	Normalized rank of nodes in test set of I_2	131
5.19	Normalized rank of top 10 nodes ranked using TENET in test set of I_3	132
5.20	Normalized rank of nodes in test set of I_4	132
5.21	Normalized rank of targets in test set of C_1	133
5.22	Normalized rank of targets in test set of C_2	133
5.23	Normalized rank of top 15 nodes ranked using TENET in test set of C_3 . † <i>none</i> is the node used to represent degraded proteins in this network.	135
5.24	Normalized rank of targets in test set of C_4	136
5.25	Performance of different prioritization approaches on the signaling networks.	137
5.26	ROC analysis on different approaches.	138
5.27	AUPR analysis on different approaches.	138
6.1	Automorphism orbits 0 to 72 defined in [287]. Nodes belonging to the same orbit in a graphlet are in the same shade.	147
6.2	Ras activation signaling network adapted from [105]. (A) A schematic diagram of the network. The modules are color coded: blue (Golgi), green (plasma membrane), and yellow (cytoplasm). (B-E) Biochemical reactions occurring in different cellular compartments. Molecules in red participate in reactions occurring in multiple compartments.	152
6.3	Some concentration-time profiles of the $MAPK-PI3K$ network.	158
6.4	ROC AUC analysis of target ranking on the Ras activation network using the prioritization model of different training networks.	181
6.5	Summary of the ranks of I_1 to I_4 obtained using various FUSSY variants.	182

6.6	Distance between the ground truth and the ranks obtained using various FUSSY variants.	182
6.7	Runtime performance of various FUSSY variants.	183
6.8	Summary of the ranks of I_1 to I_4 obtained using various approaches.	185
6.9	Distance between the ground truth and the ranks obtained using various approaches.	185
6.10	Runtime performance of various approaches.	186
6.11	Effect of varying the weights of SFB rank and PSSD rank on final prioritization ranks for I_1 to I_4 (left) and the Ras activation network I_0 (right).	189
6.12	Normalized ranks of targets in I_0 . TAPESTRY ₀ and TAPESTRY ₃ indicates the TAPESTRY approach with ω_1 set to 0 and 0.3, respectively.	190
6.13	Performance of various approaches. TAPESTRY ₀ and TAPESTRY ₃ indicates the TAPESTRY approach with ω_1 set to 0 and 0.3, respectively.	190
6.14	Runtime performance of various approaches.	191
6.15	Effect of varying the weights of SFB rank and PSSD rank on final prioritization ranks for I_0	192
6.16	Ground truth found using AUPR.	193
6.17	Distance between the ground truth (AUPR) and the ranks obtained using various approaches.	193
6.18	ROC performance of different TAPESTRY variants using best matched model selected with different methods.	194
6.19	AUPR performance of different TAPESTRY variants using best matched model selected with different methods.	195
6.20	Boxplot of results in Figure 6.18.	195
6.21	Boxplot of results in Figure 6.19.	195

7.1	Isobologram. D_1 and D_2 are the dose of each drug that achieves the desired therapeutic effect if administered alone. d_1 and d_2 together can achieve the same effect.	202
7.2	Effect of heuristics.	223
7.3	Characterization of solution sets.	223
7.4	Percentage of STERIOD-LOEWETAPESTRY and MCSA solution sets with targets in known drug combinations.	226
7.5	Effects of solution set size.	234
7.6	Effects of initial temperature of simulated annealing.	235
7.7	Effects of limits of iterations per cycle of simulated annealing.	235
7.8	Effects of adjustment factor for therapeutic effect.	236
7.9	Solution space when adjustment factor for therapeutic effect is varied.	236
7.10	Effects of adjustment factor for target interaction.	237
7.11	Effects of selective pressure.	237
7.12	Effects of combination size.	237

List of Tables

4.1	Description of symbols used in structural feature-based target characterization.	54
4.2	Summary of the four signaling networks. [†] The output node in round bracket corresponds to the node name used in the model. FBL and FFL denote feedback and feedforward loops, respectively. Figure 4.2 illustrates some examples of FBL and FFL in the network. Note that within coupled FBL and FFL, there may be multiple instances of simple FBL and FFL. Instances of such simple loops are not reflected in our count.	63
4.3	Ovarian cancer drugs in [259] and their targets in the <code>MAPK-PI3K</code> network [143].	71
4.4	T2DM drugs or food constituents in [259] and their targets in the glucose-stimulated <code>insulin</code> secretion network [168].	73
4.5	Targets that are crucial for acetate production in the glucose metabolism network in [192]. * indicates targets that are included by default due to their direct involvement (either as input or output) in the metabolic reaction being studied.	75
5.1	SVM kernel types and their associated parameters [238].	85
5.2	Wilcoxon test of the signaling networks. Each cell entry contains the p -value in the form of $xxx(yyy)$ where xxx and yyy denote the p -values for the alternative hypotheses where the features of the known target nodes are smaller and greater than the non-target nodes, respectively. Note that $p(A)$ and $p(T)$ denote feature p when applied to the entire node set and the training node set, respectively. . .	93

5.3	Area under the ROC curves of the signaling networks. Note that $p(A)$ and $p(T)$ denote feature p when applied to the entire node set and the training node set, respectively.	95
5.4	Time and space complexity for computing the different features. The proofs of those algorithm complexities that are provided without citation are reported in Appendix B.	103
5.5	Summary of data set. The network IDs are used for identifying the networks in the figures in Section 5.6. The test sets are listed in Table 5.6.	109
5.6	Test set of I_1 to I_4 . Nodes marked with $^+$ are known targets.	109
5.7	SVM kernel types and their associated parameters [238].	110
5.8	The different variants of TENET studied. $\surd$ indicates the SVM-training approach(es) incorporated into the different variants. For instance, TENET-WB incorporates the BSE feature selection approach and the weighted misclassification cost.	111
5.9	Weights ratio used in the experiments.	111
5.10	Wilcoxon test results of the networks. Each cell entry contains the p -value in the form of xxx(yyy) where xxx and yyy denote the p -values for the alternative hypotheses where the features of the known target nodes are smaller and greater than the non-target nodes, respectively.	114
5.11	ROC AUC of the networks.	115
5.12	Summary of structural features selected for different signaling networks using different approaches. $\surd$ indicates the feature selected for the approach. For instance, in I_4 , TENET-R selected σ	120
5.13	Summary of best TENET variant for different networks. The C^+ values are provided in bracket besides approaches which uses WMC.	134
6.1	Features extracted by NETSIMILE and their notations.	148

6.2	Ovarian cancer drugs in [259] and their targets in the Ras- activation network [105].	154
6.3	Summary of the various network similarity approaches and variants studied.	163
6.4	Summary of the LEARNPTHRESHOLD procedure. The $(i,j)^{th}$ cell entry is of the form $x[y]$ where x indicates the combined p -value for the (I_i, I_j) pair and y indicates the ROC AUC when prioritization model of I_j is applied to I_i . † mark entries with ROC AUC < 0.5.	184
6.5	Ranking based on inverse of the DTW and Euclidean distances with respect to $ERKPP$, denoted as $D_{DTW}(\cdot)$ and $D_{Eucl}(\cdot)$, respectively. Table is read from top to bottom and from left to right.	187
6.6	Weights ratio used in the experiments.	188
6.7	Weight ϖ_1 of training networks that yields prioritization results having ROC AUC greater than or equals to 0.8.	188
6.8	Properties of prioritization models. Refer to Table 4.1 for descriptions of the symbols of the features.	192
7.1	Description of symbols used in this chapter.	203
7.2	Details of individual drug targets in the $MAPK-PI3K$ network achieving the desired therapeutic effect. Each reversible reaction is converted into a pair of equivalent irreversible reactions where f and b indicate the forward and the backward reactions in the irreversible reaction pair, respectively. Table is read from top to bottom and from left to right.	213
7.3	The different approaches studied. $\checkmark$ indicates the heuristic used by each approach. For instance, STEROID-LOEWETAPESTRY uses LOEWE and TAPESTRY prioritization-based heuristics.	220
7.4	<i>PubMed</i> results relevant to ovarian cancer drug combinations targeting the $MAPK-PI3K$ network.	224

7.5	Mapping between targets in Table 7.4 and MAPK-PI3K network.	224
7.6	Known combinations in solution sets.	225
7.7	Top-10 target combinations for MAPK-PI3K network found using STERIOD- LOEWETAPESTRY. Note that targets present in Table 7.5 are marked with *. . .	227
7.8	Top-10 target combinations for MAPK-PI3K network found using MCSA. Note that targets present in Table 7.5 are marked with *.	228
7.9	Bottom-10 target combinations for MAPK-PI3K network found using STERIOD- LOEWETAPESTRY. Note that targets present in Table 7.5 are marked with *. . .	229
7.10	Bottom-10 target combinations for MAPK-PI3K network found using MCSA. Note that targets present in Table 7.5 are marked with *.	230
7.11	Target combinations for glucose-stimulated insulin secretion network found using STERIOD-LOEWETAPESTRY.	232
7.12	Top-5 results of STERIOD-LOEWETAPESTRY for $\mathcal{S}=\{2,3,4\}$	238
7.13	Top-5 results of STERIOD-LOEWETAPESTRY for $\mathcal{S}=5$	239
7.14	Top-5 results of STERIOD-LOEWETAPESTRY for $\mathcal{S}=6$	240
A.1	Rank of nodes in test set of C_1	294
A.2	Rank of nodes in test set of C_2	295
A.4	Rank of nodes in test set of C_4	301
A.5	Rank of nodes in test set of I_1	302
C.1	Best model parameters for the various signaling networks using the TREN- CENTO-naïve approach with different kernels. The parameters for the TREN- CENTO-naïve approach with linear kernel is found in Table C.2.	309
C.2	Best model C parameter for the various signaling networks using various ap- proaches with linear kernel.	310
C.3	Best model C parameter for the various combined signaling network variants (C_1 to C_4) using different approaches with linear kernel.	311

C.4	Best model C parameter and associated model evaluation criteria values for the combined network (formed from all learning networks) trained with a linear kernel.	312
-----	---	-----

List of Publications

- **H.E. Chua**, S. S Bhowmick, and L. Tucker-Kellogg. One Feature Doesn't Fit All: Characterizing Topological Features of Targets in Signaling Networks. In *BCB'14: Proceeding of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2014.
- S. S Bhowmick, **H.E. Chua**, and J. Zheng. INGOT: Towards Network-driven *In Silico* Combination Therapy. In *BigComp'14: Proceeding of the 1st International Conference on Big Data and Smart Computing*, 2014.
- **H.E Chua**, S. S Bhowmick, L. Tucker-Kellogg, and C.F. Dewey Jr. STERIOD: *In Silico* Heuristic Target Combination Identification for Disease-related Signaling Networks. In *BCB'12: Proceeding of the 3rd ACM conference on Bioinformatics, Computational Biology, and Health Informatics*, 2012.
- **H.E Chua**, S. S Bhowmick, L. Tucker-Kellogg, Q. Zhao, C.F. Dewey Jr. and H. Yu. *In Silico* Identification of Endo16 Regulators in the Sea Urchin Endomesoderm Gene Regulatory Network. In *IHI'12: Proceeding of the 2nd ACM SIGHIT International Health Informatics Symposium*, 2012.
- **H.E. Chua**, S. S Bhowmick, L. Tucker-Kellogg, Y. Wang, C.F. Dewey Jr. and H. Yu. PANI: An Interactive Data-driven Tool for Target Prioritization in Signaling Networks. In *IHI'12: Proceeding of the 2nd ACM SIGHIT International Health Informatics Symposium*, 2012.

- **H.E. Chua**, Q. Zhao, S. S Bhowmick, C.F. Dewey Jr., L. Tucker-Kellogg, and H. Yu. PANI: A Novel Algorithm for Fast Discovery of Putative Target Nodes in Signaling Networks. In *BCB'11: Proceeding of the 2nd ACM conference on Bioinformatics, Computational Biology, and Health Informatics*, 2011.
- **H.E. Chua**, S. S Bhowmick, and , L. Tucker-Kellogg. TENET: Structural Feature-based Target Characterization for Signaling Networks. *To be submitted for publication.*
- **H.E. Chua**, S. S Bhowmick, and , L. Tucker-Kellogg. TRISTAN: Data-driven Target Prioritization in Signaling Networks. *To be submitted for publication.*

Abstract

In recent years, there has been a gradual paradigm shift in the drug industry towards *combination therapy* after the “one-target one-drug” (mono-therapy) approach fails to increase the number of successful drugs. Although, the use of multiple drugs in a therapy is potentially useful for addressing diseases (*e.g.*, cancer) that implicate multiple genes and pathways, there is also a higher possibility of drug toxicity that is caused by various factors including the compounding of *off-target effects* of individual drugs. Careful selection of target combination is an important step towards reducing attrition rate of new drug combinations during preclinical and clinical phases. Although various techniques have been suggested for target selection, they generally apply to mono-therapies. Many of these approaches (*e.g.*, affinity matrix) cannot be applied directly to target combination identification because they lack consideration of interaction between targets that affects efficacy and toxicity profiles of drug combinations hitting these targets. In addition, the few proposed approach for target combination identification all suffer from certain key limitations. First, they assume that all targets are equally probable for a combination. However, targets influence effects of the combinations differently and appropriate choice of targets can potentially improve the target combinations. Second, they demand specific technical expertise from users (*e.g.*, knowledge of what off-target effects to optimize) and a lack of the required expertise can undermine the effectiveness of the method.

In this dissertation, we seek to address these limitations by proposing a framework called TRECENTO (*In Silico* NeTwork-dRiven IdEntifiCation of TargEt CombiNaTions for COmbination Therapy). TRECENTO uses a network-based approach for identifying target combinations in

signaling networks. In particular, it addresses three aspects of target combination identification, namely, *target characterization*, *target prioritization* and *synergistic target combination identification*. For target characterization, we propose TENET, a support vector machine-based approach, to characterize known targets in signaling networks using network topological features. The insights gained from characterizing the targets are used subsequently for target prioritization in our proposed network-based approach, TAPESTRY. Finally, for synergistic target combination identification, we propose a heuristics-based simulated annealing algorithm called STERIOD. Specifically, STERIOD uses heuristics based on target prioritization and the *Loewe additivity theorem* to identify synergistic target combinations that are optimized for off-target effects. We evaluated TRECENTO on five signaling networks that are implicated in diseases (*e.g.*, cancer) or specific biological processes (*e.g.*, glucose metabolism). Our experimental results reveal several key findings. First, known targets in different signaling networks are characterized by different sets of topological features. This is different from the observation reported by Hwang and colleagues for several protein-protein interaction networks. Second, the use of network dynamic features can improve target prioritization in certain networks. This finding highlights the importance of dynamic features and the need to explore them further in network-based analysis of signaling networks. Third, using appropriate heuristics can improve the quality of target combinations identified in terms of off-target effects when compared to state-of-the-art approaches. Note that TRECENTO optimizes off-target effects of the entire system instead of specific off-target effects. In summary, this thesis is an important first step towards improving the quality of target combination selection, a key step in enabling target combination therapies.

Chapter 1

Introduction

Cells use sophisticated communication between proteins in order to perform a variety of functions such as growth, survival, proliferation and development. As signaling proteins rarely operate in isolation through linear pathways, cell signaling can be viewed as a large and complex network. Specifically, the network view emerges due to ‘cross-talks’ between signaling pathways. Such a network contains numerous features such as feedback and feedforward loops [8], which render it virtually impossible to manually comprehend how signals are integrated in these pathways. Understanding signal flow in the network is paramount as alterations of cellular signaling events, such as those that arise by gene mutations or epigenetic changes can result in various diseases. For example, alterations to the genes that encode key signaling proteins, such as *RAS* and *PI3K*, are commonly observed in many types of cancers. In fact, although we have medically prevailed over numerous diseases that have plagued humanity throughout the ages, we are still not able to provide effective cures for diseases of greater complexity such as diabetes, cancer, heart disease and Alzheimer’s disease.

Therapeutic drug discovery that can target these altered signaling pathways to restore the physiological state of a disease network to normalcy has long been dominated by the “one-target one-drug” paradigm (identify a single chemical entity that binds to a single target) in the past decades. However, most complex diseased states are *polygenic* and are characterized by a combination of interacting genes and their products instead of a single gene. Although some

degree of efficacy is possible using such a paradigm, it is well-known that such a paradigm did not yield an increasing number of successful drugs as expected [116] because multiple targets have to be involved in disease control due to redundancy and multi-functionality of biological processes. For example, cell proliferation leverages the combined control of multiple growth factor receptor pathways, and genetic experiments reveal that inhibition of any single receptor is only partially effective in blocking growth. Furthermore, clinical treatment of this paradigm may give rise to unexpected *off-target effects* due to cross-talks between pathways in the disease network.

1.1 Combination Therapy

To address the limitations of drugs based on single targets, increasing attention has been shifted to *combination therapy* by targeting multiple molecules simultaneously in a disease-related signaling network [244]. Specifically, in this therapy, instead of a single compound interacting with a single target, a concerted pharmacological intervention of several compounds interacting with multiple targets is made. Such a strategy has the potential to yield better benefits compared to a single molecule (referred to as mono-therapy) for complex diseases due to its ability to use lesser dosage to achieve efficacy, reduction of the frequency at which acquired drug resistance arises by combining drugs with minimal cross-resistance, and enhancing potency (amount required to produce an effect of given intensity) by leveraging *additivity* or *synergism* in the biochemical activities of multiple drugs. Examples of such a strategy can be found in the combination therapy of AIDS, cancer, and hypercholesterolaemia [244]. Even for diseases that are caused entirely by disruption of a single pathway, combination therapy might still offer benefits over mono-therapy by virtue of spreading out the side effects to sub-toxic levels, while concentrating the desired effects on the target pathway. However, not all combination therapies produce better effects than mono-therapies. For instance, in a study of combinations of analgesic drugs, some combinations (*e.g.*, aspirin and pentazocine) were beneficial, while

others (*e.g.*, acetaminophen and pentazocine) were detrimental [216]. Hence, it is important to formulate strategies to develop good drug combinations which maximize the overall *therapeutic effect* while minimizing the *off-target effects*.

The traditional approach for combination therapy is generally based on designing combinations based on clinical experience of doctors, knowledge of biological mechanisms, and practical constraints in the design of clinical trials [379]. Most drugs were initially developed as effective single agents and only later combined clinically. A common assumption in this case is that only drugs that are effective individually should be used as part of a drug combination. Because the effect of drugs depends on the dose, several doses need to be studied for the drug combinations, and the number of possible combinations can grow exponentially. For instance, a cancer chemotherapy regimen typically consists of six or more drugs from more than 100 anticancer drugs. However, investigating all six combinations out of 100 (including partial combinations) at three different doses to determine which combination is effective generates 8.9×10^{11} possibilities. As clinical trials are very expensive, exhaustive study of all possible combinations clearly becomes intractable. Although high-throughput screening technology allows the testing of pairs of drugs over a range of doses, combinatorial explosion still prevents exhaustive measurement of combinations of more than two drugs [410]. Hence, alternative procedures are needed to enable the rapid search for superior drug combinations targeting disease-related networks.

1.2 Data-driven Combination Therapy

With the fast accumulation of experimental data from high-throughput screening and omic data measurements along with growing availability of disease networks, data-driven tools that can facilitate early detection of efficacious and non-toxic drug combinations *in silico* can serve as a powerful discovery and pre-screening platform when coupled with other complementary technologies such as high-throughput screening. Several groups have recently undertaken a data-

driven *model-based* approach [11, 46, 144] where biological measurements (either obtained by the authors or derived from the literature) are used to build explicit models of the target biological network for the optimization of drug combinations using simulations. For example, Yang et al. [392] modeled the arachidonic acid metabolic network, related to inflammation, with a 2-step simulated annealing approach to optimize (1) drug candidates and (2) level of intervention for multiple targets. On the other hand, advocates of *model-free* approach [47, 161, 386] have described biological search algorithms, where the search is not conducted *in silico* but directly using biological measurements from *in vitro* or *in vivo* systems. These methods can be stochastic, for example using Monte Carlo or evolutionary algorithms, or non-stochastic. For instance, Calzolari et al. [47] used non-stochastic algorithms derived from one used in digital decoding to perform biological searches for drug combinations, both *in vivo* (using fruit flies) and *in vitro* (using cell lines).

1.3 Motivation

The model-based and model-free approaches, however, suffer from two key limitations. First, they ignore off-target effects of the combinations. Consequently, these approaches may yield combinations satisfying the user-desired therapeutic effect, but with excessive off-target effects, rendering the combinations useless due to their toxicity. Note that in [392] a user needs to specify *a priori* specific side effects (as input to the algorithm) in terms of the ratio of concentration of two relevant nodes. Due to the complexity of biological networks, such a strategy is often impractical as it is highly unlikely for a user to know all *system-wide* side effects ahead of time. Second, although the target activity affects the combination effects, it is chosen randomly without considering *synergistic*¹ combinations. Synergistic drug combinations use lower doses

¹A joint action of two drugs in such a manner that one supplements or enhances the action of the other to produce an effect greater than that which may be obtained with either one of the drugs in equivalent quantity or produce effects that could not be obtained with any safe quantity of either drug, or both.

of the drug constituents to achieve the same therapeutic effect. Hence, it has often been proposed as a strategy for reducing adverse side effects [348]. A judicious selection process that is “synergism-aware” can provide us an opportunity not only to select superior drug combinations but also improve efficacy of the overall process by ignoring non-synergistic combinations.

The process of identifying drug combinations essentially consists of two key steps, namely identification of good target combinations and identification of appropriate set of drugs hitting these targets. In this thesis, we focus on the former step and propose a novel *in silico* framework for identifying target combinations called TRECENTO² (*In Silico NeTwork-dRiven IdEntifiCation of TargEt CombiNaTions for COmbination Therapy*) for combination therapy. In particular, TRECENTO addresses the problem of *target combination identification*. Informally, given a signaling network and a desired therapeutic effect, *target combination identification* finds suitable sets of targets and the required *target activities* (type and extent of perturbations) of these targets that achieve the desired therapeutic effect and have minimized off-target effects. Note that the goal of this thesis is to identify synergistic combinations of targets with reduced off-target effects and excludes the evaluation of drug compounds that bind and regulate the target molecules.

1.4 Overview of Research

In this thesis, we propose a framework called TRECENTO which identifies target combinations rationally from the systems biology perspective. Note that TRECENTO is different from approaches that are based on genomics and proteomics alone (reviewed in Chapter 2), which typically identify target combinations using phenotypic screenings without necessarily understanding the role of the targets in the disease. Recall that these genomics and proteomics methods generally treat the biological system as a black box and identify targets by correlating genomic or proteomic experiments with desired phenotypes. For instance, multiple gene

²The *Trecento* refers to the 14th century Italian culture and is commonly regarded as the beginning of the Renaissance in art history [19].

knockdowns experiments are performed and phenotypes are observed to identify target genes that can induce the desired phenotypes. This lack of understanding of the role of targets in the disease hinders toxicology study in preclinical and clinical phases [193]. For example, what biomarkers are appropriate for accessing toxicity? In particular, TRECENTO employs a network-based approach that facilitates identification of rational targets. That is, it considers the biological system as a network consisting of different molecules interacting together. The following are three key steps in TRECENTO.

- (i) *Target characterization* step (Chapter 5): In this step, known targets of signaling network are characterized using structural features (referred to as *predictive structural features*) and *prioritization models* based on these predictive structural features are generated. Note that in the subsequent step (*target prioritization*), targets of the given signaling network are usually not known. In our proposed target characterization step, signaling networks whose targets are known are used to learn predictive structural characteristics that can identify known targets from non-targets. These predictive structural features shall be used to construct prioritization models that can be used to identify and prioritize targets of future unseen signaling networks in the subsequent step. Specifically, we proposed TENET, a support vector machine (SVM)-based approach to identify predictive structural features (*e.g.*, bridging centrality and degree centrality) and learn prioritization models that best characterize known targets in signaling networks. Note that TENET rejects the assumption of existing state-of-the-art approaches that targets in all signaling networks share common characteristics in terms of topological features (*e.g.*, *bridging centrality* [160]).
- (ii) *Target prioritization* step (Chapter 6): In this step, *target prioritization* is performed by ranking targets according to a *putative target score* that is derived from structural and dynamic features of targets. We proposed TAPESTRY, a network-based target prioritization

approach. First, TAPESTRY identifies an appropriate prioritization model (generated by TENET) for a given network whose targets are unknown. The prioritization model is used to generate *structural feature-based ranks* for candidate targets in the given network. Then, *dynamic feature-based ranks* are derived from *profile shape similarity distance*, our proposed dynamic feature. Finally, the *structural feature-based ranks* are *integrated* with the *dynamic feature-based ranks* to obtain a putative target score for prioritizing the targets. In our experiments, TAPESTRY-prioritized targets are found to correlate well with a set of curated targets. Unlike existing approaches, TAPESTRY filters off nodes that are unlikely targets and considers both structural and dynamic features for target prioritization.

- (iii) *Target combination identification step* (Chapter 7): In this step, synergistic target combinations based on targets prioritized in the previous step are identified. In order to achieve this, we studied the use of heuristics such as target prioritization and target synergy to identify efficacious target combinations with reduced off-target effects. We modeled the *target combination identification* problem as a *constraint satisfaction problem* and proposed STERIOD to identify synergistic target combinations. Unlike state-of-the-art approaches, STERIOD identifies target combinations that are synergistic and optimized for off-target effects. In particular, STERIOD uses simulated annealing (SA) and two heuristics for target prioritization. The *target prioritization-based heuristic* is used to select for appropriate targets for the combination whereas the *LOEWE-based heuristic* is used to select for appropriate activities of these targets.

We evaluated TRECENTO on several real-world biological signaling networks implicated in human diseases or other biological processes. Specifically these networks are:

- (i) The MAPK-PI3K network [143] that is implicated in ovarian cancer.

- (ii) The glucose-stimulated insulin secretion network [168] that is implicated in type II diabetes mellitus.
- (iii) The Ras activation network [105] that is implicated in ovarian cancer.
- (iv) The glucose metabolism network [192] that describes glucose metabolism.
- (v) The endomesoderm gene regulatory network [196] that describes embryonic development.

1.5 Novel Contributions

We shall now discuss several novel contributions of TRECENTO, our proposed network-based framework for identifying synergistic target combinations.

- **A systematic framework to perform network-based characterization of targets.** Target characterization using network features (*e.g.*, degree centrality) provides insights to targets that can be used to guide the search of new targets. For instance, identification of potential targets sharing similar network characteristics as known targets. We propose TENET, a SVM-based approach for characterizing known targets in signaling networks using structural features. Unlike existing work [191, 206, 233] that performs cursory network-based target characterization studies using statistical methods and few (generally about four) features, TENET provides a framework to facilitate the study of a comprehensive set³ of topological features. In particular, TENET identifies a set of structural features that characterizes known targets and uses these features to generate a prioritization model. The prioritization model captures the network structural characteristics of known targets and can be used to facilitate network-based target identification and prioritization. To the best of our knowledge, this is the first supervised machine

³In this thesis, we studied a total of 16 structural features using TENET. The SVM-based framework that TENET uses is flexible and can accommodate a large number of features without modification to the framework

learning-based approach for characterizing targets in signaling networks using structural features.

- **A novel approach to prioritize targets.** We propose TAPESTRY, a novel network-based target prioritization approach that generates a score using structural and dynamic features for ranking targets. Unlike existing network-based target prioritization approaches [160, 171] that consider only structural features, TAPESTRY considers both structural and dynamic features. Instead of using the same network features (*e.g.*, bridging centrality in [160]) to prioritize targets for all networks, TAPESTRY identifies appropriate features for the given network by using *network similarity ranking* techniques to select appropriate prioritization models (generated by TENET) that are most likely to characterize targets in the given network. Although TAPESTRY uses multiple features for prioritizing targets, it does not require users to specify which features to use or what weights to use for the features as these are learnt using examples of signaling networks with known targets. Hence, compared to state-of-the-art approaches (*e.g.*, *NetworkPrioritizer* [171]), TAPESTRY facilitates network-based prioritization using multiple features without imposing additional burdens on the users to choose appropriate features or weights for target prioritization.
- **A rational approach to identify synergistic target combinations *in silico*.** TRECENTO uses three key steps to perform rational identification of synergistic target combinations. Specifically, it performs structural feature-based target characterization using TENET, followed by target prioritization using TAPESTRY and finally identification of target combinations using STERIOD. In particular, TAPESTRY uses the prioritization models generated by TENET for prioritizing targets and STERIOD uses prioritization ranks generated by TAPESTRY to guide the selection of candidate target combinations. Unlike

existing state-of-the-art *in silico* approaches [161, 392], TRECENTO leverages network-based analysis to guide selection of candidate target combinations, resulting in potentially safer and more efficacious target combinations. In particular, STERIOD employs heuristics based on target prioritization and target synergism to identify synergistic target combinations that are optimized for off-target effects.

- **Curation of Targets.** A key issue in target prioritization is the validation of targets that are identified and prioritized by TAPESTRY. In this thesis, we undertake manual target curation of five signaling networks that we study in order to validate the target prioritization results of TAPESTRY. Chapter 4 documents the curation process and [66] reports details of curation targets which can serve as a benchmark for validating new target prioritization approaches. Similar curation approach utilizing the *ClinicalTrials.gov* [259] and the *DrugBank* [385] repositories have been reported in [390] for purpose of drug repositioning.

Note that unlike STERIOD which is targeted towards combination therapy, TENET and TAPESTRY are general target characterization and prioritization algorithms that can be applied to different biological problems, including characterizing and prioritizing regulators of a particular gene (basic research) or drug targets of a particular disease (industrial application).

1.6 Organization of the Thesis

The rest of this thesis is organized as follows.

- In Chapter 2, we introduce the discovery and development pipeline for combination therapies and discuss representative work related to different stages of the pipeline. In particular, we focus on work related to *in silico* target characterization, target prioritization and identification of target combinations.

- In Chapter 3, we present the architecture of the TRECENTO framework and discuss the roles of the components in the framework.
- In Chapter 4, we introduce the structural features and signaling networks that we shall use for structural feature-based target characterization (Chapter 5).
- In Chapter 5, we present our work on TENET to solve the *target characterization problem*. In particular, TENET learns a *prioritization model* based on topological features, which can be used subsequently to *prioritize* targets for signaling networks.
- In Chapter 6, we present our work on TAPESTRY which uses a network-based approach to solve the *target prioritization problem*.
- In Chapter 7, we present our work on STERIOD to solve the *target combination identification problem*. In particular, STERIOD uses *target prioritization-based heuristic* to select for appropriate targets for the combination and *LOEWE-based heuristic* to select ensure that selected targets act synergistically in the combination. As we shall see later in Chapter 7, these heuristics led to more efficient and effective target combination solutions.
- In Chapter 8, we discuss limitations of the TRECENTO framework and future work.

Chapter 2

Related Work

In this chapter, we present an overview of the representative work in target combination research. First, we introduce the drug discovery and development pipeline in Section 2.1. Then, in Section 2.2, we briefly discuss key research areas and exemplary work related to it. Finally, we focus on work related to the discovery phase in the pipeline. Specifically, we shall review recent work in target characterization (Section 2.3), target prioritization (Section 2.4) and target combination identification (Section 2.5). Note that emphasis shall be placed on *in silico* techniques that are most related to this thesis. Interested readers may refer to [80] for a comprehensive review of network-based approaches for drug discovery.

2.1 The Drug Discovery and Development Pipeline

In this section, we briefly describe the various stages in the *target-based* drug discovery and development pipeline. Note that in *target-based* drug research, biologically-relevant targets are first identified. Then, compounds (drugs) acting on these targets are found or synthesized. This is different from *cell-based* (or *phenotype-based*) drug research where drugs are found by performing phenotypic screening and the mechanism of action of drugs created may not be fully understood [78]. Compared to target-based approach, the preclinical and clinical phases for the cell-based approach are technically more challenging because lack of knowledge of targets

hinders mechanism-based toxicity studies [193]. Figure 2.1 illustrates the pipeline which consists of three phases, namely, the *discovery phase*, the *preclinical phase* and the *clinical phase*. Note that this pipeline can be generalized to both mono-therapies and combination therapies. The term “target” in Figure 2.1 is used in a general sense and shall imply single target in the context of a mono-therapy and target combination in the context of a combination therapy. This pipeline typically spans 15 years for a mono-therapy drug [304] and may be longer for a combination therapy drug.

In the *discovery phase*, *targets* are first identified, screened and validated. A *target* is typically an endogenous molecule such as a protein, a gene or a nucleic acid sequence that affects the outcome of a disease or a medical condition. For example, proteins, such as *P TEN* and Akt, on the *P I 3 K* signaling pathway, implicated in cell growth and survival, have been suggested as potential targets for cancer [76]. In addition, in pathogen-related diseases, the target can sometimes be endogenous to the pathogen, instead of the host. For instance, fungal cell wall that is unique to the pathogen, is an attractive target for treating fungal infections [87]. Note that in this thesis, our focus is on non-pathogen-related diseases. Then, *lead compounds* for the validated targets are identified, screened and optimized. *Lead compounds* are chemical compounds that demonstrate positive pharmacological activities that are in line with the desired therapeutic effect and are most likely to be successful in preclinical and clinical trials. They are typically selected from a group of compounds (referred to as *hits*) that display desirable pharmacological activities. Next, in the *preclinical phase*, optimized lead compounds are tested in animal models to establish drug safety and efficacy before proceeding to further testing in human subjects in the *clinical phase*.

2.2 Research Areas Related to Phases in the Pipeline

The drug discovery and development process is highly complex and involves various research topics in each phase. In this section, we shall briefly discuss key research areas in each phase.

Figure 2.1: The drug discovery pipeline adapted from [304].

2.2.1 Discovery Phase

The primary goals of the discovery phase are to identify targets that play specific roles in diseases and that are likely to be *druggable*, and to design drugs for these targets. A target is *druggable* if it can be modulated with a drug [268].

Target Identification. Target identification, the first step in target-based drug research, is particularly challenging in combination therapies because multiple targets assert effects that are often non-linear and interaction between these targets can exacerbate toxic effects of particular target combinations. It is important for the methods proposed for identifying target combinations to take these factors into considerations. In particular, methods for identifying target combinations can be broadly classified into two categories of approaches (Figure 2.2), namely, *model-free* [75, 323, 365, 372] and *model-centric* [101, 230, 368] approaches. *Model-free* approaches generally conduct target combination search *in vivo* using animal models or *in vitro* using cell-based assays. *Model-centric* approaches, on the other hand, perform target combinations identification *in silico* using network-based analysis.

The model-free approaches typically use *targeted phenotypic screening* coupled with biological search algorithms (*e.g.*, sequential decoding algorithm [47]) or *non-targeted phenotypic screening* coupled with chemical proteomics and genetics analysis. Note that different phenotypes are produced by either drug intervention or using genetic approaches such as gene knockdown [256]. Recall that molecular targets of drugs are not always known [78]. Hence, compared to drug intervention, genetic-based approach can be considered a *targeted* approach

Figure 2.2: Summary of approaches for identification of target combinations.

since it involves specific perturbation on particular genes and proteins. For instance, [204] is an example of an *in vivo* targeted phenotypic screening. Specifically, in [204], Leong et al. investigated the use of RNA interference (RNAi) to perturb two genes (*zerg2* and *zerg3*) in zebrafish that are implicated in Long QT syndrome, a human cardiac disorder. Although the RNAi technique is found to be functional in the zebrafish model, there are concerns about toxicity [204,407]. For example, siRNA-injected zebrafish embryos suffered from developmental defects such as loss of midbrain-hindbrain boundary [407]. Hence, targeted phenotypic screening approaches are still limited *in vivo* and *in vitro* especially when multiple genes have to be perturbed simultaneously [126]. Thus, most model-free approaches continue to use drugs to induce phenotypic changes. The drug treatments can be performed *in vivo* using small animal models such as zebrafish [411], or *in vitro* using high-throughput screening conducted on cell-based assays [36,383]. Chemical proteomics (such as affinity chromatography¹ [197,322] and activity-based protein profiling² [18,90]) and genetics analysis (such as mRNA display³ [235]) have to be performed following the drug-based phenotypic screenings [203] in order to identify potential target combinations. *In vivo* phenotypic screening has been suggested as a better

¹Affinity chromatography uses a solid support or tagging group such as biotin to immobilize molecules and isolate them with techniques such as chromatography and gel electrophoresis [203].

²Activity-based protein profiling uses activity-based probes which are small molecule tools to monitor the activity of specific classes of enzymes [203].

³mRNA display uses an *in vitro* translation system to produce mRNA-protein fusions that are used to probe molecules identified from affinity screens [203].

screening approach as compounds found using *in vitro* screening typically fail in preclinical and clinical settings. This is because biological processes in *in vitro* assays tend to be poor substitutes for actual processes happening in whole organisms [411]. However, various issues make *in vivo* phenotypic screening challenging. It is difficult to control gene expression and protein activity *in vivo* [126]. Moreover, animal models used for *in vivo* screening may behave quite differently from human beings.

In addition, the model-free approaches all suffer from certain key limitations. First, they treat the biological system as a black box and focus on which drugs or molecular perturbations resulted in a particular phenotype. The roles of targets within the system are ignored. For instance, what other molecules do these targets interact with? Where are these targets located in the network? Viewing the biological systems as a network of interacting proteins provide valuable insights that can be used to guide the selection of targets in the combinations [79, 80, 141]. For example, network-based analysis can be used to characterize known targets and the insights gained be used to prioritize potential targets that are likely to yield efficacious target combinations. Figure 2.3 illustrates the black-box view of model-free approaches and network-view of model-based approaches. Second, genomic-based and proteomic-based approaches focus on identifying genes and proteins that are differentially expressed, translated or modified in the disease phenotypes [352]. However, molecules that are not altered at the genomic or proteomic level, may still play essential roles in diseases [273]. Third, there are still substantial difficulties in establishing functional interactions between genes and drugs on a global scale [43, 349].

Model-centric approaches generally consist of two categories, namely, *optimization-based* and *association-based* techniques. *Optimization-based* model-centric approaches [337, 392] use optimization algorithms to search for target combinations in a graph-based computational model of the biological network. *Association-based* model-centric approaches [247, 396], on the other hand, build bipartite networks involving targets, drugs and side-effects. These approaches leverage promiscuity of drugs to identify novel targets or reposition existing drugs

Figure 2.3: The model-free and model-based approach paradigms.

[396]. For instance, a drug that is indicated for cancer but shares targets with a drug indicated for another disease is potentially efficacious for that disease as well. Unlike association-based techniques that can at most predict qualitative side-effects of particular target combinations, optimization-based approaches can quantify the potential side-effects by performing simulation on the computational models. In addition, the use of computer simulated models presents an opportunity to investigate human response to particular target combinations at an early stage with the use of computational models that mimick the human physiological systems [35, 138]. Although still in its early days, these computational models have proven to be useful in areas such as toxicology studies [359]. Note that we shall discuss representative model-centric approaches further in Section 2.5.

Drug Design. A key consideration when designing drug combination is whether to formulate the drug as a *fixed-dose* combination (FDC) or as a *free-drug* combination. A FDC is a single pill (referred to as *polypill*) that contains all components of the drug combinations whereas a *free-drug* combination consists of multiple pills where each pill contains the active ingredient of one drug [137]. Several strategies have been proposed for FDC development. In particular, crossover pharmacokinetic studies are used for comparing FDC and free-drug combinations to determine the effectiveness of using FDC [245]. Although FDC offers convenience to patients and improves adherence to treatment regimes [89], there are technical challenges that prevent such formulation in certain drug combinations. For instance, the active ingredients may not be

physically and chemically compatible or the active ingredients when combined as a single pill, may generate new impurities.

In addition, drug delivery, another crucial aspect of drug design, is dependent on the intended site of action and characteristics of drugs. For instance, there are unique considerations (*e.g.*, overcoming the blood-brain barrier and blood-cerebrospinal fluid barrier) when delivering drugs to the central nervous system. Likewise, the administration of hydrophobic (poorly soluble) drugs can be particularly challenging since poor solubility affects their bioavailability and effectiveness [298].

Furthermore, there is a need to ensure that multiple drugs in the combination therapies reach their intended site of action. Several drug delivery strategies including the use of liposomes, microspheres and prodrugs [175, 354] have been proposed. In particular, the use of nanoparticles has received much attention.

2.2.2 Preclinical Phase

In the preclinical phase, the primary goal is to determine the safety profile of the drug by performing animal studies. Various research activities take place during this phase, including the selection of appropriate animal models and identification of relevant biomarkers.

Development of Humanized Animal Models. Differences in animal models and human beings are often cited as a key reason for failure of drugs in clinical trials following success in preclinical studies [411]. Hence, there is a push towards research of animal models that can better mimick physiological processes in human. Development of humanized animal models is a relatively new field that seeks to close the gap between animal models and human. Advancement in this field is greatly enabled by development of severe combined immune deficiency (*scid*) mice and genomic techniques such as gene knockout [40]. Several groups have demonstrated the usefulness of humanized mouse models. For instance, Hofer and colleagues created

a pre-clinical model of `CCR5` knockout in human hematopoietic stem cells by zinc finger nucleases using immunodeficient mice for investigating novel anti-HIV treatments [151]. In another example, `PML-RAR $\alpha$` transgenic mice was found to respond to all-trans retinoic acid (`ATRA`) treatment similarly to human acute promyelocytic leukemia (APL) patients carrying the `PML-RAR $\alpha$` translocation [295]. Rego et al. then use the same mouse model to discover a combination therapy involving `ATRA` and arsenic trioxide (`As2O3`) for APL [295]. However, several challenges remain and hinder the development of robust humanized animal models. These include immature differentiation [164] and developmental defects such as poorly developed lymph nodes [41].

Biomarkers Identification. *Biomarkers* identification is another critical part of the pipeline as it can help in selecting appropriate animal models for preclinical studies, as well as for identifying lead compounds [42, 232]. A *biomarker* is an indicator of a biological process that is measurable and exhibits sufficient sensitivity and specificity required for use in preclinical or clinical setting [114]. In the context of combination therapies, appropriate choices of biomarkers are crucial for confirmation of the mechanism-of-action of the drug combination. Towards this goal, Dawson et al. suggested the use of reverse phase protein arrays (RPPA)⁴ analysis with phenotypic response in a dose matrix study to identify potential biomarkers for drug combinations [84]. Mathew et al., on the other hand, proposed a systematic approach to identify genomic markers of drug sensitivity in cancer cells [125]. In particular, they use high-throughput cancer cell line screening and genomic analysis to establish links between drug activities and functional complexity of cancer genomes.

⁴RPPA is a chip and antibody-based proteomics approach for performing broad multiplex analysis of proteins, such as identifying post-translational modifications in protein extracts [84].

2.2.3 Clinical Phase

The primary goal of the clinical phase is to establish the safety and efficacy of the proposed treatment in human beings after successful testing in animal models.

Clinical Study Design. Clinical studies can be classified into *clinical trials* and *observational studies*. *Clinical trials* are studies in which investigators actively decide on the specific interventions to be given to participants whereas in *observational studies*, the investigators simply observe treatment outcomes of participants [236]. Although the participants can be treated with the interventions being studied, the investigators do not dictate specific interventions. Clinical trials, especially randomized, placebo-controlled trials (also referred to as randomized trials), remain the preferred method for assessing treatment as observational studies have been found to overestimate treatment effects sometimes due to selection bias [236].

The shift from mono-therapies towards combination therapies introduce new demands on clinical study design. In particular, phase I clinical trials for combination therapies should reveal the interaction of individual compounds in the drug combination [225]. For example, are the toxicity profiles of the compounds additive or overlapping? Is the efficacy of the compounds additive, antagonistic or synergistic? However, to date, there are few dose escalation strategies for combination therapies in phase I clinical trials [225]. These approaches include up-and-down design for combinations [165] and continual reassessment method for partial reordering [371]. The former approach is proposed by Ivanova et al. and uses isotonic regression for estimating the maximum tolerated dose (MTD) [300]. The latter approach proposed by Wages et al. is model-based. Specifically, it uses the continual reassessment method and considers orderings between combinations in order to relax assumption of a monotonic dose-toxicity curve [371]. Interested readers may refer to [300] for a review of these strategies. Apart from dose escalation strategies, there is a need for improved methods to determine interaction of drugs within combinations. In [62], Chou discussed the use of the combination index

equation for determining synergism and antagonism in drug combinations. Software such as *CompuSyn* [64] has also been developed to facilitate dose-effect analysis.

Clinical Trial Management. Clinical trial management is another important area in the clinical phase as mismanagement can lead to failure of trials [109]. Despite its importance, active research in this area is still lacking. Several areas of concerns raised by Farrell et al. in [109] include the need for trial management model and methodology and efficient monitoring systems to support the trials.

2.3 Target Characterization

Target characterization is the process of defining the characteristics of the targets and is useful in drug design for these targets and in the identification of novel targets that share similar characteristics with known targets. Traditionally, targets are characterized based on their molecular characteristics (*e.g.*, structure and binding sites of targets [223]) and biological function (*e.g.*, regulation of apoptosis [391]). These target characterization approaches all have a key limitation. They focus primarily on that target alone and ignore the presence of other interacting molecules in the system. Understanding how the target interacts with other molecules in the system provides an additional level of insight that is useful for target identification. For example, the degree centrality of a target provides information that can be leveraged to assess potential toxicity of the targets. Targets with high degree are potentially more toxic than those with low degree due to a higher probability of these targets being involved in essential protein-protein interactions [146]. In particular, network-based approaches for target characterization address this limitation. Furthermore, in a recent review [80], Csermely et al. summarized certain key characteristics (*e.g.*, low clustering coefficient, high degree centrality) of disease-related genes.

In the following, we shall discuss two network-based target characterization approaches. In these approaches, targets are characterized in terms of topological features of several different type of networks, including protein co-abundance networks, and protein-protein interaction network.

2.3.1 Topological Feature Characterization of Targets in Protein Co-abundance Network

In [234], McDermott et al. performed characterization of targets in *protein co-abundance networks* using several topological features, including degree centrality and clustering coefficient. These *protein co-abundance networks* are essentially protein-protein interaction (PPI) networks constructed by identifying highly differentially regulated proteins from proteomics data using three filters (*abundance filter*, *correspondence filter* and *correlation filter*). In particular, the *abundance filter* is used to remove protein with differential abundance ratios below a certain threshold whereas the *correspondence filter* and *correlation filter* are used to remove protein pairs with correlation values below a certain threshold. Their study, which evaluates the topological enrichment of targets, reveals that network topology of protein co-abundance networks are useful in identifying potential targets for hepatitis C virus (HCV) infection.

Although this study demonstrates the potential use of global proteomics data for characterizing targets, the proposed approach has certain drawbacks. First, the protein co-abundance networks can contain many false-positive protein-protein interactions due to non-specific binding [148], resulting in poor target characterization. Second, although the study suggests that multiple topological features can be combined for better target characterization, exactly how the topological features should be combined remains as part of future work.

2.3.2 Topological Feature Characterization using Machine Learning

In [404], Zhang et al. proposed the use of machine learning techniques such as support vector machines and logistic regression for characterizing known targets in a manually curated human

PPI network obtained from *UniHI* [55]. The nodes in the PPI networks were grouped into four categories, namely, approved drug targets (obtained from *DrugBank* [385]), human disease genes (obtained from *GeneCards* database [309]), human essential genes and putative non-drug-target proteins. The human essential genes were obtained by first extracting essential mouse genes from the Mouse Genome Informatics Database [30], then linking to these mouse genes to human genes using human-mouse ortholog associations. A total of 15 topological features were used for target characterization. Unlike the previous work, the goal in this work is to identify topological characteristics of drug targets in general, instead of drug targets of specific diseases.

There are certain limitations in this approach. First, similar to the previous work, the use of PPI network for analysis may result in poor target characterization. Although manual curation is used to ensure that the network is accurate, it cannot address issues such as presence of false-positive protein-protein interactions due to limitation in experimental techniques. Second, characterization of drug targets in general may not yield useful target characteristics that can be used for identifying targets of specific diseases. As we shall see later in Chapter 5, known targets in signaling networks tend to be characterized by different sets of topological features.

2.4 Target Prioritization

Target prioritization is the process of ranking targets according to some importance metrics such as *sensitivity* or gene expression level [60]. It has various uses including functional annotation of genes [377] and drug discovery [387]. Several approaches have been suggested for performing target prioritization and these approaches can be broadly classified into *network-centric* and *network-free* approaches. *Network-centric* approaches such as *NetworkPrioritizer* [171] adopts a network view of biological networks and performs *in silico* network-based analysis to prioritize targets in networks. *Network-free* approaches, on the other hand, use experimental data (*e.g.*, gene expression data [220]) and non-network features (*e.g.*, gene struc-

ture and sequence [398]) for prioritizing targets. Unlike network-free approaches that treat the biological system as a black box, network-centric approaches consider biological systems as interactions of molecules. This network view of biological networks provides a means to capture the complexity of these networks. Hence, we advocate that they are more appropriate for prioritizing targets in biological networks.

In the following, we shall discuss two *network-free* approaches (sensitivity analysis [139, 156] and *TargetMine* [60]) and two *network-centric* approaches (*GeneWanderer* [186] and *NetworkPrioritizer* [171]). Interested readers may refer to [248] for a review of additional computational tools for gene prioritization.

2.4.1 Sensitivity Analysis

Sensitivity analysis is often used to assess the influence of model parameters on model output. *Local sensitivity analysis* (LSA) is the most frequently used sensitivity analysis approach [202]. LSA evaluates the influence of parameters using single parametric perturbations [409]. The parameter influence is quantified using *sensitivity coefficients* that are denoted as S_{ij} and defined as follows:

$$S_{ij} = \frac{\partial O_i}{\partial p_j} \quad (2.1)$$

where O_i is the i^{th} model output and p_j is the j^{th} model parameter. This is in contrast to *global sensitivity analysis*-based approach (e.g., SOBOL [331] and *mpsa* [409]) which evaluates the influence of parameters using multiple parametric perturbations simultaneously [202]. Generally, the most sensitive parameters are chosen as the best targets. In particular, Schoeberl et al. applied LSA on a computational model of the ErbB signaling network and identified ErbB3 as a potential target for cancer [315].

However, there are certain key limitations in using sensitivity analysis as target prioritization tools. First, sensitivity analysis approaches are by nature model-free. That is, they treat the

biological system as a black box. As mentioned before in Section 2.2.1, the lack of consideration of interactions between molecular players in the system may undermine the effectiveness of the solutions found. Second, the reliance on a single metric (sensitivity coefficient) precludes the consideration of targets that are not “sensitive”. In addition, this approach biased the search towards targets that are potentially effective, but with high risk of toxicity due to their ability to influence model outputs.

2.4.2 *TargetMine*

TargetMine is an integrated data warehouse designed for gene prioritization and target discovery [60]. In particular, *TargetMine* uses data in the *InterMine* database [329] and incorporates additional source of data such as protein 3D structure (from PDB database [345]), drug (from *DrugBank* repository [385]) and disease (from Disease Ontology [317]).

Chen et al. proposed a general protocol for target prioritization using *TargetMine*. The protocol involves three key steps. First, the candidate genes are uploaded to *TargetMine*. Then, enriched biological themes (*e.g.*, top- k KEGG and GO associations) of these genes are identified. Finally, genes are prioritized by performing a union of biological themes and selecting for genes that are enriched in all these themes.

In summary, *TargetMine* employs an unsupervised database-based approach that facilitates the integration of various data sources for target prioritization. This approach assumes that enriched biological themes are key characteristics of targets in the given network. This assumption works well if the candidate genes are implicated in several different pathways and diseases. However, when the candidate genes are all in the same pathway and associated to the same disease, this assumption would result in similar enrichment of biological themes for all candidate genes, making gene prioritization difficult.

2.4.3 *GeneWanderer*

Köhler and colleagues proposed *GeneWanderer* [186], a gene prioritization method based on a random walk algorithm which computes a score that determines the global similarity of candidate genes to known disease genes. Target prioritization using *GeneWanderer* involves three key steps [248]. First, candidate genes are identified (*e.g.*, all genes located on chromosome 9q). Then, known genes related to disease (referred to as *seed genes*) are gathered by searching for specific keywords (*e.g.*, “Usher syndrome”) on disease related databases (*e.g.*, Online Mendelian Inheritance in Man) or by using literature curation. Finally, the random walk algorithm is applied on the disease-gene network. The random walker assigns a score to the candidate genes based on their distance to all seed genes. Note that this score is used for prioritizing genes.

The reliance on known disease genes limits the usage of *GeneWanderer* to networks associated to diseases that have been well-studied such as cancer. Moreover, it uses a single metric (distance to seed genes) for prioritizing targets. As we shall see later in Chapter 5, known targets in different networks are characterized by different set of metrics. Hence, it raises the possibility that *GeneWanderer* may only be useful in prioritizing targets of a subset of gene networks with particular characteristics, which further limits the use of this tool.

2.4.4 *NetworkPrioritizer*

In [171], Kacprowski et al. proposed *NetworkPrioritizer*, a target prioritization approach based on network topological features (*e.g.*, betweenness centrality and closeness centrality). *NetworkPrioritizer* provides user-customizable rank aggregation algorithm such as *Weighted Borda Fuse* (WBF) and *Weighted AddScore Fuse* (WASF) to aggregate the rankings of the topological features. The aggregated rank is then used for prioritizing targets in the network. In particular, WBF is a generalization of the Borda count aggregation method [85]. First, it ranks

each node based on individual topological features and assigns a score to individual node according to its position in the ranked list. Then, it sums the score of each node across topological features to obtain the aggregated score. WASF, on the other hand, assigns the aggregated score based on a weighted sum of scores for each node when ranked according to individual topological features. Note that a higher rank is assigned to a node with a larger aggregated score.

The full flexibility that *NetworkPrioritizer* provides in terms of choice of topological features, weights and rank aggregation method to use for target prioritization itself presents technical challenges to users. Users need to have a keen understanding of the biological network and characteristics of targets in that network to make informed decisions on appropriate settings to use. However, users may not be equipped with such specific in-depth knowledge. The focus on using only topological features is another limitation of *NetworkPrioritizer*. Network dynamics, an important aspect of biological networks, is ignored although it can provide insights into biological network which is complementary to that provided by network structures [70].

2.5 Target Combination Identification

Improving the quality of target combination selection is a key step in enabling target combination therapies. In this section, we shall focus on work that are related to model-centric target combination identification. Specifically, we shall discuss a association-based approach (TIMMA [349]) and two optimization-based approaches (OPMET [337] and MTOI [392]).

2.5.1 TIMMA: Target Inhibition Inference using Maximization and Minimization Averaging

TIMMA is a network-centric association-based approach for identifying target combinations using a *target inhibition network* that is constructed from functional data on drugs and their targets obtained from target binding assays and high-throughput drug sensitivity screens [349]. Tang and colleagues demonstrated the use of TIMMA in the context of cancer using three different cell lines.

TIMMA consists of four key steps. First, drug-target interaction profiles and single-drug treatment efficacy are used to identify potential targets that are relevant to the survival of the particular cancer cell type. Then, in the *model selection step*, target combinations that are likely to achieve specific drug efficacy are identified. Next, in the *model construction step*, a model (referred to as *efficacy matrix*) is constructed to predict the efficacy of inhibiting the targets in the combinations. Finally, *synergy scores* of target pairs are computed using the efficacy matrix and used for identifying synergistic target combinations.

Specifically, TIMMA makes use of the *probabilistic kinase inhibition map* (PKIM) rule proposed by Pal et al. [269] for predicting drug efficacy. The PKIM rule is defined by the following equation:

$$\rho_{\text{PKIM}}(x) = \frac{\sum_{i \in N} I(d_x \supset d_i) y_i}{\sum_{i, j \in N} I(d_x \supset d_i) y_i + I(d_x \subset d_j) (1 - y_j)} \quad (2.2)$$

where $\rho(x)$ is the predicted efficacy for drug x ; d_i is the set of targets of drug i ; y_i is the drug treatment efficacy data of drug i ; N is the entire set of potential targets found in the first step of TIMMA and $I(\cdot)$ is an indicator function that is equals to one when the argument is true, and zero otherwise.

TIMMA begins by finding *identical set* using the average of the predicted efficacy of drugs with target profiles matching the new drug x . That is,

$$\rho_{\text{TIMMA}}(x) = \frac{\sum_{i \in N} I(d_x = d_i) y_i}{\sum_{i \in N} I(d_x = d_i)} \quad (2.3)$$

Note that the identical set may not always be present. Hence, TIMMA utilizes two additional rules to calculate and update the predicted efficacy (ρ_{TIMMA}). These rules are:

- (i) Maximization prediction rule: update predicted efficacy with Equation (2.4) if the drugs have target profiles that are subsets of a new drug x and supersets of drug h (drug with the highest efficacy value), but have efficacy values lower than h .

$$\rho_{\text{TIMMA}}(x) = \frac{y_h + \sum_{j \in N, j \neq h} I(d_x \supset d_j \cap d_j \supset d_h \cap y_j < y_h) y_j}{1 + \sum_{j \in N, j \neq h} I(d_x \supset d_j \cap d_j \supset d_h \cap y_j < y_h)} \quad (2.4)$$

- (ii) Minimization prediction rule: update predicted efficacy with Equation (2.5) if the drugs have target profiles that are supersets of a new drug x and subsets of drug l (drug with the lowest efficacy value), but have efficacy values higher than l .

$$\rho_{\text{TIMMA}}(x) = \frac{y_l + \sum_{j \in N, j \neq l} I(d_x \subset d_j \cap d_j \subset d_h \cap y_j > y_l) y_j}{1 + \sum_{j \in N, j \neq l} I(d_x \subset d_j \cap d_j \subset d_h \cap y_j > y_l)} \quad (2.5)$$

The estimates given by Equations (2.4) and (2.5) provide the lower and upper bound for the predicted efficacy and the average of the two estimates is used as the maximum likelihood estimate of the predicted efficacy.

Although this approach yields synergistic target candidates that are druggable, it relies on drug treatment data that may not always be publicly available. Moreover, it restricts the target combinations to those whose targets are associated with existing drugs. That is, novel targets that are not hit by existing drugs but are beneficial in treating the disease are ignored. In addition, the lack of consideration of potential off-target effects of target combinations may lead to effective combinations that are toxic.

2.5.2 OPMET: Optimal Enzyme Drug Target Identification Algorithm Based on Metabolic Networks

OPMET, introduced by Sridhar and colleagues in [337], identifies enzyme combinations for metabolic networks. It uses a branch-and-bound algorithm and two heuristics to search the candidate solution space according to a cost model that takes side-effects into account. The two heuristics are:

- (i) Static OPMET: Prioritizes enzymes according to their impacts on specific metabolite production when they are inhibited and preferentially selects for enzymes having the greatest impact.
- (ii) Dynamic OPMET: Updates the priorities of the enzymes during the search.

Specifically, OPMET uses two pruning strategies to perform branch-and-bound. They are:

- (i) Target filter: Remove a subspace when there is no enzyme combinations in that space that can inhibit the production of all metabolites that are marked for inhibition.
- (ii) Non-target filter: Remove a subspace when no more enzyme combinations in that space have side-effects that are less than the current optimal solution found.

OPMET was evaluated on several small *Escherichia coli* metabolic networks (14 to 32 enzymes) and found to reduce search time considerably when compared to an exhaustive search. In particular, the optimal solution was found within 0.005% of the total search space that was explored. However, OPMET suffers from certain limitations. First, the branch-and-bound algorithm it employs requires enumeration of the entire search space and are suited only for networks of small size [16]. Second, OPMET considers only full inhibition of the enzymes in the metabolic networks. However, restoration of a diseased state to a normal state do not always warrant total elimination of enzymatic activities and finer tuning of activity may be necessary [208]. For instance, to treat diabetes, it is important to maintain the blood glucose level at a healthy range instead of eliminating blood glucose completely.

2.5.3 MTOI: Multiple Target Optimal Intervention

In [392], Yang et al. introduce MTOI, a simulated annealing-based approach for identifying target combinations in disease-related molecular network that are optimized for specific off-target effects. MTOI consists of two phases, namely, *drug target search phase* and *optimal multi-target control solution identification phase*.

In the *drug target search phase*, potential drug targets in the disease network are identified based on their ability in restoring the disease state to the normal state when it is perturbed alone (referred to as *target influence*). An objective function⁵ is used to determine the target influence

⁵The objective function used in MTOI is the difference between the current network state and the user-desired state [392].

and is minimized using Monte-Carlo simulated annealing. Targets that can better restore the normal state when perturbed are deemed as having greater target influence and these targets are selected for the second phase.

In the *optimal multi-target control solution identification phase*, the target combinations that can achieve the best “control” (restoration from disease to normal state) are selected from the targets identified in the first phase. The target combinations are found using Monte-Carlo simulated annealing with the same objective function as in the *drug target search phase*.

Unlike OPMET, MTOI uses simulated annealing which does not require enumeration of the entire search space. Hence, it can be used for networks of larger size. In addition, it does not restrict the target activity to be either zero or one, but instead, allows its full range which is a more realistic representation of the target combination problem. However, MTOI has two major limitations. First, specific off-target effects have to be specified by the user. Knowledge of which off-target effects are more important for a particular network demands expert knowledge of that network and are not always known to the user. Second, MTOI treats all candidate target combinations equally and selects candidate combination randomly although some combinations may have greater likelihood of achieving the desired therapeutic effect. For instance, combinations with targets that act synergistically are likely to result in smaller off-target effects since lesser perturbation of these targets are required to achieve the same effect as when the targets are perturbed individually. A judicious selection process of candidate target combinations can provide us an opportunity to effectively reduce the vast search space and at the same time increase the likelihood of finding better target combinations. In this thesis, we propose TRECENTO, a heuristics-based simulated annealing approach for target combination identification that addresses the abovementioned limitations.

2.6 Summary

Compared with existing network-based approaches, our research has the following novelty.

- Unlike state-of-the-art network-based approaches that uses PPI networks, we study target characterization in signaling networks. Recall that PPI networks have two key limitations when compared to signaling networks. First, they do not provide insights into the dynamics of the interaction molecular players within the system. Second, PPIs have high false-positive rates that can undermine the study of target characterization. Hence, we advocate that it is important to study target characteristics in the context of signaling networks. In addition, these state-of-the-art approaches perform study on a single network that is either implicated in a specific disease (*e.g.*, HCV) or general (*e.g.*, human PPI containing targets from various diseases). In this thesis (Chapter 5), we perform target characterization across several signaling networks. The use of multiple networks allow us to gain better insights on target characteristics. For example, can a single set of features characterize targets in all signaling networks?
- Existing network-based approaches on target prioritization utilize various network features and rank aggregation methods. These approaches generally demand users to have in-depth knowledge of the biological networks and the characteristics of targets in these networks in order to make appropriate configuration settings for algorithms and tools to perform target prioritization. The lack of appropriate technical expertise in the users can undermine the effectiveness of these approaches. In this thesis, we study the use of a network-based approach, TRISTAN, in Chapter 6 that leverages on learning of target characteristics using examples of signaling network with known targets to perform target prioritization. TRISTAN, like existing approaches, prioritizes targets using various network features. However, users do not have to make specific decisions on the exact features or weights to use for target prioritization as these configuration settings are made internally in TRISTAN based on learning from other signaling networks. In addition, unlike state-of-the-art approaches that considers only structural features, TRISTAN

considers both structural and dynamic features of biological network for target prioritization.

- Existing network-based approaches on target combination identification use techniques that are either association-based or optimization-based. We advocate the use of the latter techniques as these allow us to better quantify off-target effects that are important for identifying target combinations that are not only effective, but also safe. However, few optimization-based approaches have been proposed and they all suffer from two key limitations. They assume that all targets are equally probable for a combination. However, targets influence effects of the combinations differently and appropriate choice of targets can potential improve the target combinations. In addition, they demand specific technical expertise from users (*e.g.*, knowledge of what off-target effect to optimize) and a lack of the required expertise can undermine the effectiveness of these methods. In this thesis (Chapter 7), we study the use of heuristics to improve target selection and propose STERIOD, a heuristics-based simulated annealing approach for identifying synergistic target combinations that are optimized for off-target effects that are general to the entire system.

Chapter 3

Architecture Of TRECENTO

In the previous chapter, we reviewed representative work in *in silico* target characterization, prioritization and combination identification and discussed their limitations. In this chapter, we describe the architecture of TRECENTO, a novel network-based framework for identifying target combinations in signaling networks, that addresses some of these limitations. The rest of this chapter is organized as follows. First, we introduce the concept of signaling network models in Section 3.1. Then, we discuss the challenges of designing the TRECENTO framework in Section 3.2. Next, in Section 3.3, we briefly introduce the MAPK-PI3K [143] signaling network implicated in ovarian cancer, which we use as a running example to illustrate our framework. Finally, we discuss the architectural framework in Section 3.4. Detailed discussions of the various subsystems in the framework are found in Chapters 5 to 7.

3.1 Models of Signaling Networks

In this section, we describe the graphical model representation of signaling networks and ordinary differential equation (ODE) models which are essential for understanding various network-based analyses described in this thesis (Chapters 5 to 7).

Figure 3.1: MAPK-PI3K signaling cascade adapted from [143]. The green solid line highlights an example of a hyperedge involving 3 nodes Akt_{PIP3} , Akt and $PIP3$. The red broken line highlights an example of feedback loops involving ERK , $ERKP$, $ERKPP$, $MEKPP$ and $MKP3$. The grey circle highlights an example of a hub node.

3.1.1 Graphical Model Representation

Systems biology studies the complexity of biological systems by modelling them at the systems level. This provides a more holistic view of the various interactions between different molecular players. In particular, a biological signaling network can be modelled as a directed hypergraph $G = (V, E)$ [172, 182, 299] where the nodes V represent molecules (e.g., proteins) and the hyperedges E represent interactions. Intuitively, a hyperedge connects more than 2 nodes. Formally, a hyperedge $(U, W) \in E$ is an edge connecting one set of nodes U to another

set of nodes W where $U, W \subset V$. An example of a hyperedge provided in Figure 3.1 (green solid line) which illustrates the complex formation of $\text{Akt}_{\text{PIP}3}$ from Akt and $\text{PIP}3$ can be denoted as $((\text{Akt}, \text{PIP}3), \text{Akt}_{\text{PIP}3})$. Analysis of directed hypergraphs is generally more complex than graphs and many graph algorithms cannot be used directly on hypergraphs [182]. Hence, they are often transformed into graphs containing simple edges¹ for analysis. Methods (*e.g.*, bipartite and substrate graph representation) exist for such transformation [182] (Figure 3.1). In this thesis, we use the bipartite graph² representation as it retains the original information of the hypergraph [182]. Details of converting a hypergraph to its bipartite representation are provided in Chapter 6.

Signaling networks generally contain characteristics such as feedback and feedforward loops which are common in complex regulatory control [198]. An example of feedback loops in Figure 3.1 (red broken line) is the double phosphorylation-dephosphorylation cycle involving ERK , ERK_P and ERK_{PP} where $\text{MKP}3$ acts as the phosphatase and MEK_{PP} acts as the kinase. These feedback loops in turn give rise to graph characteristics such as strongly connected components (SCC³).

3.1.2 Ordinary Differential Equation (ODE) Model

The ordinary differential equation (ODE) model describes the system's behaviour over time by using mass-action kinetics⁴ to model the production and consumption rates of different molecular species [6]. These models are typically constructed by translating prior knowledge of production and consumption rate of different molecular species into differential equations

¹A simple edge connects a maximum of two nodes.

²A bipartite digraph is denoted as $B = (P \cup Q, E_B)$ where P and Q are disjoint node sets and there is no edge connecting nodes in the same set.

³Formally, a SCC is defined as a graph $G = (V, E)$ such that for each node $u \in V$, there exists a path $p_{(u,v)}$ for each and every node $v \in V$

u where $p_{(u,v)}$ is a series of serially connected edges that together connects u to v and is denoted as $p_{(u,v)} = \{e_1, \dots, e_n\}$ where $e_n \in E$ and $|V| > 1$.

⁴The law of mass-action [370] states that for an elementary reaction where all the stoichiometric coefficients of the reactants are one, the rate of reaction is proportional to the concentrations of the reactants.

[6]. The determination of these reaction kinetics can be technically challenging [408]. Hence, a large proportion of these kinetics are usually estimated using methods such as numerical approximation of an initial value problem [28, 123] and collocation approaches using basis function expansions [33, 291]. Interested readers may refer to [292] for a review of different parameter estimation approaches. Hence, in general, many of these kinetic parameters are constrained but not fully defined. Despite this uncertainty, these underdetermined ODE systems can still model real, observable biological behaviour. These models in fact provide a means for quantitative study and are valuable for both basic research and industry (*e.g.*, drug discovery and development) since experiments are often difficult and costly [6]. Although these models are still small in size now, they are expected to grow in the future and become an important and accepted way of representing biological knowledge [6].

In this thesis, we develop a framework called TRECENTO (Section 3.4) that analyses signaling networks modelled using ODEs. Note that TRECENTO assumes the ODE model for the user-specified signaling network is fully specified and available.

3.2 Challenges

In this thesis, we present TRECENTO, a framework for identifying synergistic target combinations for signaling networks. In particular, TRECENTO uses ODE models of signaling networks as inputs. Recall that in the previous section, these networks tend to be small due to difficulty in determining kinetic parameters. Though small, these networks present challenges that have to be addressed in the design of TRECENTO. We shall now discuss these challenges along with other key challenges related to the design of the TRECENTO framework.

First, in the context of identifying target combinations, even a small model with tens to hundreds of nodes can pose a computational challenge when considering large combination size (*e.g.*, 6-target combinations for cancer therapy). Recall that exhaustive testing of all 6-target combinations for a 100-node network using just 3 different *target activities* already requires

examining 8.9×10^{11} different candidate combinations. Coupled with the fact that the actual range of valid *target activities* may be wider, the potential calculations required tend to increase exponentially, rendering a brute force approach impractical. Hence, there is a need to design techniques that can reduce the target combination search space and allow an efficient search of the reduced search space.

Second, following from the above-mentioned challenge, an important aspect of TRECENTO design is the way candidate target combinations are generated. Instead of randomly generating candidate target combinations from the entire search space, a search that is focussed on candidate target combinations satisfying “good” candidate rules, such as the potential of being a good target, may yield better candidate solutions. However, there is currently a lack of understanding of what characterizes “good” targets in signaling networks. For example, do known targets tend to have the same characteristics in different networks? What characteristics best identifies the known targets? The lack of understanding is due in part to lack of available known targets of signaling networks to facilitate such network-based target characterization study. Although relevant clinical studies of these targets may have been conducted, the information are either buried within some medical journals or stored in several different databases, such as *ClinicalTrials.gov* [259] and *DrugBank* [385], and has to be integrated before use.

Finally, despite a lack of thorough understanding of target characteristics, several network-based approaches [58, 160, 171] have been proposed to address *target prioritization*. In particular, Hwang and colleagues found that bridging centrality is able to identify known targets in several protein-protein interaction networks [160]. However, these work focus primarily on using network topology and ignore network dynamics, an important aspect of signaling networks [7, 288]. The lack of consideration of network dynamics in these recent work is likely due to scarcity of relevant dynamic features that can be applied to target prioritization studies. Note that we did not find any appropriate dynamic features to use for target prioritization and consequently propose a dynamic feature in Chapter 6.

3.3 The MAPK-PI3K Signaling Network

In this section, we describe the MAPK-PI3K network that we use as a recurring example throughout this thesis. The MAPK-PI3K signaling network is involved in up to 30% of human cancers [218] due to its roles in cell survival signaling. Here, we briefly describe the heregulin (HRG)-induced ErbB receptor signaling network in Chinese hamster ovary cells constructed by Hatakeyama and her colleagues [143]. This network consists of two interacting pathways, the MAPK and PI3K-Akt signaling cascades. The ODE model is found in the *BioModels* repository [201] with accession ID BIOMD0000000146. Other networks being studied in this thesis are described in Sections 4.2 and 6.4.

MAPK cascade. Extracellular signals such as HRG can result in dimerization of receptor tyrosine kinases (R) on the cell surface [314]. This causes the intracellular portions of the receptors to be phosphorylated, which then binds Src-homology and collagen domain protein (Shc) forming the R-Shc complex (RShc) [265]. Shc in turn recruits an adaptor protein known as growth factor receptor-bound protein 2 (Grb2) [265]. This complex binds son of sevenless (SOS), thereby activating SOS [389]. The resultant complex ShcS facilitates exchange of membrane-bound Ras-GDP to Ras-GTP [333]. The activated Ras-GTP in turn binds Raf leading to activation and phosphorylation of Raf (Raf^{*}) [15]. Raf^{*} then phosphorylates and activates MEK which in turn phosphorylates ERK. Phosphorylated ERK (ERKPP) is translocated from the cytoplasm to the nucleus where it activates various transcription factors (*e.g.*, c-Myc) [217].

PI3K-Akt cascade. Parallel to the MAPK cascade is the PI3K-Akt pathway. This pathway is activated when HRG-stimulated receptor (RP) binds to the acceptor sites for the SH2 domain of the p85 subunit of PI3K, thereby activating PI3K (PI3K^{*}) [135]. PI3K^{*} phosphorylates

Figure 3.2: The system architecture of TRECENTO.

phosphoinositol lipids which recruit and activate Akt ($\text{Akt}_{\text{PIP}}^{\text{PIP}}$). The MAPK and PI3K-AKT cascades interact at the level of Raf and PP2A [156]. Specifically, $\text{Akt}_{\text{PIP}}^{\text{PIP}}$ phosphorylates Raf on its Ser^{259} residue, inhibiting Raf. The role of PP2A in the cross-talk between the two cascades is more subtle and is a result of its catalysis of MEK and Akt deactivation which cause competition for PP2A . Hence, activation of one results in inhibition of the other.

3.4 The TRECENTO Architecture Framework

The primary goal of TRECENTO is to identify target combinations that achieve the user-desired *therapeutic effect* with minimized *off-target effects* for a given disease-related signaling net-

work. TRECENTO is a data-driven network-based framework. That is, it uses signaling networks as inputs and performs various analysis on these networks. The results of these analysis are used to guide the selection of synergistic target combinations. In particular, we design the TRECENTO framework to address the challenges discussed in Section 3.2. We use simulated annealing as the backbone of the framework as it is robust and facilitates search of a global optimum solution within the vast target combination search space (Chapter 7) [130]. In addition, TRECENTO uses heuristics based on *target prioritization* (Chapters 5 and 6) and the *Loewe additivity theorem* (Chapter 7) to effectively reduce the solution search space. Note that in the target prioritization-based heuristic, TRECENTO first performs structural feature-based characterization of known targets in signaling networks to identify relevant predictive structural features that shall be used (Chapter 5), then incorporates profile shape similarity (our proposed dynamic feature) to perform the final target prioritization (Chapter 6). Recall that in Section 3.2 there is a lack of available known targets and this poses a challenge in performing target characterization. In particular, we perform manual target curation (detailed in Chapter 4) to address this challenge.

The architecture essentially consists of two subsystems: *Target Prioritization* (Chapters 5 and 6) and *Target Combination Finder* (Chapter 7). *Target Prioritization* prioritizes nodes in the network according to its effect on the *output node* whereas *Target Combination Finder* identifies combinations of targets that satisfy the therapeutic goal with minimized *off-target effects*. In the context of a disease-related network, the *output node* is usually a downstream node whose activity determines the phenotypes that are manifested in response to signals flowing in the network [294]. For instance, in the MAPK-PI3K network, activated ERK (ERKPP) can be selected as an output node since double phosphorylation is required for ERK activation and nuclear translocation [209] in order to regulate transcription factors including those related to growth [301]. High activity of ERKPP can result in excessive growth which contributes to the formation of cancer [340].

In the following, we first give an overview of each subsystem and briefly discuss the rationale of choosing specific algorithms and heuristics, then we use the `MAPK-PI3K` signaling network as an example to illustrate the functions of these subsystems in the framework.

3.4.1 Target Prioritization

Target prioritization approaches (reviewed in Chapter 2) have been used in various contexts to address information overload in the biological field. For instance, Chen et al. used a protein-protein interaction network-based method to rank disease candidate genes [58]. Research in target prioritization is potentially useful in helping researchers with planning experiment [142] since resources is limited and experiments can be costly and time-intensive. This is especially true in drug development [249]. In this thesis, we use a network-based approach for ranking candidate targets in the signaling network for the purpose of identifying good target combinations.

In [58], Chen and colleagues adapted network features (*e.g.*, *PageRank* with priors) to prioritize gene targets. They found that under appropriate settings, different features had the same performance. This observation gives rise to several questions related to network-based prioritization: (1) Is there a single feature that consistently outperform others in terms of prioritizing targets? (2) Are multiple features necessary? (3) Are structural features alone enough? Hence, there is a need to better understand the link between network characteristics and the set of features that are predictive of targets in the network and how best to obtain the score (*putative target score*) that shall be used for prioritizing targets. We design the *Target Prioritization* subsystem with this consideration.

Specifically, the *Target Prioritization* subsystem is split into an offline and an online component. The offline component is responsible for *characterizing* signaling network targets. Specifically, it learns the structural features that are predictive of targets in a given signaling network and how best to combine the features to obtain the putative target score for a given

Figure 3.3: Details of some modules in TRECENTO.

signaling network. The result of the offline component is a structural feature-based *prioritization model* that can be used to generate structural feature-based rankings of targets in a given network. Note that the offline component generates many different prioritization models for different signaling networks with known targets. Given a signaling network with unknown targets (referred to as *unseen network*), the online component first identifies an appropriate prioritization model (generated by the offline component) for the unseen network and uses it to obtain structural feature-based rankings of its targets. Then, these rankings are integrated with the ranks based on a dynamic property for performing target prioritization. In this thesis, we examine one dynamic feature (PSSD) and sixteen structural features. The structural features are further elaborated in Chapter 4 whereas PSSD is described in Chapter 6. Note that the lack of available dynamic features hinders our analysis of the effects of dynamic features on target prioritization and our study is consequently restricted to the study of PSSD, our proposed dynamic feature. In the design of the *Target Prioritization* subsystem, we deliberately plan for an offline component to characterize known targets of signaling networks in order to obtain insights on predictive structural features that best characterize targets. Then, the insights are used to make informed decisions in the online component when prioritizing targets.

Offline Component. In the *Target Characterization Module* (detailed in Chapter 5), signaling networks having known targets (referred to as *training networks*) from repositories, such as

BioModels [201], are used to learn the best set of structural features (referred to as *predictive structural features*) for characterizing known targets in signaling networks. A *prioritization model* based on these features is then generated. The *predictive structural features* and *prioritization models* are stored in the *Target Characterization Knowledge Base*. Note that the prioritization model shall be used subsequently to generate structural feature-based ranks to facilitate target prioritization of a given signaling network (without known targets) in the online component of *Target Prioritization*.

For each training network, an appropriate output node is chosen based on the network. For instance, in the Ras-activation network [105] implicated in ovarian cancer, we choose the guanosine triphosphate (GTP)-bound Ras located on the plasma membrane (Ras_{GTP-PM}) as the output node. GTP binding activates Ras and Ras localization in the plasma membrane is necessary for activating its downstream effectors [2]. As we shall see later, these output nodes are required by the *Preprocessor* submodule .

We shall now step through details of the *Target Characterization Module* illustrated in Figure 3.3. The *Preprocessor* submodule performs several functions. First, it identifies nodes (referred to as *candidate targets*) that are likely regulators (nodes positioned upstream) of the output node based on the interconnections of nodes. Then, it extracts the sixteen structural features that we study. Finally, it ranks each candidate target based on each structural feature. The *Data Partitioner* submodule, as the name suggests, partitions the preprocessed data into a training, a model selection (validation) and a test set for use by the *Learner* submodule. In the *Learner* submodule, a supervised machine learning algorithm is developed to learn the set of predictive structural features that best characterizes known targets of a given network, and a prioritization model (based on these features) for generating the structural feature-based ranking score. The algorithm uses a set of known targets as benchmark for learning. The known targets are curated from literature published in the *PubMed* repository and clinical trials repositories such as [259]. The curation process is described in detail in Chapter 4.

In this thesis, we model the *target characterization* problem as a *supervised learning problem* and develop a support vector machine (SVM)-based algorithm for the *Learner* submodule. The SVM algorithm uses *structural feature selection* and *weighted misclassification cost* to improve the performance of the final prioritization model by addressing the issues of irrelevant features, class membership uncertainty and imbalanced data set (Chapter 5). Specifically, *structural feature selection* removes redundant structural features from the prioritization model while *weighted misclassification cost* assigns different misclassification cost according to class membership. The results of the offline learning are a set of predictive structural features and a prioritization model based on these features. Note that this architecture is a general framework and is not tied to a specific implementation for the modules. For instance, SVM can be substituted with other superior machine learning techniques in the *Learner* submodule.

Online Component. The user-specified signaling network (e.g., MAPK-PI3K network) is first preprocessed by the *Network Preprocessor Module* which filters for candidate nodes that are likely regulators of the output node based on the interconnections of the nodes. Note that targets of the given network are generally unknown. Then, four key steps are performed. First, the *Network Mapper Module* selects an appropriate prioritization model for the given network from the *Target Characterization Knowledge Base* using a *network similarity ranking* approach. Second, the *Structural Features Ranker Module* generates a structural feature-based ranks for the given network using the prioritization model. Third, the *Dynamic Structure Ranker Module* computes the PSSD rank of the nodes. Finally, the *Putative Target Score Ranker* combines PSSD ranks and the structural feature-based ranks to get the putative target scores. This score is used subsequently in the *Combination Search Module* in the *Target Combination Finder* subsystem for selecting appropriate targets in the candidate combination.

Note that signaling networks used for generating prioritization models in the offline component are different from signaling networks whose targets have to be prioritized in the online

component. In the design of TRECENTO, we assume that it is unlikely for one prioritization model to generalize the characteristics of known targets in all signaling networks due to the complexity and diversity of signaling networks. As we shall see later in Section 5.6, experiments on four signaling networks support this assumption. Hence, there is a need to identify an appropriate prioritization model generated by the offline component for prioritizing targets in a given signaling network (with unknown targets). In Section 6.7.1, we examine several network similarity ranking methods to identify an appropriate method for the *Network Mapper Module* to select a relevant prioritization model.

In the generation of the *putative target score*, we deliberately classify the network feature-based scores as structural feature-based scores (integrated scores based on various structural features) and dynamic feature-based scores in order to distinguish the two types of network features. These two scores are *integrated* using a weighted sum approach to facilitate investigation of the effects of structural features and network features on target prioritization since the relative weights of the scores indicate the relative influence of the feature types on the prioritization results.

3.4.2 Target Combination Finder

The *Target Combination Finder* subsystem consists of two modules, *Network Modifier Module* and *Combination Search Module*, which we shall elaborate in turn. Figure 3.3 shows the details of each module.

In the *Network Modifier Module*, the *Reaction Modifier* submodule converts the reversible reactions into pairs of irreversible reactions using [310] and modifies the reactions (according to Chapter 7) to simulate the effects of the targets when modulated by non-competitive inhibitors or essential activators. Note that the network modifications are for simulating the effects on the network when perturbed by an inhibitor or activator and the modified network behaves no differently from the original network when there is no inhibitor or activator activity. The *Individual Target Activity Extractor* submodule obtains the activities of each individual

target that can achieve the user-desired therapeutic effect (e.g., 50% $ERKPP$ down-regulation) alone (individual target activities). The modified network is used subsequently in the *Combination Search Module* for evaluating the effects of candidate target combinations while the individual target activities are for guiding the selection of appropriate target activities in the candidate combination.

In the *Combination Search Module*, a global optimization approach is developed to search for combinations of targets that together achieve the desired therapeutic goal with minimized off-target effect. The *Candidate Combination Generator* submodule generates the candidate combination to be tested with the help of the *Target Selector* and the *Activity Selector* submodules. In particular, these submodules provide guidance on which target and what activity to select for the combination. In this thesis, we develop a heuristics-based simulated annealing (SA) algorithm (Chapter 7) for selecting suitable targets and target activities for the candidate combination in the *Combination Search Module*. Specifically, the *target prioritization heuristic* selects for more important targets with higher probability. The importance of the target is determined by its prioritization rank obtained from the *Target Prioritization* subsystem. The *LOEWE-based heuristic* bounds the activity range of the target to within the individual target activities to ensure synergistic⁵ effect. It is adapted from the *Loewe additivity isobologram analysis* [406] which assesses drug interaction in a combination. Then, the candidate combination is assessed in the *Combination Effect Simulator* submodule which simulates the effect on the network when the candidate combination is perturbed. Candidates satisfying the therapeutic goals are added to the solution set and ranked based on their off-target effect.

3.4.3 Implementation Overview

We shall now provide an implementation overview of TRECENTO. First, we develop a *support vector machine* (SVM)-based algorithm for the *Target Characterization Module*. Specifically,

⁵Synergy occurs when a pair of targets exerts effects that are greater than the sum of their individual effects, and is generally a beneficial strategy for maximizing effect while minimizing toxicity.

our SVM-based approach (detailed in Chapter 5) uses *feature selection* and *weighted misclassification cost* to improve the prediction accuracy of the prioritization models generated. Then, in the online component of *Target Prioritization*, we implement the *Network Mapper Module* and the *Putative Target Score Ranker* using a *network similarity ranking* approach and a *weighted-sum approach*, respectively. Details can be found in Chapter 6. Finally, in the *Target Combination Finder*, we develop a *heuristics-based simulated annealing* algorithm (detailed in Chapter 7) for the *Combination Search Module*. In particular, the algorithm uses two heuristics that are based on *target prioritization* and *Loewe additivity theorem* to guide the generation of candidate target combinations.

3.4.4 An Example: The MAPK-PI3K Signaling Network

Here, we will use the MAPK-PI3K signaling network implicated in ovarian cancer [143] as an example to illustrate the use of the framework. We choose double phosphorylated ERK (ERKPP in Figure 3.1) as the output node. Double phosphorylation of ERK is required for its activation which leads to rapid nuclear translocation [209]. In the nucleus, ERKPP phosphorylates and regulates many transcription factors including those related to growth [301]. Silencing of ERK1/2 protein expression using siRNA was found to inhibit tumor cell proliferation in ovarian cancer [340]. We set the desired therapeutic effect to 50% ERKPP down-regulation but in practice would depend on the stage of the disease. In the sequel, we assume an ordinary differential equation (ODE) model of the MAPK-PI3K network is available.

We begin by describing target characterization in the offline component using examples of signaling networks with known targets, then move on to describe target prioritization of the MAPK-PI3K network in the online component and finally identification of target combinations in the MAPK-PI3K network. The steps are illustrated in Figure 3.4.

First, the *Target Characterization Module* uses three example of signaling networks with known targets (referred to as *training network*), namely, endomesoderm gene regulatory network, the glucose metabolism network and insulin-stimulated glucose secretion network,

Figure 3.4: Steps to identify target combinations in the MAPK-PI3K network. The black box represents a module in the architecture; the wavy-bottom box represents the given signaling network; the white box represents data generated by the modules and the cylinder represents a repository.

to characterize known targets in these networks using structural features by performing supervised learning. The known targets are generated using manual curation of clinical trials repository and *PubMed*. The results are three prioritization models and their associated predictive structural features. For instance, the predictive structural features associated with the prioritization model of the insulin-stimulated glucose secretion network are *bridging coefficient*, *closeness centrality*, *cycle clustering coefficient* and *undirected clustering coefficient*.

Then, the *Network Preprocessor Module* filters for nodes in the MAPK-PI3K network that are regulators of ERKPP. In the MAPK-PI3K network, all the nodes have a path leading to ERKPP. Hence, none of the nodes can be pruned. Then, the *Dynamic Structure Ranker Module* computes the PSSD rank of each node (according to Chapter 6). Next, the *Network Mapper Module* uses a network similarity ranking approach to identify the training network that is most “similar” to the MAPK-PI3K network and retrieves the prioritization model of that training network for use by the *Target Prioritization* online component. In particular, the *Structural Features Ranker Module* computes the structural feature-based ranks of the MAPK-PI3K network using that prioritization model. For instance, let the insulin-stimulated glucose secretion network be the most “similar” training network to the MAPK-PI3K network. Then, the *Structural Features Ranker Module* shall compute the *bridging coefficient*, *closeness centrality*, *cycle clustering coefficient* and *undirected clustering coefficient* score of nodes in the MAPK-PI3K network and combine the ranks using the prioritization model to find the structural feature-based ranks of the MAPK-PI3K network. Following which, the PSSD ranks and the structural feature-based ranks are combined in the *Putative Target Score Ranker* to obtain the putative target scores. This score is used to prioritize the nodes in the MAPK-PI3K network and used later in the *Combination Search Module*.

Finally, in the *Target Combination Finder* subsystem, the *Reaction Modifier* submodule in the *Network Modifier Module* converts reversible reactions in the MAPK-PI3K network into irreversible ones. For instance, $\text{PIP3} + \text{Akt} \rightleftharpoons \text{Akt} + \text{PIP3}$ is converted into a pair of irreversible

reactions. This pair of reactions ($\text{AktPIP3} \rightarrow \text{PIP3} + \text{Akt}$ and $\text{PIP3} + \text{Akt} \rightarrow \text{AktPIP3}$) is then modified into $\frac{k[\text{AktPIP3}](1+\frac{[A]}{K_A})}{1+\frac{[I]}{K_I}}$ and $\frac{k[\text{PIP3}][\text{Akt}](1+\frac{[A]}{K_A})}{1+\frac{[I]}{K_I}}$, respectively, where k is the affinity constant; A and I are the activator and inhibitor with dissociation constants K_A and K_I , respectively. The *Individual Target Activity Extractor* submodule obtains the activities of the targets that can achieve 50% ERKPP down-regulation alone (individual target activities). In reality, some targets may not achieve the desired therapeutic effect alone. We set the individual target activities of these targets to infinity. For instance, in the reaction $\text{AktPIP3} \rightarrow \text{PIP3} + \text{Akt}$, activation of the reaction at $\frac{[A]}{K_A} = 50.5$ is required to achieve 50% ERKPP down-regulation alone. The *Combination Search Module* makes use of the individual *target activities* to bound the range of *activity* from which the candidate *target activity* is selected from (*Activity Selector*) when generating a candidate target combination (*Candidate Combination Generator*). The prioritized list of nodes from the *Target Prioritization* subsystem is used to generate a *selection probability* (likelihood of selecting the target) of the target in the *Target Selector* so that high priority nodes are selected with higher probability. For instance, consider a two-target combination. Suppose the reaction $\text{AktPIP3} \rightarrow \text{PIP3} + \text{Akt}$ denoted as r_1 is randomly selected as the first target, it will be accepted if its selection probability δ_{r_1} is greater than a certain threshold. The activity of r_1 is selected from within the synergistic range as specified by LOEWE-based heuristic. The second target (e.g., r_2 ($\text{Raf} \rightarrow \text{Raf}^*$)) is randomly selected from the set of candidates excluding r_1 and will be accepted if δ_{r_2} is greater than a certain threshold. The activity of r_2 is again selected from the synergistic range. The candidate target combinations generated are assessed based on whether they achieve the desired *therapeutic effect* and minimize *off-target effects*. In this thesis, a heuristics-based SA is developed for the *Combination Search Module*. Hence, the Metropolis criterion [237] will be used to assess if the combination will be added into the solution set.

3.5 Summary

In conclusion, drug discovery is a multifaceted problem. While others have focussed on monotherapies and combination therapies based on approved drugs (drug-centric approach), our emphasis is quite different. We are interested in finding combinations of targets (target-centric approach) which would best achieve the user-desired therapeutic goal with reduced off-target effect. Unlike the drug-centric approach, the target-centric approach is not limited by drug availability. In particular, our proposed system, TRECENTO, uses network-based analyses to identify suitable target combinations. Since the network wirings reflect the mechanism underlying the disease, the target combinations found are rational. That is, the roles of targets in the disease are known. An obvious caveat associated with this approach is that erroneous networks result in poor combinations. Hence, it is important that the user ensures the input of good quality network. We expect the quality of the networks to improve in the future as research effort in this field intensifies. TRECENTO consists of two main subsystems, *Target Prioritization* and *Target Combination Finder* which are responsible for prioritizing the targets and for identifying the target combinations, respectively. These components leverage on heuristics to address limitations of existing approaches discussed in Chapter 2. Specifically, TRECENTO rejects the assumption that the same predictive features are applicable to all signaling networks and uses heuristics such as structural feature selection to identify predictive features and prioritization models relevant to signaling networks. TRECENTO also recognizes that not all candidate targets are equal. Hence, we use target prioritization and LOEWE-based heuristics to select for appropriate *target* and *target activity* in the *Combination Search Module* instead of doing random selection. We also illustrate the functionality of the subsystems using a real world biological signaling network (MAPK-PI3K network). In subsequent chapters, we will elaborate further on these subsystems.

Chapter 4

Topological Features and Signaling Networks

In Chapter 3, we presented the architecture of TRECENTO, a framework for identifying target combinations in a given signaling network. In this chapter, we introduce some preliminaries that we shall use for performing topological (or structural) feature-based target characterization in the following chapter. The rest of this chapter is organized as follows. In Section 4.1, we first describe the network structural features explored in this thesis. Then, in Section 4.2, we introduce the signaling networks studied in the subsequent chapter. Next, we describe the curation process used to obtain targets in these networks in Section 4.3. Note that the targets found are used subsequently for validating the effectiveness of our proposed structure feature-based target characterization approach in Chapter 5. Finally, we conclude the chapter in Section 4.4.

4.1 Topological Features

In this section, we describe the sixteen structural features studied for structural feature-based target characterization. We provide their formal definitions and discuss their significance in the context of the `MAPK-PI3K` signaling network (Figure 3.1). These structural features are selected as they are generally used to measure the relative importance of a node in a network [189, 324]. Note that although some of these features (*e.g.*, *proximity prestige*) are more

commonly used in non-biological networks (*e.g.*, social networks), this does not preclude their importance in this domain. For instance, as we shall see later in Chapter 5, *proximity prestige* can better discriminate known targets from non-targets than other commonly-used features such as bridging centrality in the glucose-stimulated insulin secretion network [168]. Henceforth, we shall use the notations given in Table 4.1.

Symbol	Description
θ_u	Degree of node u . The in, out and total degrees are denoted as $\theta_{in(u)}$, $\theta_{out(u)}$ and $\theta_{total(u)}$, respectively.
α_u	Eigenvector centrality of node u .
β_u	Closeness centrality of node u .
ϑ_u	Eccentricity centrality of node u .
δ_u	Betweenness centrality of node u .
π_u	Bridging coefficient of node u .
ζ_u	Bridging centrality of node u .
κ_u	Clustering coefficient of node u . The undirected, in, out, cycle and middleman clustering coefficients are denoted as $\kappa_{undir(u)}$, $\kappa_{in(u)}$, $\kappa_{out(u)}$, $\kappa_{cyc(u)}$ and $\kappa_{mid(u)}$, respectively.
μ_u	Proximity prestige of node u .
ω_u	Target downstream effect of node u .
ϕ	Prediction accuracy of support vector machine (SVM) model. The accuracy for the validation and test sets are denoted as $\phi_X(val)$ and $\phi_X(test)$, respectively, where X indicates the method used for training the SVM model. Average prediction accuracy is denoted as $\bar{\phi}$.

Table 4.1: Description of symbols used in structural feature-based target characterization.

4.1.1 Degree Centrality

Degree centrality in networks measures the importance of a node based on the number of edges it has [118]. It is a local centrality measure as it only considers the immediate neighbourhood of the node. For directed networks such as signaling networks, there are three variants of *degree centrality*, namely, *in degree*, *out degree*, and *total degree centrality* which consider only incoming edges, only out-going edges and all edges of a node, respectively. Generally, a node with high *degree centrality* (referred to as hub) (*e.g.*, RP in Figure 3.1) is considered an important node. In particular, studies have found that biological networks resemble *scale-free networks* [272] in that they are robust against random perturbation of non-hub nodes [5]. In signaling

networks, the hyperedges can represent a variety of biological processes such as formation or dissociation of molecular complexes and modification of proteins (*e.g.*, phosphorylation). The commonality among these myriad reactions is that the output of a reaction is usually used as an input for reactions that are further downstream resulting in a cascade of signals. Hence, a high *in degree* node acts as a signal integrator by integrating multiple signals while a high *out degree* node acts as a signal differentiator. For instance, double phosphorylated MEK (MEK_{PP}) is an out degree hub and consequently functions as a signal differentiator.

Definition 4.1 Given a signaling network $G = (V, E)$, the **in degree centrality** of a node $u \in V$ is defined as $\theta_{in(u)} = \sum_{v \in V} |e_{vu}|$ where $e_{vu} \in E$ is the edge connecting node $v \in V$ to u . The **out degree centrality** and **total degree centrality** are denoted as $\theta_{out(u)} = \sum_{v \in V} |e_{uv}|$ and $\theta_{total(u)} = \theta_{in(u)} + \theta_{out(u)}$, respectively.

Another measure that is closely related to the in degree centrality is *degree prestige*. In particular, *degree prestige* is defined as $DP_u = \frac{\theta_{in(u)}}{|V|-1}$ [376]. The key difference between the in degree centrality and degree prestige is the normalization of degree prestige using a constant factor $|V| - 1$. Hence, these two measures have perfect positive correlation. In this thesis, we shall study degree centrality instead of degree prestige because the former centrality values are more commonly used in the analysis of networks including signaling networks [190].

4.1.2 Eigenvector Centrality

Eigenvector centrality (also known as *Bonacich's centrality*) is a refinement of degree centrality as it not only considers how well a node is connected to other nodes in the network, but also how important these other nodes are [34]. Nodes with high *eigenvector centrality* are well-connected to other central nodes. In a signaling network, these nodes tend to be located in the network where signals either converge or diverge depending on whether these central nodes

have high in-degree or out-degree. For instance, activated ErbB4 receptor (ErBB4) which has high *eigenvector centrality* is connected to many other central nodes such as PI3K^* , and provides a means for converging and diverging the various signals passing through the network.

Definition 4.2 Given a signaling network $G = (V, E)$, let N_u be the set of neighbouring nodes of node $u \in V$ and $A = \{a_{ij}\}$ be the adjacency matrix that specifies the connectivity of the network such that $a_{ij} = 1$ implies an edge joining node i to j . Then, the **eigenvector centrality** of u is defined as

$$\alpha_u = \frac{1}{\lambda} \sum_{v \in N_u} \alpha_v \quad (4.1)$$

where λ is a constant and α_v is the v^{th} entry of the corresponding eigenvector of A .

According to the Perron–Frobenius theorem¹, λ has to be the largest eigenvalue of the adjacency matrix A if the centralities are to be non-negative [258].

4.1.3 Closeness Centrality

Closeness centrality measures the importance of a node based on its proximity to other nodes in the network [118]. Nodes with high *closeness centrality* values tend to be close to other nodes in the network. In a signaling network, this measure gives a sense of how influential a node is in terms of propagating its signal throughout the entire network (referred to as influence range). It can also be regarded as how central a node is to the regulation of other nodes in the network [313]. For instance, ShgS which lies near the center of the network is well connected to many other nodes in the network. Hence, it has higher *closeness centrality* compared to other nodes (e.g., MKP3) that lie near the boundary of the network.

¹The Perron–Frobenius theorem states that any square matrix with strictly positive entries has a unique largest real eigenvalue with strictly positive components [215]. Note that in eigenvector centrality, the adjacency matrix A represent distances between the nodes and the entries are assumed to be positive.

Definition 4.3 Given a signaling network $G = (V, E)$, the **closeness centrality** of a node $u \in V$ is defined as

$$\beta_u = \frac{|V|}{\sum_{v \in V} l_{uv}} \quad (4.2)$$

where l_{uv} is the shortest path length between nodes u and v .

Another measure that is closely related to the closeness centrality is *barycenter score* which is defined as $BS_u = \frac{1}{\sum_{v \in V} l_{uv}}$ for a node u [266]. The only difference between the closeness centrality and barycenter score is the scaling of closeness centrality using a constant factor $|V|$. Hence, these two measures have perfect positive correlation. In this work, we will study closeness centrality which is a more commonly used measure in network analysis [190].

4.1.4 Eccentricity Centrality

Eccentricity centrality assigns node centrality value based on the largest shortest path distance from a node to all other nodes in the network [388]. This is in contrast to the closeness centrality value which takes the sum of the shortest path distance. Hence, a node u with high *eccentricity centrality* implies that u is close to all other nodes in the network while a node v with high closeness centrality could be a result of two scenarios: v is very close to many nodes but far away from a few nodes or v is generally close to all nodes [313]. In signaling networks, nodes with high *eccentricity centrality* are likely to be influential signal transmitters, regulating many other nodes [313]. For instance, $P13K^*$ which lies near the center of the network had higher eccentricity centrality compared to other fringe nodes such as ERK since fringe nodes tend to be farther away from other nodes in the network.

Definition 4.4 Given a signaling network $G = (V, E)$, let l_{uv} be the shortest path from node u to node v where $u, v \in V$. Then, the **eccentricity centrality** of u is defined as

$$\vartheta_u = \frac{1}{\max_{v \in V, v \neq u} (l_{uv})} \quad (4.3)$$

where $\text{MAX}(\cdot)$ is the maximum operator.

4.1.5 Betweenness Centrality

Betweenness centrality measures a node's importance based on its ease (proximity) in reaching other nodes. Nodes with high betweenness centrality can reach other nodes in the network via relatively short paths and they tend to lie on a majority of the shortest paths between node pairs in the network [39]. In a signaling network, these nodes can be considered efficient and crucial signal transmitters. For instance, AktPI3 , a hub node, has high betweenness centrality in the network as it is well connected to many other central nodes, hence providing fast access to other nodes in the network. Comparatively, nodes (e.g., Akt) that lie on the fringe of the network have low betweenness centrality.

Definition 4.5 Given a signaling network $G = (V, E)$, let d_{st} denote the number of shortest paths from node s to node t and $d_{st}(v)$ denote the number of shortest paths from s to t passing through node v where $s, t, v \in V$. Then, the **betweenness centrality** of v is defined as

$$\delta_v = \sum_{s \neq v \neq t \in V} \frac{d_{st}(v)}{d_{st}}. \quad (4.4)$$

4.1.6 Bridging Centrality and Bridging Coefficient

The *bridging centrality* identifies *bridging nodes* (nodes with high *bridging centrality* value) which are located between functional modules in a signaling network and mediate signal flow between the modules [160]. Hwang and colleagues found that compared to hub nodes, bridging nodes are more effective drug targets² with fewer off-target effects in several networks³ [160].

²The drug targets were found by literature curation.

³The yeast PPI network, C21-steroid hormone metabolism network, steroid biosynthesis network and a protein-interaction network of sudden cardiac death susceptibility genes

Figure 4.1: Directed triangle graph [107].

The *bridging centrality* of a node is derived from the betweenness centrality [39] and the *bridging coefficient* [160]. Specifically, the *bridging coefficient* measures the average probability of a node transmitting signals to its direct neighbourhood [160]. For instance, `PIP3` has *high bridging coefficient* and *bridging centrality* since it is positioned at the boundary of a strongly connected component (SCC) within the network and helps to transmit signals between nodes outside the SCC and those within it.

Definition 4.6 Given a signaling network $G = (V, E)$, let $\theta_{total(v)}$ be the total degree of node $v \in V$, N_v be the set of neighbors of v , and η_i be the number of outgoing edges of node i , where $i \in N_v$. Then, the **bridging coefficient** of v is defined as

$$\pi_v = \frac{1}{\theta_{total(v)}} \sum_{i \in N_v, \theta_{total(i)} > 1} \frac{\eta_i}{\theta_{total(i)} - 1} \quad (4.5)$$

Definition 4.7 Given a node v , let the inverses of its betweenness centrality rank and bridging coefficient rank be denoted as $\psi_{\frac{1}{\delta:v}}$ and $\psi_{\frac{1}{\pi:v}}$, respectively, where a rank based on feature x orders the nodes in decreasing value of x . Then, the **bridging centrality** of v is defined as

$$\zeta_v = \psi_{\frac{1}{\delta:v}} \times \psi_{\frac{1}{\pi:v}}. \quad (4.6)$$

4.1.7 Clustering Coefficient

Clustering coefficient, a local measure, determines how well the neighbourhood of a node is connected [378] by considering how close the neighbourhood is to being a clique where every node within the clique is connected to every other node in it [5]. Intuitively, the *clustering*

coefficient of a node is the ratio of the number of edges between its neighbours to the total number of possible edges between its neighbours. The original definition was designed for undirected graphs. For a directed graph, a variety of definitions exists [107] when the direction of the edges of the triangles in the node neighbourhood (Figure 4.1) are considered. We evaluate both the *directed* (four variants) and *undirected clustering coefficient* of the signaling networks in this thesis to assess if directions of the edges provide additional insights to our target prioritization problem. Nodes with high *clustering coefficient* are deemed to be more robust signal transmitters as there exist redundant paths to other nodes in its immediate neighbourhood. For instance, RP has very low *clustering coefficient* and removal of this node can disrupt signal transmission extensively in this network. Indeed, RP has roles in many processes such as survival, cell differentiation and growth inhibition [49].

Definition 4.8 Given a signaling network $G = (V, E)$, let $e_{ij} \in E$ denote an edge connecting node i to j where $i, j \in V$ and $A = \{a_{ij}\}$ be the adjacency matrix where $a_{ij} = 1$ if and only if $\exists e \in \{e_{ij}, e_{ji}\}$ and zero otherwise. Then, the **undirected-, in-, out-, cycle- and middleman-clustering coefficient** of a node $u \in V$ denoted as $\kappa_{\text{undir}(u)}$, $\kappa_{\text{in}(u)}$, $\kappa_{\text{out}(u)}$, $\kappa_{\text{cyc}(u)}$ and $\kappa_{\text{mid}(u)}$, respectively, are defined as

$$\kappa_{\text{undir}(u)} = \frac{(A^3)_{ii}}{\theta_{\text{total}(u)}(\theta_{\text{total}(u)} - 1)}, \quad (4.7)$$

$$\kappa_{\text{in}(u)} = \frac{(A^T A^2)_{ii}}{\theta_{\text{in}(u)}(\theta_{\text{in}(u)} - 1)}, \quad (4.8)$$

$$\kappa_{\text{out}(u)} = \frac{(A^2 A^T)_{ii}}{\theta_{\text{out}(u)}(\theta_{\text{out}(u)} - 1)}, \quad (4.9)$$

$$\kappa_{\text{cyc}(u)} = \frac{(A^3)_{ii}}{\theta_{\text{in}(u)}\theta_{\text{out}(u)} - A_{ii}^2}, \quad (4.10)$$

$$\kappa_{mid(u)} = \frac{(AA^T A)_{ii}}{\theta_{in(u)}\theta_{out(u)} - A_{ii}^2}, \quad (4.11)$$

where $\theta_{in(u)}$, $\theta_{out(u)}$ and $\theta_{total(u)}$ are the in, out and total degree of u , respectively; A^T is the transpose of A ; A^n is the matrix product of n copies of A ; and A_{ii} denotes the i^{th} element of the main diagonal of A .

Note that in the above definition, the neighbourhood size must be greater than one. For smaller neighbourhood sizes ($N_u = 0$ and $N_u = 1$), the coefficients are set to zero.

4.1.8 Proximity Prestige

Proximity prestige measures how close all other nodes in the network is to a particular node u [376]. This measure is closely related to closeness centrality. The main difference is that *proximity prestige* is assessed based on the set of nodes that can reach u (influence domain) while closeness centrality is based on the set of nodes that u can reach (influence range). In a signaling network, a node with high *proximity prestige* is one which is easily influenced or regulated by many upstream nodes. For instance, MKP3 which has only outgoing edges, has low *proximity prestige* since it is not regulated by other nodes in the network model.

Definition 4.9 Given a signaling network $G = (V, E)$, let $I_u \subseteq V$ be the set of nodes having at least one path leading to node u and l_{vu} be the shortest path from node v to u where $u, v \in V$. Then, the *proximity prestige* of u is defined as

$$\mu_u = \frac{\frac{|I_u|}{|V|-1}}{\frac{\sum_{v \in V} l_{vu}}{|I_u|}}. \quad (4.12)$$

4.1.9 Target Downstream Effect

When a node in a signaling network is perturbed, all the pathways leading from this node are disrupted as well. The *target downstream effect* [68] of a node v assesses this network disruption based on the probability of perturbing a *downstream node* w and the likelihood of w

causing *off-target effects*. Node w is *downstream* of v if there exists a path from v to w . A node having high *target downstream effect* is likely to cause large network disruption when perturbed. In certain applications such as drug design, it may be desirable to select for targets with low *target downstream effect* so as to reduce potential *off-target effects*. Indeed, gefitinib, a tyrosine kinase inhibitor frequently used in cancer treatment was found to induce interstitial lung disease [255]. The drug-induced pulmonary toxicity is not particular to gefitinib, and is observed in other tyrosine kinase inhibitors as well [274]. Recent research found that the toxicity could be due to suppression of heat shock protein 70 (HSP70) via JNK, a protein downstream of the tyrosine kinase receptor [255]. In the MAPK-PI3K network, upstream nodes such as heregulin-bound ErbB4 receptor (RHrg) have larger *target downstream effect* compared to downstream nodes such as MEK. Selecting targets with smaller *target downstream effect* can minimize such drug-induced toxicity.

Definition 4.10 Given a signaling network $G = (V, E)$, let W denote the set of downstream nodes of $v \in V \setminus W$, $\rho_{v,w}$ denote the probability of perturbing node $w \in W$ when v is perturbed and $\theta_{total(w)}$ denote the total degree of w . Then, the **target downstream effect** of v is defined as

$$\omega_v = \sum_{w \in W} (\rho_{v,w} \times \theta_{total(w)}). \quad (4.13)$$

4.2 Signaling Networks

In this section, we describe the four signaling networks studied in Chapter 5. These networks are obtained from Biomodels.net [201]. The lack of gold standard sets of benchmark targets relevant to these networks necessitate the creation of benchmark targets in order for us to validate our proposed structural feature-based target characterization approach in the subsequent chapter. Manual curation of literature is generally preferred as the process generates substantially lower error rates compared to text-mining approaches [338]. The low error rate

	MAPK-PI3K	Glucose-Stimulated Insulin Secretion	Endomesoderm Gene Regulatory	Glucose Metabolism
BioModel ID	BIOMD0000000146	BIOMD0000000239	BIOMD0000000235	BIOMD0000000244
Related disease or biological phenomenon	Ovarian cancer	Type 2 diabetes mellitus	Embryonic development	Glucose to acetate metabolism
Output node [†]	Double phosphorylated ERK (ERKPP)	Mitochondria ATP (ATP _{mito})	Endo16 protein (Protein_E_Endo16)	Acetate (ACT)
Number of nodes	36	59	622	47
Number of hyperedges	34	45	778	109
Number of targets	9	6	206	16
Average number of neighbors	2.54	3.17	4.40	3.27
Simple FBL (2 nodes)	1	0	6	0
Simple FBL (3 nodes)	1	0	0	0
Coupled FBL (>3 nodes)	1 (6-node), 1 (33-node)	1 (58-node)	1 (364-node)	1 (41-node)
Simple FFL (3 nodes)	0	0	0	0
Coupled FFL (>3 nodes)	1 (44-node)	0	3 (368-node)	0

Table 4.2: Summary of the four signaling networks. [†]The output node in round bracket corresponds to the node name used in the model. FBL and FFL denote feedback and feedforward loops, respectively. Figure 4.2 illustrates some examples of FBL and FFL in the network. Note that within coupled FBL and FFL, there may be multiple instances of simple FBL and FFL. Instances of such simple loops are not reflected in our count.

is important especially if the purpose of the benchmark is for assessing the performance of our proposed approaches. We shall describe later the generation of manually curated benchmark targets in Section 4.3. Focused curation while manageable, can be tedious and time-intensive. Hence, we restrict the number of networks we study to a manageable size of four. In particular, we select networks that have been well-studied with sufficient literature to facilitate the curation effort. Although this thesis examines the target combination identification problem in the context of drug targeting for diseases, the application of target prioritization is not restricted to such and can be applied to areas such as prioritizing regulators of a specific biological process. For instance, in [69], target prioritization is used for ranking key genes that regulate Endo16, a widely-accepted gastrulation marker in sea urchin. Hence, in our selection of networks, we choose two networks that are relevant to human diseases and two that are associated with general biological processes. The human disease associated networks are the MAPK-PI3K [143], and the glucose-stimulated insulin secretion networks [168], while the biological process description networks are the endomesoderm gene regulatory network [196] and the glucose

Figure 4.2: Some feedforward and feedback loop motifs commonly found in directed graphs. The coupled loops in the figure contain instances of simple loops.

metabolism [192] networks. These networks are summarized in Table 4.2.

4.2.1 MAPK-PI3K Network

The MAPK-PI3K network [143] (Figure 3.1) describes the heregulin (HRG)-induced ErbB receptor signaling network in Chinese hamster ovary (CHO) cells. Details of this network is reported in Chapter 3. In this work, we associate it with ovarian cancer since it is derived from CHO cells and the targets that we prioritize should correlate well with drug targets of ovarian cancer within this network.

4.2.2 Glucose-Stimulated Insulin Secretion Network

The glucose-stimulated insulin secretion network [168] (Figures 4.3 and 4.4) describes the glucose-stimulated insulin secretion in pancreatic β cells which consists of five subsystems, namely, glycolysis, the tricarboxylic acid cycle (TCA), the respiratory chain, nicotinamide adenine dinucleotide (NADH) shuttles and the pyruvate cycle. We follow [168]

in associating this network with type 2 diabetes mellitus ($T2DM$) and targets that we prioritize should correlate well with drug targets of diabetes within this network.

4.2.3 Endomesoderm Gene Regulatory Network

The endomesoderm gene regulatory network [196] describes the endomesoderm gene regulation in sea urchin, a model organism for embryonic development. In particular, it describes the key steps in endomesoderm development, namely, the initiation of the endomesoderm specification signal, the maintenance of the specification signal, the activation of the Delta/Notch signaling pathway, and the specification of veg_1 endoderm. We associate this network with the biological phenomenon of embryonic development and the targets that we predict should correlate well with regulators of this phenomenon. Endo16, a downstream protein in this network, was found essential for gastrulation [303], a process that happens early in embryogenesis when the blastula (unstructured assembly of cells) rearranges and forms the three germ layers (ectoderm, mesoderm, and endoderm) of the embryo [332]. In other words, we can interpret the target prioritization task for this model as ranking regulators of Endo16.

4.2.4 Glucose Metabolism Network

This network [192] (Figures 4.5 and 4.6) describes the metabolism of glucose to acetate in *Escherichia coli*. In particular, it describes the Embden-Meyerhoff pathway which focus on glycolysis or gluconeogenesis (depending on the flux direction), the tricarboxylic acid cycle and the carbon flux flow through the glyoxylate shunt. We associate this network with the biological phenomenon of glucose to acetate metabolism in *Escherichia coli* and the targets that we predict should correlate well with regulators of this phenomenon. Since the end product of this process is acetate, we can interpret the target prioritization task for this model

Figure 4.3: Glucose-stimulated insulin secretion network in pancreatic β cells adapted from [168]. (A) A schematic diagram of the network. The modules are color coded: blue (glycolysis), purple (NADH shuttles), yellow (pyruvate cycle), grey (TCA cycle) and beige (respiratory chain). The steps linking glucose uptake to insulin secretion are numbered 1 to 4. (B) Table providing a mapping between the abbreviated node name and the molecule the node represents. (C-E) Biochemical reactions occurring in different modules. For ease of drawing, some molecules (marked in red) are repeated multiple times within the same module.

Figure 4.4: Glucose-stimulated insulin secretion network in pancreatic β cells adapted from [168]. Continuation from Figure 4.3. (F-H) Biochemical reactions occurring in different modules. For ease of drawing, some molecules (marked in red) are repeated multiple times within the same module.

as ranking regulators of acetate.

4.3 Curation of Signaling Network Targets

In this section, we describe the curation process used to obtain the targets for the signaling networks described in Section 4.2. These targets are used subsequently for validating our proposed structural feature-based target prioritization approach in Chapter 5. A summary of the number of targets in each signaling network is reported in Table 4.2. Note that the literature curated were selected based on organism, cell type and environment condition that best match those reported by the literature from which the networks are generated.

4.3.1 Human Disease Associated Networks

Recall that amongst the four networks we study, two are associated with human diseases. These networks are the `MAPK-PI3K` [143] and the glucose-stimulated insulin secretion networks [168].

The curation process for these networks is as follows:

- (i) Obtain a list of unique drugs and compounds relevant to the human disease from a clinical trial database [259] using specific keywords.
- (ii) Obtain the targets of these drugs and compounds via manual curation of drug related databases [385] and literature survey.
- (iii) Identify the targets that are within the scope of the signaling network.

`MAPK-PI3K` Network. Recall that this network is associated with ovarian cancer. Hence, we use the keywords “ovarian cancer drug” to search the clinical trial repository [259] which yielded 2054 studies⁴. However, only 989 studies involve drugs specific for the purpose of

⁴as of 29 April 2014.

Figure 4.5: Glucose metabolism network in *Escherichia coli* adapted from [192]. (A) Biochemical reactions of the glucose metabolism network. For ease of drawing, some molecules (marked in red) are repeated multiple times. (B) Table providing a mapping between the abbreviated node name and the molecule the node represents.

Figure 4.6: Glucose metabolism network in *Escherichia coli* adapted from [192]. Continuation from Figure 4.5. (C-D) Biochemical reactions related to production and degradation of the molecules.

Ovarian Cancer Drugs in [259]	Mechanism of Action	Target
Lapatinib (Phase I) [289]	Bind to ATP-binding site of receptor (R), preventing its autophosphorylation	RP
Sorafenib (Phase II) [383]	Bind to ATP-binding site of Raf, preventing activation of Raf	Raf*
ISIS 5132 (Phase II) [81]	Bind to Raf mRNA to downregulate Raf expression	Raf
AZD6244 (Phase II) [395]	Bind and lock MEK into inactive conformation	MEKPP
XL147 (Phase I) [257]	Bind to ATP-binding site of PI3K, preventing activation of PI3K	PI3K*
Perifosine (Phase I) [188]	Bind to lipid-binding PH domain of Akt	AktPIP, AktPIPP, AktPIP3
ECO-4601 (Phase I) [48,257]	Degrade Raf1 through proteasomal-dependent mechanism	Raf
PKI-587 (Phase I) [364]	Inhibits PI3K and mTOR kinases	PI3K*
PKI-179 (Phase I) [257]	Small-molecule mimetic of ATP that inhibits PI3K and mTOR kinases	PI3K*
BKM120 (Phase I) [257]	ATP competitive inhibitor of PI3K kinase	PI3K*
AZD5363 (Phase I) [12,257]	ATP-competitive pan-Akt inhibitor	Akt
BYL719 (Phase I, II) [257]	Specifically inhibits PI3K in the PI3K/Akt kinase signaling pathway, thereby inhibiting the activation of the PI3K signaling pathway	PI3K*
Dabrafenib (Phase I, II) [257]	Selectively binds to and inhibits the activity of B-raf, which may inhibit the proliferation of tumor cells which contain a mutated BRAF gene	Raf*
GSK1120212 (Phase I, II) [163]	Potent and selective allosteric inhibitor of MEK1/2	MEKPP
GSK2110183 (Phase I, II) [336]	ATP-competitive pan-Akt inhibitor	Akt
GSK2141795 (Phase I) [12]	ATP-competitive Akt inhibitor	Akt
MEK162 (Phase I) [257]	Non-competitive with ATP. Binds to and inhibits the activity of MEK1/2	MEKPP
MK-2206 (Phase II) [394]	Oral pan-Akt inhibitor	Akt
Pimasertib (Phase II) [257]	Selectively binds to and inhibits the activity of MEK1/2, preventing the activation of MEK1/2-dependent effector proteins and transcription factors.	MEKPP
SAR245409 (Phase II) [257]	Inhibits both PI3K kinase and mTOR kinase	PI3K*, mTOR*
Trametinib (Phase II) [257]	Specifically binds to and inhibits MEK 1 and 2	MEKPP
Triciribine (Phase I, II) [257]	Inhibits the phosphorylation, activation, and signalling of Akt-1, -2, and -3	Akt

Table 4.3: Ovarian cancer drugs in [259] and their targets in the MAPK-PI3K network [143].

ovarian cancer study while the rest involve questionnaires, radiation treatment, polycystic ovary syndrome, cytomegalovirus infection for cancer (including ovarian cancer) patients undergoing transplant or general terms such as chemotherapy. In addition, the same drug can be used in multiple studies. Hence, duplicated drugs are removed from the 989 studies, resulting in

a total of 458 unique drugs. Amongst these 458 drugs, some drugs target other networks. For instance, cisplatin targets the DNA synthesis network by crosslinking to DNA to trigger apoptosis [65]. Targets of the 458 drugs were identified using literature survey and are listed in details in [66]. Amongst these, only those that are relevant to the network are considered as reference drug targets (Tables 4.3).

Glucose-Stimulated Insulin Secretion Network. The network is associated with T2DM. Hence, we use the keywords “type 2 diabetes mellitus drug” to search the clinical trial repository [259] which yielded 5858 studies⁵. However, only 3880 studies involve the study of effects of drugs or food constituents on type 2 diabetes and these studies implicate a total of 617 unique compounds. The compounds that are relevant to this network are listed in Table 4.4 and used as reference drug targets. Note that details of these compounds are reported in [66].

4.3.2 Biological Process Related Networks

The remaining networks we studied describe specific biological processes of particular organisms. The endomesoderm network [196] describes the gene regulation of the sea urchin endomesoderm while the glucose metabolism network [192] describes the glucose-to-acetate metabolism in *Escherichia coli*.

The curation process for these networks is as follows:

- (i) Obtain a list of unique molecules (genes or proteins) relevant to the biological process of the specific organism from *PubMed* repository using specific keywords.
- (ii) Identify molecules that are within the scope of the signaling network.

⁵as of 25 January 2013.

T2DM Drugs/Food Constituents in [259]	Mechanism of Action	Target
Alcohol (Phase II) [100]	Causes hyperglycemia	glucose
Avandamet (Rosiglitazone + Metformin) (Phase IV) [162]	Rosiglitazone is a highly selective and potent agonist for the $\text{PPAR}\gamma$. Metformin decreases hepatic glucose production, decreases intestinal absorption of glucose and increases peripheral glucose uptake and utilization	activated $\text{PPAR}\gamma$, glucose
Metformin (Phase IV) [183, 366]	Inhibit respiratory-chain complex I of mitochondria, thereby reducing cellular energy transiently and activating AMPK . Increase peripheral glucose uptake and utilization	activated respiratory-chain complex I of mitochondria, glucose
Benfluorex (Phase II) [185]	Reduce β -oxidation (process by which fatty acid molecules break down in mitochondria to generate acetyl-CoA) rates and ketogenesis. Reduce gluconeogenesis from lactate/pyruvate	acetyl-CoA, glucose
Berberine (Phase III) [342]	Mimick insulin action, improve insulin action by activating AMPK , reduce insulin resistance through PKC -dependent up-regulation of insulin receptor expression, inducing glycolysis, promoting GLP-1 secretion and modulating its release, inhibition of DPP-4	insulin, activated AMPK , glucose, GLP-1 , activated DPP-4
Cod (Phase II) [200]	Increase insulin-stimulated glucose uptake	intracellular glucose
Deproteinised hemoderivative of calf blood (Actovegin) (Phase III) [221]	Actovegin is composed of small molecules present under normal physiological conditions, therefore pharmacokinetic and pharmacodynamic studies to determine its active substance are not feasible. Increase glucose uptake and improve oxygen uptake under conditions of ischemia	glucose
Tagatose (Phase III) [104]	Interfere with carbohydrate absorption by inhibiting intestinal disaccharidases and glucose transport	activated intestinal disaccharidases, plasma glucose
Vinegar [169]	Reduce plasma renin activity and aldosterone concentration. Reduce serum glucose	plasma renin, plasma aldosterone, serum glucose
Rice [157]	Increase dietary glycemic load	blood glucose
CS-917 (Phase II) [103]	Competitively inhibits fructose-1,6-bisphosphatase (FBPase) at the AMP binding site	activated FBPase
MB07803 (Phase II) [361]	Fructose-1,6-bisphosphatase (FBPase) inhibitor	activated FBPase
Glycerol [207]	Synthesized from glycerol using glycerol kinase	glycerol-3-phosphate
Methylcobalamin [115]	Provides methyl group that couples to Co to synthesize acetyl-CoA	acetyl-CoA
Gynostemma Pentaphyllum tea (Phase II) [27, 242]	Increase superoxide dismutase (SOD). Strong inhibition on IL-6 and Ptgs2 mRNA expression and weak inhibition on $\text{TNF-}\alpha$ mRNA expression. Gypenosides, the major components of Gynostemma pentaphyllum, increase Bax, reduce Bcl-2 and stimulate release of cytochrome c, AIF (apoptosis-inducing factor), and Endo G (endonuclease G) from mitochondria. Ferricytochrome c reduction to ferrocytochrome c may be required for depolarization of the mitochondrial	SOD , IL-6 , Ptgs2 , $\text{TNF-}\alpha$, Bax, Bcl-2, cytochrome c, AIF, Endo G
Xanthohumol [27, 88]	Binds annexin V-FITC, cleaves PARP-1 and activates procaspases-3, -8 and -9. Depolarizes mitochondrial leading to release of cytochrome c. Inhibits Akt, $\text{Nf}\kappa\text{B}$, mTOR , Bcl-2 and survivin. Ferricytochrome c reduction to ferrocytochrome c may be required for depolarization of the mitochondrial	cleaved PARP-1 , activated procaspase-3, -8 and -9, cytochrome c, Akt, $\text{Nf}\kappa\text{B}$, mTOR , Bcl-2, survivin

Table 4.4: T2DM drugs or food constituents in [259] and their targets in the glucose-stimulated insulin secretion network [168].

Endomesoderm Gene Regulatory Network. The network is associated with gene regulation in endomesoderm development in the sea urchin (*Strongylocentrotus purpuratus*). We performed manual curation (detailed in [69]) by searching the *PubMed* repository using “sea urchin endomesoderm” as keywords. A regulatory pathway (Figure 4.7) pertaining to the regulation of Endo16, a critical protein in the formation of the endomesoderm (see Section 4.2.3) is constructed from these publications. Nodes in this pathway are considered as targets for regulating Endo16. Details of this pathway⁶ are available in [69].

Glucose Metabolism Network. The network is associated with glucose-to-acetate metabolism in *Escherichia coli*. We performed manual curation by searching the *PubMed* repository using “E Coli glucose metabolism to acetate” as keywords. The search yielded 545 publications⁷, of which 21 were relevant literature on glucose metabolism in *Escherichia coli* under aerobic condition. We define targets in this network as genes or proteins in these literature that caused 50% or more change in acetate production when modified (knockout, knockdown or overexpressed). However, not all literature included acetate production as a measurement.

⁷as of 14 November 2013.

Target ID in BIOMD0000000244	Name	Reference
ACT	acetate	*
GLC	glucose	*
G6P	glucose-6-phosphate	[406]
ICT	isocitrate	[360]
PEP	phosphoenolpyruvate	[53, 108]
AceB	malate synthase A	[276]
Acoa2act	enzyme for reaction from ACoA to ACT	[52]
Cya	adenylate cyclase	[275]
Fdp	fructose-1,6-bisphosphatase I	[277]
Icd	unphosphorylated isocitrate dehydrogenase	[170]
Icd_P	phosphorylated isocitrate dehydrogenase	[170]
Pdh	pyruvate dehydrogenase	[250]
Ppc	phosphoenolpyruvate carboxylase	[108]
PpsA	phosphoenolpyruvate synthase	[271]
EIIA	unphosphorylated PTS protein EIIA	[167]
EIICB	PTS protein EIICB (ptsG)	[278]

Table 4.5: Targets that are crucial for acetate production in the glucose metabolism network in [192]. * indicates targets that are included by default due to their direct involvement (either as input or output) in the metabolic reaction being studied.

Hence, we also consider genes or proteins with significant difference in transcription or translation level between *Escherichia coli* strains with distinct difference in acetate production as targets. These strains are BL21 (low acetate producer) and JM109 (high acetate producer). Note that both acetate and glucose are considered targets by default since they are the input and output, respectively, of the metabolic reaction that we are interested in. Table 4.5 contains the targets relating to BIOMD0000000244 [192]. Curation details are reported in [66].

4.4 Summary

In conclusion, we described 16 structural features commonly used for identifying important nodes in the network and introduced four signaling networks with different topology and characteristics. In the next chapter, we shall study the use of these 16 structural features to perform structural feature-based target characterization for the four signaling networks.

Chapter 5

TENET: Structural Feature-Based Target Characterization

In Chapter 4, we introduced structural features and signaling networks that we shall use for characterizing targets in signaling networks. In this chapter, we discuss TENET (**T**arget **ch**aracterization using **N**etwork **T**opology), the offline component of the *Target Prioritization* subsystem, that is responsible for learning the set of structural features that best predicts targets of signaling networks and how to combine these features to obtain a *ranking score* for the purpose of target prioritization.

The rest of this chapter is organized as follows. First, we briefly introduce the challenges of the drug discovery problem and how target prioritization address some of these challenges in Section 5.1. Then, we discuss the role of network features in target prioritization, the issues related to it and the motivation of TENET in Section 5.2. Next, we formally introduce the *target characterization problem* in Section 5.3. In Section 5.4, we introduce support vector machine (SVM), a supervised machine learning technique that we use to solve the *target characterization problem*. Then, we describe the TENET algorithm in Section 5.5 and discuss experimental results in Section 5.6. Finally, we conclude the chapter in Section 5.7. A preliminary version of this work has been accepted for publication in the 2014 ACM Conference on Bioinformatics, Computational Biology and Health Informatics (ACM BCB 2014).

5.1 Introduction

Combination therapies have often been suggested as a treatment strategy for complex diseases especially when monotherapies fail to achieve the desired therapeutic effect [79]. However, the design of combination therapies is difficult due to the large number of possible targets and drug doses which results in exponential number of possible combinations to be tested. In addition, it has been observed that combination therapies do not always produce better effects than monotherapies [216]. Hence, it is important to develop approaches to enable the rapid search for superior drug combinations targeting disease-related networks. One such computational approach is *target prioritization* [58, 60, 96], which is the process of finding a set of nodes (or targets) in a given signaling network that are most likely able to achieve a user-specified goal such as modulating an output node. Recall that in this thesis, the goal is to identify target combinations. In the context of the `MAPK-PI3K` network, target combination identification finds groups of targets in the network that can modulate and reduce the activity of `ERKPP` to maintain physiological growth. Target prioritization can help to improve the search of suitable target combinations by ranking nodes according to their ability in reducing `ERKPP`'s activity. Particularly, nodes given higher ranks can be preferentially selected as targets in the combination.

5.2 Motivation

Various ways exist to prioritize targets in the literature (reviewed in Chapter 2) and they include using molecular functional annotations, network analyses, gene expression data, sequence and structures of proteins. Methods based on functional annotations are limited by annotation coverage while those based on gene expression data, sequence and structures generally focus on individual molecules and ignore intermolecular interactions which are crucial in determining the system behavior [401]. Network-based approaches, on the other hand, hold promise as

increasing number of studies strengthen the hypothesis that network topology is an essential feature in the emergent system function of the protein when it is perturbed [160, 252]. In the sequel, we focus on network-based target prioritization. In particular, we leverage network features to obtain a ranking score that can be used for ranking the targets. The network-based approach requires us to address the following questions. First, is there a single feature that can prioritize targets effectively in all signaling networks? Second, does single feature prioritize targets better than multiple features? Finally, can we simply rely only on topological features of a network?

In order to answer these questions, we analyzed some networks and found that the features that are crucial for target prioritization differ from network to network. For instance, bridging centrality, a feature that Hwang and colleagues found to identify effective drug targets with fewer off-target effects in several networks such as the yeast PPI network [160], performs sub-optimally when prioritizing targets in other networks. In particular, prioritization of nodes in the `MAPK-PI3K` network (Figure 3.1) based on bridging centrality yields an area under the receiver operating characteristics curve (ROC AUC)¹ of 0.842 while total degree centrality has ROC AUC=0.696. However, in another network [168] (Figures 4.3 and 4.4) that describes glucose-stimulated insulin secretion, the ROC AUC based on bridging centrality is 0.619 and other features such as total degree centrality performed better (ROC AUC=0.825). Moreover, different features have different effects on different nodes in the network and a single feature may not be sufficient for target prioritization. For instance, in the `MAPK-PI3K` network, we found that `AktPIP`, a known target, was given low priority by bridging centrality and high priority by proximity prestige. Another known target, `RasGTP`, however, was assigned high priority by bridging centrality and low priority by proximity prestige. Hence, target prioritization using a combination score based on both bridging centrality and proximity prestige may

¹The ROC curve plots the true positive rates (TPR) against the false positive rates (FPR) and its area is typically used for assessing the performance of a binary classifier system, with a larger area implying better performance.

provide better results than using either structural feature alone. Finally, *profile shape similarity distance* (PSSD) (Chapter 6), a non-topological feature seems to improve the ROC AUC for some networks modestly. For instance, in the glucose metabolism network [192] (Figures 4.5 and 4.6), target prioritization based on a weighted sum score involving both structural features and PSSD resulted in ROC AUC=0.94, a slight improvement over a score based on the optimal set of structural features alone (ROC AUC=0.938). Note that the slight improvement was only achieved when we set a low weight for PSSD. The benefits of incorporating the non-topological feature diminishes when PSSD was given higher weight. Hence, compared to topological features, the non-topological feature is likely to have a minor influence on target prioritization. In summary, these observations imply that using the same topological features (*e.g.*, bridging centrality) to derive the prioritization score for any network may not be effective and there is a need to devise techniques to find the optimal set of *predictive structural features* for a specific network, the impact of structural and dynamic feature on prioritization and how these features should be combined to generate a superior prioritization score to rank the targets.

In this chapter, we design techniques to identify topological features that are important for prioritizing targets and explore the interplay between optimal set of features and network characteristics for target prioritization.

5.3 Problem Statement

In this section, we first introduce the concept of a candidate target. Then, we formally define the problem of *target characterization* which involves two subproblems:

- (i) *Structural feature selection problem*: identification of a set of structural features that best predicts targets (referred to as *predictive structural features*), and
- (ii) *Structural feature-based target characterization problem*: generation of a prioritization model that ranks targets based on predictive structural features.

5.3.1 Candidate Target

Recall in Section 5.1 that a candidate target is a regulator of a given output node. In other words, if there is no path connecting a node u to the output node, then u is likely not a potential target and can be pruned from further analysis.

Definition 5.11 *Given a signaling network $G = (V, E)$ and an output node $x \in V$, let the set of nodes having a path leading to x be denoted as $V_x \subseteq V$. Then, the set of **candidate target** nodes in G relevant to x is denoted as $T_x \subseteq V_x$.*

5.3.2 Target Characterization

Intuitively, the goal of structural feature-based target characterization is to use a set of *predictive structural features* to identify known targets in a network. The known targets can be found by using curation approaches described in Chapter 4. Hence, the *target characterization problem* can be formulated as a *supervised learning problem*. In a *supervised learning problem*, a training set $\{\langle x_i, f(x_i) \rangle\}$ is given where $f(x_i)$ is the predictor of x_i and the goal is to learn some target function $f : X \rightarrow Y$ which can be applied to predict unseen data w [149]. The problem can be subdivided into two categories: regression when the predictor yields a continuous outcome and classification when the outcome is discrete [149]. A regression problem can be converted into a binary classification problem by specifying a threshold h and assigning x_i with f greater than h to one class and the remaining to the other class [355]. We advocate that the *structural feature-based target characterization problem* is best represented as a regression problem. In this problem, we are interested in finding out how likely one node is a target relative to another node based on a set of *predictive structural features*. This is different from the target classification problem where we want to find out the class membership of a node. The regression problem can be converted into a classification problem by specifying a threshold h

and assigning nodes having target function greater than h to the target class and the rest to the non-target class.

As mentioned in Section 5.2, this study examines a total of 16 structural features. However, as we shall see later, some features may be irrelevant to the network and can have a negative impact on the performance of the prediction model [106]. Hence, it is important to learn the set of *predictive structural features* that best characterizes targets (referred to as *structural feature selection*).

Definition 5.12 Given a signaling network $G = (V, N)$ and an output node $x \in V$, let T_x and R denote the set of known targets in G relevant to x , and the set of structural features of G , respectively, such that $T_x \subseteq V$. Then, **structural feature selection** finds a subset of features $S \subseteq R$ that maximizes the prediction accuracy for

$$f(\xi(u, S)) \quad (5.1)$$

subject to the conditions

$$\begin{cases} f(\xi(u, S)) = 1 & \text{when } u \in T_x, \\ f(\xi(u, S)) = 0 & \text{otherwise.} \end{cases} \quad (5.2)$$

The subset of features S is also known as the set of **predictive structural features**.

The predictive structural features in S are then used to generate a *prioritization model* (or function) for prioritizing targets.

Definition 5.13 Given a signaling network $G = (V, N)$, an output node $x \in V$ and a set of predictive structural features S , let T_x denote the set of known targets in G relevant to x , such that $T_x \subseteq V$. Then, for a threshold h , **structural feature-based target characterization** learns a *prioritization model*

$$g(\xi(u, S)) \quad (5.3)$$

subject to the conditions

$$\begin{cases} g(\xi(u, S)) \in \mathfrak{R}, \\ g(\xi(u, S)) \geq h & \text{when } u \in T_x, \\ g(\xi(u, S)) < h & \text{otherwise.} \end{cases} \quad (5.4)$$

Note that in Definition 5.13, there is no need to explicitly specify a threshold h if we are only interested in obtaining the relative rankings of the nodes. The threshold is required if we want to assign class labels (*e.g.*, target class) to the nodes.

5.4 Support Vector Machine SVM

Recall that the target characterization problem can be formulated as a supervised learning problem. A variety of approaches exist to solve supervised learning problems, ranging from decision trees [308] to Bayesian networks [121] to support vector machines (SVM) [363]. We choose a SVM-based approach for the following reasons. First, SVM-based methods are more robust for problems with large feature set and small sample size data [158]. Note that TRECENTO requires signaling networks with fully defined kinetic information (*e.g.*, ordinary differential equations (ODE)) as inputs for assessing the effect of candidate target combinations in the *Target Combination Finder* subsystem. Such networks tend to be small, each with at most tens to hundreds of nodes, since it is technically challenging to obtain the kinetic parameters for these ODEs [305]. Second, SVM-based approaches perform well for problems with sparse and noisy data [122]. Signaling networks that we use as inputs to TENET are inherently noisy [21]. Third, SVM-based methods perform well even when input features and output responses have a nonlinear relationship [346]. Hence, they are well suited for our problem.

In this section, we first provide an overview of SVM. Then, we describe the SVM training and parameter tuning process. Finally, we discuss several issues that affect the performance of SVM and examine how *structural feature selection* and *cost sensitive learning* can improve prediction accuracy.

5.4.1 An Overview

SVM is a supervised learning method which has its origins in the *Generalized Portrait* algorithm developed by Vapnik et al. [363] for empirical risk minimization. Although originally developed to solve classification problems [44, 45, 283], it has recently been extended to solve regression problems as well [71, 316, 330]. In the sequel, we use the terms *support vector classification* (SVC) and *support vector regression* (SVR) to differentiate the two types of problems. In SVC, the goal is to classify a binary-class data by specifying an appropriate hyperplane that best separates the two groups of data [9], whereby the margin between the nearest members of the opposing classes is maximized. Non-linearity in the data is addressed by choosing a suitable kernel function which transforms the input space $\mathbb{R}$ to a higher dimensional space $\mathbb{F}$ where linear separation is achieved [9]. Figure 5.1 illustrates the use of such a kernel. SVM can be extended to solve regression problems by incorporating an alternative loss function usually in the form of the ϵ -insensitive loss function [136]. This loss function is preferred over others such as quadratic or Laplacian as it is less sensitive to outliers, works well for data with unknown distribution and allows for sparseness in the support vectors [136]. In SVR, the goal is to find a function that approximates a set of data and allows a maximum ϵ deviation from the actual target for all the training data [136]. That is, a maximum loss of ϵ is allowed.

SVC: Linearly Separable Case. Let $T = \{(\mathbf{x}, y)\}$ be a linearly separable binary-class training data where $\mathbf{x} \in \mathbb{R}^n$ and $y \in \{-1, +1\}$. The hyperplane H is defined by the equation $\mathbf{w}^T \mathbf{x} + b = 0$ where $\mathbf{w}$ is the weight vector that is perpendicular to H and b is the bias. The two classes of data are specified by the following pair of equations

$$\begin{cases} \mathbf{w}^T \mathbf{x} + b \geq +1, & \text{if } y=+1 \\ \mathbf{w}^T \mathbf{x} + b \leq -1, & \text{if } y=-1 \end{cases} \quad (5.5)$$

$$\quad \quad \quad (5.6)$$

where Equation 5.5 defines hyperplane H_1 and the circle-containing class while Equation 5.6 defines hyperplane H_2 and the square-containing class. The equations can be further combined

Figure 5.1: Support vector classification. The filled shapes lying on the hyperplanes H_1 and H_2 are the support vectors.

into a single equation [44]

$$y(\mathbf{w}^T \mathbf{x} + b) \geq 1. \quad (5.7)$$

The margin ($\frac{2}{\|\mathbf{w}\|}$) in Figure 5.1 is the distance between the nearest member in the circle and in the square classes which are denoted by $\mathbf{x}_+$ and $\mathbf{x}_-$, respectively [44]. The optimal hyperplanes for separating the data is found by maximizing this margin, subject to the constraint specified by Equation 5.7. A slack variable τ is typically introduced to handle situations of imperfect separation [136] that may occur in real-world problems. The optimization problem is then reformulated as

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \tau_i \\ & \text{subject to} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \tau_i. \end{aligned} \quad (5.8)$$

where the cost parameter C controls the trade-off between misclassification of training data and enforcement of a rigid margin [159]. This constrained optimization problem is typically solved using the Lagrangian multiplier method [44].

svc: Nonlinearly Separable Case. For nonlinearly separable data, the input space can be mapped onto a feature space where the data becomes linearly separable. The mapping is

Kernel	Formula	Parameters
Linear	$u^T v$	-
Polynomial	$\gamma(u^T v + c_0)^d$	γ, d, c_0
Radial Basis Function (RBF)	$e^{-\gamma u-v ^2}$	γ
Sigmoid	$\tanh(\gamma u^T v + c_0)$	γ, c_0

Table 5.1: SVM kernel types and their associated parameters [238].

Figure 5.2: Support vector regression. The filled shapes lying on the hyperplanes H_1 and H_2 are the support vectors.

achieved using a kernel function ($K(\mathbf{x}, y)$) like the one found in Figure 5.1. Table 5.1 lists some of the common kernels used. The choice of kernels depends on the data. For instance, the radial basis function (RBF) kernel is suited for data that has nonlinear relationship with its class labels [155]. We shall examine all four kernels in Table 5.1 as we do not have any prior knowledge on the type of relationship present between our data and its class labels.

SVR. Figure 5.2 illustrates the regression problem which is very similar to SVC. A key difference is that SVR predicts a continuous outcome compared to SVC which predicts a discrete outcome [149]. In SVC, a penalty of $C \sum_i \rho_i$ is imposed when a data point i is on the wrong side of the hyperplane (Figure 5.1) and correctly labelled data points are not penalized. A similar notion is present in SVR. A penalty is imposed if the data point is far away from the hyperplane H and points that are close enough (within a distance of ϵ) incur no penalty.

The optimization problem is reformulated as

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i (\tau_i + \tau_i^*) \\ & \text{subject to} && \begin{cases} y_i - \mathbf{w}^T \mathbf{x}_i - b \leq \varepsilon + \tau_i \\ \mathbf{w}^T \mathbf{x}_i + b - y_i \leq \varepsilon + \tau_i^* \\ \tau_i, \tau_i^* \geq 0 \end{cases} \end{aligned} \quad (5.9)$$

where C is the cost parameter controlling the trade-off between the flatness of the function defining H and the maximum deviation from ε that is tolerated [330]; and τ^* and τ are the orthogonal distances away from H_1 and H_2 , respectively.

5.4.2 SVM Training and Parameter Tuning

As we have seen in the previous subsection, SVM models contain multiple parameters that affect the learning and performance of the models [54]. These parameters have to be tuned in order to minimize the generalization error which can be estimated by testing on some unseen data using techniques such as cross-validation. The parameters that are tuned when training a SVM are usually the cost parameter C and the parameters associated with the kernel function [159]. We follow the procedure specified in [155] for training the SVM. The steps are as follows:

- (i) Perform data scaling.
- (ii) Find the best model parameter using cross-validation.
- (iii) Train the whole training set using the best model parameter.
- (iv) Evaluate the performance of the model using the test set.

We scale the feature values linearly to the range of $[0, 1]$ for each signaling network. This is to avoid features with larger ranges dominating those with smaller ranges. As mentioned in Section 5.4.1, we shall study a variety of kernels and use stratified cross-validation and grid-search on the training data to identify the best model parameter. The use of cross-validation

helps to avoid the issue of overfitting the data and stratification helps to keep the percentage of targets in the different folds similar to the original dataset. In a n -fold cross-validation, the training data is divided into n portions of equal sizes, of which $n - 1$ portions are used for training the SVM while one portion (validation set) is used for testing the model. This process is repeated for n times such that each portion is used exactly once for testing. The best parameter is the one that yields the best average prediction accuracy for the cross-validation process. Where possible, we use a 10-fold stratified cross-validation since larger fold numbers reduce pessimistic bias and 10 folds generally gave good performances [184]. However, in certain networks, the target class size may be small and there may not be sufficient targets to spread out amongst the different folds. Hence, the number of folds applied to a particular network will be either 10 (default) or the target class size minus one, whichever is smaller, since we require at least one target in the test set. The grid-search proposed in [155] starts off with a coarse grid-search to identify a potential region where the best parameter may lie and then switch to fine grid-search in this prospective region. This search approach was found to work well for moderate size problems (thousands or more data points) [155]. We expect this approach to perform well since our problem size is much smaller.

5.4.3 Challenges of Training the SVM Model

Several non-trivial issues, namely, irrelevant or redundant features, noisy labels and imbalanced data set, need to be addressed in training the SVM model for prioritizing targets. We shall now discuss each of these in turn.

First, the SVM can perform poorly if features used to train the model are irrelevant. In [106], the authors performed classification using a one class SVM for various features on artificial data and found that irrelevant features added to the data resulted in poor classification.

Next, the SVM training task relies on the correctness of the class labeling to generate a model. In our problem, the target class labeling is generated via manual curation. In the case

of the non-target class, a node is assigned as a non-target if the curation process produces no literature with evidence that supports the node acting as a target. However, in some cases, nodes in the non-target class may in fact be targets in reality. Such discrepancies could be due to several reasons such as

- (i) literature with supporting evidence of a node as a target that are unpublished, or
- (ii) literature with supporting evidence of a node as a target that are published, but are not indexed in the *PubMed* repository, or
- (iii) on-going research on potential target nodes.

Hence, there is an element of uncertainty in the labeling (noisy label) which can affect the resulting model. For instance, in the `MAPK-PI3K` signaling network, Table 4.3 contains all the nodes with available information supporting their role as targets. Obviously, it is possible that nodes missing from this table may in reality be targets if supporting evidences are found subsequently.

Finally, the problem of imbalanced data set is due to the relatively smaller size of the target class compared to the non-target class in many of the networks. Table 4.2 summarizes the number of targets present in each network. The percentage of targets in each network ranges from 10.2% to 34.0%. The imbalanced data problem happens when the class of interest is the minority class. A high prediction accuracy classifier may be skewed towards correct prediction mainly in the majority class [286] and have substantial misclassification of data in the minority class. As we shall see later (Section 5.6.2), the sensitivity and precision of the SVM models are generally poor as the target class size is much smaller compared to the non-target class size.

In Section 5.6.2, we examine how these challenges can be alleviated by using approaches such as *feature selection* and *cost sensitive learning*. We shall now briefly discuss work related to these two approaches. Interested readers may refer to [307] for a review of *feature selection*

approaches applicable to bioinformatics and [211] for a review of *cost sensitive learning* approaches.

Feature Selection. Feature selection is used frequently in model building for a faster and more accurate model and also to avoid overfitting a model [307]. Unlike other dimension reduction methods (*e.g.*, principal component analysis [334] and information theory [97]), where the original data undergoes transformation such as projection and compression, feature selection methods retain the data in its original form but selects a subset of it. Hence, the original semantics is not lost and lends itself to interpretation by experts in the field [307].

Feature selection methods for supervised learning (classification) can be broadly categorized into two classes based on their interaction with the classifier: *classifier-aware* and *classifier-independent*. Compared to classifier-independent methods, classifier-aware methods interact with the classifiers and such interaction can lead to better classification results [307]. However, they are typically computationally expensive and run the risk of model-overfitting.

Examples of classifier-aware approaches include optimization-based methods [210, 212, 227] and penalization-based methods [219]. The former methods use optimization strategies such as greedy optimization [210, 227] or particle swarm optimization [212] to select features whereas the latter methods use a penalty function such as Lasso and features are selected to minimize the penalty function. The key difference between these two types of methods is that the former methods focus on how optimization is performed whereas the latter methods focus on what is being optimized. Classifier-independent approaches, on the other hand, typically use some scoring methods to weigh the features and select for features that are above a certain threshold [397]. Statistical measures such as correlation [140, 307] are commonly used for generating feature weights.

In this chapter, we shall explore three feature selection approaches, namely, backward stepwise elimination [227] (classifier-aware), Wilcoxon-ROC based elimination (classifier-

independent) and WRE-BSE (hybrid approach). Details of these approaches are found in Section 5.5.1.

Cost Sensitive Learning. *Cost sensitive learning* has been proposed to address issues such as data imbalance [4, 99, 344, 350] and noisy labels [187, 213]. In particular, cost sensitive learning is an algorithmic approach that chooses an appropriate strategy specific to the classifier to overcome the bias introduced by imbalanced data. This is different from resampling-based techniques that resamples the data to achieve an optimal class distribution in order to address the data imbalance issue [344]. Note that the determination of the optimal class distribution is a challenge since it differs across data sets [344]. Cost sensitive learning approaches can be broadly classified into *threshold-based* and *sampling-based* approaches.

Threshold-based approaches require classifiers to produce probability estimates and then performs classification by applying a threshold on the probability estimates. MetaCost [94] and Empirical Thresholding [321] are examples of these approaches. MetaCost is a four-step approach. First, it performs bootstrapping to generate multiple subset of training set and learns a classifier on each subset. Then, it estimates the probability of each class using bagging on decision trees. Next, it relabels the training data using the probability estimates. Finally, the relabeled training data is used to construct a cost sensitive classifier. Unlike MetaCost, Empirical Thresholding uses cross-validation to identify the threshold probability which is subsequently used to perform classification.

Sampling-based approaches such as costing [400] and instance-weighting [353] use different strategies to modify the class distribution of training data. In particular, costing uses proportional sampling with replacement and rejection sampling to re-sample the data before performing classification. Instance-weighting [353], on the other hand, uses weight allocation strategies to improve the classifier. Specifically, it manipulates misclassification cost to modify class distribution and achieves an effect similar to classical sampling approaches such as costing. The smaller class (that is also the class of interest) is given a higher misclassification cost

(or higher weights) that is proportional to the size of the class and the objective of the classifier is to minimize the overall cost. Unlike these weight-based approaches, classical sampling-based approaches use a subset of the original training data for generating the classifier. Hence, some information may be lost and the classifier performance may be affected.

In this thesis, we use *weighted misclassification cost*, an approach similar to instance-weighting, that proportionates the misclassification cost of the training data according to class to address issues of data imbalance and noisy labels. The effects of varying the misclassification cost shall be examined in Section 5.6.2. Note that the key difference between *weighted misclassification cost*, our proposed approach, and instance-weighting is that in our proposed approach, the weights of the misclassification cost is learnt using various signaling networks. That is, the weights need not be proportional to the class size.

5.5 Target Characterization

In this section, we present TENET, a SVM-based approach for characterizing targets using structural features. First, we provide an overview of TENET. Then, we describe the TENET algorithm and analyze its complexity

5.5.1 Overview of TENET

TENET identifies the set of predictive structural features that characterizes known targets and a prioritization model based on these features for a given signaling network. First, it identifies relevant nodes (denoted as $V_{candidate}$) in the networks that shall be used for training the SVM by removing nodes that are unable to influence the output node. This step is intended to reduce the computations required. Then, the entire set of structural features that are intended for characterizing known targets is computed for all nodes in $V_{candidate}$. Finally, TENET uses SVM to learn the structural features that are predictive of known targets. We choose SVM over other supervised machine learning techniques for the reasons stated in Section 5.4. The SVM model

that is trained functions as a prioritization model to facilitate target prioritization subsequently. In particular, TENET uses structural feature selection to help eliminate redundant features and weighted misclassification cost to address issues of imbalance classes and noisy labels.

In this chapter, we study several feature selection methods and examine their effect on prioritization results. Specifically, these methods are

- (i) classifier-aware approach: **B**ackward **S**teppwise **E**limination (BSE),
- (ii) classifier-independent approach: **W**ilcoxon and **R**OC analysis-based **E**limination (WRE),
and
- (iii) hybrid approach: WRE-BSE, which performs WRE followed by BSE.

The BSE approach [227] uses a simple greedy optimization approach to select for features and has been found to be efficient in applications that analyze metabolome data [210]. It progressively removes features from the naïve SVM model (built using all available features) and builds a new best model after each feature removal (using the training and parameter tuning approach described in Section 5.4.2). The elimination process stops when removal of additional features result in a worse average accuracy of the validation set prediction.

The WRE approach, on the other hand, uses results of statistical tests to filter out features that do not discriminate between targets and non-targets. Note that although we do not have prior knowledge of the feature distribution, we know that the data is binary-class (target class and non-target class) and the size of the classes is likely to be different. In addition, some nodes may have the same rank for certain features. We use the Wilcoxon Rank-Sum (Wilcoxon) test² [226] and receiver operating characteristic (ROC) analysis [110] because they are non-parametric, handle tied ranks and allow the target and non-target class sizes to be different

²In the 1-tailed Wilcoxon test for two given distributions A and B , the null hypothesis $H_0 : A = B$ is tested against either of the alternative hypotheses $H_1 : A \geq B$ or $H_2 : A \leq B$. Let n_A be the number of observations in A , w_A be the sum of the ranks for observations from A and W_A be the corresponding random variable. The p -value is given as $p = Pr(W_A \geq w_A)$ or $p = Pr(W_A \leq w_A)$ which indicate the probability of seeing a value greater than or equals, or less than or equals to the observed w_A , respectively.

Structural Feature	MAPK-PI3K	Glucose-Stimulated Insulin Secretion	Endomesoderm Gene Regulatory	Glucose Metabolism
$\delta(\mathbf{A})$	9.62×10^{-4} (0.999) [†]	0.934 (0.069)	1.77×10^{-5} (1) [†]	0.770 (0.237)
$\delta(\mathbf{T})$	0.004 (0.996) [†]	0.936 (0.068)	7.81×10^{-5} (1) [†]	0.852 (0.154)
$\zeta(\mathbf{A})$	8.83×10^{-4} (0.999) [†]	0.173 (0.833)	3.34×10^{-11} (1) [†]	0.889 (0.115)
$\zeta(\mathbf{T})$	0.002 (0.998) [†]	0.186 (0.823)	7.38×10^{-10} (1) [†]	0.887 (0.119)
$\pi(\mathbf{A})$	0.095 (0.911)	0.018 (0.983) [†]	0.088 (0.912)	0.923 (0.078)
$\pi(\mathbf{T})$	0.062 (0.943)	0.024 (0.978) [†]	0.108 (0.892)	0.932 (0.072)
$\beta(\mathbf{A})$	1.76×10^{-4} (1) [†]	0.993 (0.007) [†]	0.941 (0.059)	0.631 (0.377)
$\beta(\mathbf{T})$	5.66×10^{-4} (1) [†]	0.980 (0.022) [†]	0.936 (0.064)	0.751 (0.258)
$\kappa_{\text{cyc}}(\mathbf{A})$	0.414 (0.601)	0.976 (0.025) [†]	0.764 (0.236)	0.397 (0.614)
$\kappa_{\text{cyc}}(\mathbf{T})$	0.519 (0.5)	0.965 (0.038) [†]	0.775 (0.225)	0.531 (0.482)
$\vartheta(\mathbf{A})$	0.015 (0.986) [†]	0.917 (0.088)	0.994 (0.006) [†]	0.399 (0.616)
$\vartheta(\mathbf{T})$	0.016 (0.986) [†]	0.889 (0.119)	0.995 (0.005) [†]	0.394 (0.623)
$\alpha(\mathbf{A})$	0.058 (0.946)	0.999 (0.001) [†]	1.75×10^{-7} (1) [†]	0.790 (0.217)
$\alpha(\mathbf{T})$	0.120 (0.888)	0.994 (0.007) [†]	2.09×10^{-6} (1) [†]	0.795 (0.213)
$\kappa_{\text{in}}(\mathbf{A})$	0.297 (0.717)	0.876 (0.130)	0.942 (0.058)	0.461 (0.550)
$\kappa_{\text{in}}(\mathbf{T})$	0.472 (0.547)	0.959 (0.044)*	0.947 (0.053)	0.424 (0.588)
$\theta_{\text{in}}(\mathbf{A})$	0.718 (0.295)	0.983 (0.018) [†]	0.062 (0.938)	0.731 (0.279)
$\theta_{\text{in}}(\mathbf{T})$	0.725 (0.291)	0.983 (0.018) [†]	0.079 (0.921)	0.831 (0.177)
$\kappa_{\text{mid}}(\mathbf{A})$	0.445 (0.571)	0.976 (0.025) [†]	$1 (2.80 \times 10^{-8})$ [†]	0.059 (0.944)
$\kappa_{\text{mid}}(\mathbf{T})$	0.556 (0.463)	0.965 (0.038) [†]	$1 (1.09 \times 10^{-7})$ [†]	0.092 (0.913)
$\kappa_{\text{out}}(\mathbf{A})$	0.545 (0.470)	0.764 (0.246)	0.682 (0.319)	0.489 (0.523)
$\kappa_{\text{out}}(\mathbf{T})$	0.555 (0.464)	0.666 (0.349)	0.739 (0.261)	0.674 (0.339)
$\theta_{\text{out}}(\mathbf{A})$	0.006 (0.994) [†]	0.997 (0.003) [†]	5.54×10^{-5} (1) [†]	0.683 (0.326)
$\theta_{\text{out}}(\mathbf{T})$	0.012 (0.989) [†]	0.989 (0.012) [†]	3.42×10^{-4} (1) [†]	0.768 (0.241)
$\mu(\mathbf{A})$	0.011 (0.990) [†]	0.993 (0.007) [†]	0.549 (0.451)	0.832 (0.174)
$\mu(\mathbf{T})$	0.029 (0.973) [†]	0.980 (0.022) [†]	0.539 (0.461)	0.907 (0.097)
$\sigma(\mathbf{A})$	0.961 (0.042)*	0.054 (0.949)	0.983 (0.017) [†]	0.071 (0.932)
$\sigma(\mathbf{T})$	0.908 (0.099)	0.093 (0.913)	0.979 (0.021) [†]	0.031 (0.971)*
$\theta_{\text{total}}(\mathbf{A})$	0.033 (0.969)*	0.996 (0.004) [†]	3.64×10^{-4} (1) [†]	0.586 (0.424)
$\theta_{\text{total}}(\mathbf{T})$	0.078 (0.928)	0.992 (0.009) [†]	0.001 (0.999) [†]	0.726 (0.284)
$\kappa_{\text{undir}}(\mathbf{A})$	0.994 (0.007) [†]	0.008 (0.993) [†]	$1 (1.84 \times 10^{-7})$ [†]	0.105 (0.899)
$\kappa_{\text{undir}}(\mathbf{T})$	0.980 (0.022) [†]	0.011 (0.990) [†]	$1 (3.77 \times 10^{-6})$ [†]	0.080 (0.924)

Table 5.2: Wilcoxon test of the signaling networks. Each cell entry contains the p -value in the form of xxx(yyy) where xxx and yyy denote the p -values for the alternative hypotheses where the features of the known target nodes are smaller and greater than the non-target nodes, respectively. Note that $p(\mathbf{A})$ and $p(\mathbf{T})$ denote feature p when applied to the entire node set and the training node set, respectively.

[251]. Note that the ROC analysis is a standard test used for evaluating performance of binary classifiers [110]. In particular, the Wilcoxon test determines if individual feature is able to significantly discriminate ($p\text{-value} \leq 0.05$) a known target from the rest of the nodes, while the ROC analysis examines the performance of binary classification using individual feature. We use the area under the ROC curve (ROC AUC) to assess the classification performance. An area in the range of [0.7, 0.8] and [0.8, 0.9] indicate acceptable and excellent performances, respectively [153]. The p -values and the ROC AUC of these signaling networks are found in Tables 5.2 and 5.3, respectively. As we shall see later, when training the SVM model, the node set of each signaling network is divided into a test node set and a training node set. In Tables 5.2 and 5.3, $p(A)$ and $p(T)$ denote feature p when applied to the entire node set and the training node set, respectively. Table entries marked with * are only significant in either $p(A)$ or $p(T)$ while those marked with † are significant in both. In general, features that can discriminate known targets from other nodes in a significant manner correlate well for $p(A)$ and $p(T)$ except for a few exceptions that are marked in *. In practice, it is likely that we will know the targets for the training data, but not for the test data. Hence, we shall use the results of $p(T)$ for selecting the features. Intuitively, we want to eliminate features that do not discriminate between targets and non-targets in a significant manner (using Wilcoxon test) and that do not classify targets well (using ROC analysis). In other words, we seek to exclude features that perform poorly for both the Wilcoxon test and the ROC analysis. Note that we perform two 1-tailed Wilcoxon tests and for each test, p -values smaller than 0.05 are considered significant. Hence, we take the difference of the p -values for both test hypotheses (referred to as p -value difference) and remove features with p -value difference less than 0.9. For the ROC analysis, features with AUC less than 0.7 are considered poor performers and are removed.

Note that compared to the BSE approach, which is a form of greedy optimization, the WRE approach functions more like a screening approach for features. It would be interesting to

Structural Feature	MAPK-PI3K	Glucose-Stimulated Insulin Secretion	Endomesoderm Gene Regulatory	Glucose Metabolism
$\delta(\mathbf{A})$	0.838 [†]	0.687	0.6	0.566
$\delta(\mathbf{T})$	0.807 [†]	0.707*	0.596	0.599
$\zeta(\mathbf{A})$	0.842 [†]	0.619	0.662	0.609
$\zeta(\mathbf{T})$	0.831 [†]	0.625	0.658	0.615
$\pi(\mathbf{A})$	0.644	0.763 [†]	0.467	0.629
$\pi(\mathbf{T})$	0.679	0.773 [†]	0.468	0.642
$\beta(\mathbf{A})$	0.89 [†]	0.807 [†]	0.461	0.529
$\beta(\mathbf{T})$	0.877 [†]	0.78 [†]	0.46	0.563
$\kappa_{cyc}(\mathbf{A})$	0.523	0.717 [†]	0.516	0.478
$\kappa_{cyc}(\mathbf{T})$	0.502	0.716 [†]	0.517	0.505
$\vartheta(\mathbf{A})$	0.729 [†]	0.649	0.553	0.485
$\vartheta(\mathbf{T})$	0.742 [†]	0.645	0.557	0.483
$\alpha(\mathbf{A})$	0.673	0.879 [†]	0.626	0.572
$\alpha(\mathbf{T})$	0.638	0.843 [†]	0.62	0.578
$\kappa_{in}(\mathbf{A})$	0.554	0.623	0.531	0.492
$\kappa_{in}(\mathbf{T})$	0.49	0.705*	0.534	0.483
$\theta_{in}(\mathbf{A})$	0.556	0.755 [†]	0.467	0.545
$\theta_{in}(\mathbf{T})$	0.56	0.777 [†]	0.468	0.574
$\kappa_{mid}(\mathbf{A})$	0.515	0.717 [†]	0.628	0.629
$\kappa_{mid}(\mathbf{T})$	0.512	0.716 [†]	0.628	0.619
$\kappa_{out}(\mathbf{A})$	0.49	0.575	0.51	0.497
$\kappa_{out}(\mathbf{T})$	0.488	0.548	0.515	0.534
$\theta_{out}(\mathbf{A})$	0.758 [†]	0.832 [†]	0.587	0.54
$\theta_{out}(\mathbf{T})$	0.744 [†]	0.802 [†]	0.581	0.566
$\mu(\mathbf{A})$	0.75 [†]	0.807 [†]	0.503	0.585
$\mu(\mathbf{T})$	0.72 [†]	0.78 [†]	0.502	0.626
$\sigma(\mathbf{A})$	0.688	0.701*	0.551	0.629
$\sigma(\mathbf{T})$	0.65	0.682	0.551	0.675
$\theta_{total}(\mathbf{A})$	0.696	0.825 [†]	0.58	0.518
$\theta_{total}(\mathbf{T})$	0.662	0.825 [†]	0.576	0.554
$\kappa_{undir}(\mathbf{A})$	0.767 [†]	0.802 [†]	0.618	0.614
$\kappa_{undir}(\mathbf{T})$	0.732 [†]	0.814 [†]	0.61	0.636

Table 5.3: Area under the ROC curves of the signaling networks. Note that $p(\mathbf{A})$ and $p(\mathbf{T})$ denote feature p when applied to the entire node set and the training node set, respectively.

find out if WRE or an optimized form of WRE can perform better than BSE. A simple way to optimize WRE is to first perform WRE followed by BSE (WRE-BSE). Hence, we include this hybrid approach in our study.

Apart from feature selection, in this chapter, we use weighted misclassification cost, a cost-sensitive learning approach that proportionates the misclassification cost of the training data

according to class to address both data imbalance and noise and examine the effect of varying the misclassification cost. Note that Equations 5.8 and 5.9 remain unchanged except for the fixed cost parameter C which is substituted as a variable C_i as follows:

$$C_i = \begin{cases} C^+ & \text{if } y_i = +1 \\ C^- & \text{if } y_i = -1 \end{cases} \quad (5.10)$$

subject to the following constraints:

$$\begin{cases} C^+ + C^- = 1, \\ C^+ > 0, \\ C^- > 0, \end{cases}$$

where C^+ and C^- denote the misclassification cost of the target and non-target classes, respectively.

5.5.2 The Algorithm TENET

The TENET algorithm is outlined in Algorithm 1. Given a signaling network $G = (V, E)$, an output node x , a set of targets T_x that is relevant to x , a set of features to be studied $\mathcal{X}_{all}$ and the step-size of the misclassification cost s , TENET produces a prioritization model $\mathcal{M}$ and its associated set of predictive structural features $\mathcal{F}$. Note that $\mathcal{X}_{all}$ and s are optional inputs which will be set to default values³ if they are not given. In particular, the targets T_x of the signaling networks we study in this chapter are known and is used for learning $\mathcal{F}$ and $\mathcal{M}$. TENET achieves this through the following steps.

First, in Line 2, it filters out nodes that do not have paths leading to the output node x to yield a set of potential candidate nodes $V_{candidate} \subset V$ using the FILTERCANDIDATE procedure (outlined in Algorithm 2). Then, for each candidate node, it extracts the structural features in $\mathcal{X}_{all}$ and store it in a $|V_{candidate}| \times |\mathcal{X}_{all}|$ matrix H (Line 3). Finally, in Line 4, TENET learns the optimal set of predictive structural features $\mathcal{F}$ and the best model parameters of the prioritization model $\mathcal{M}$ for prioritizing the targets using procedure LEARNFEATURESANDSETTINGS

³ $\mathcal{X}_{all}$ will be set to the 16 structural features in Section 4.1 whereas s will be set to 0.1.

Algorithm 1: Algorithm TENET

Input: Signaling network G , output node x , target set T_x , structural feature set $\mathcal{X}_{all}$ (optional) and step-size of the misclassification cost s (optional).

Output: Predictive structural feature set $\mathcal{F}$ and prioritization model $\mathcal{M}$.

- 1 $\mathcal{F}, \mathcal{M}, \mathcal{X}_{all} \leftarrow \text{INITIALIZE}(\mathcal{F}, \mathcal{M}, \mathcal{X}_{all})$
 - 2 $V_{candidate} \leftarrow \text{FILTERCANDIDATE}(G, x)$
 - 3 $H \leftarrow \text{EXTRACTALLFEATURES}(G, V_{candidate}, \mathcal{X}_{all})$
 - 4 $\mathcal{F}, \mathcal{M} \leftarrow \text{LEARNFEATURESANDSETTINGS}(H, T_x, s)$
-

Algorithm 2: Algorithm FILTERCANDIDATE

Input: Signaling network G , output node x .

Output: Candidate target node set $V_{candidate}$.

- 1 $G_{bi} \leftarrow \text{CONVERT2BIPARTITEGRAPH}(G)$
 - 2 $G_{DAG} \leftarrow \text{CONVERT2DAG}(G_{bi})$
 - 3 $U \leftarrow \text{GETROOTNODES}(G_{dag})$
 - 4 **foreach** iteration $i=1$ to $|U|$ **do**
 - 5 $G_{dag} \leftarrow \text{INDEX}(G_{dag}, U_i, \text{null})$
 - 6 $V_{candidate} \leftarrow \text{ASSESSREACHABILITY}(G_{dag}, x)$
-

(outlined in Algorithm 5).

FILTERCANDIDATE Procedure. First, we discuss the node-filtering step outlined in Algorithm 2. The algorithm consists of four key steps, namely bipartite graph conversion (Line 1), directed acyclic graph (DAG) conversion (Line 2), DAG indexing (Line 5) and reachability-based filtering (Line 6). Conceptually, nodes positioned along a parallel path or downstream of the output node do not influence it and can be pruned. Note that retroactivity phenomenon in *reduced signaling network models* (referred to as *reduced models*) may result in a downstream node influencing an upstream node [180]. The reduced models are created by further lumping of reactions in original models which are constructed from experimental data fitting. In this thesis, we use published models constructed directly from experimental data fitting and do not perform further network reduction on these models prior to analysis.

The first two steps in Algorithm 2 preprocess the input hypergraph into a DAG which has a consistent topological ordering, making indexing of the DAG easier subsequently for

Figure 5.3: (a) Hypergraph (adapted from Figure 3.1), (b) Bipartite graph representation of (a), and (c) DAG representation of (b).

Algorithm 3: Algorithm CONVERT2DAG

Input: Bipartite graph $G_{bi} = (V_{bi}, E_{bi})$.

Output: Directed acyclic graph (DAG) G_{dag} .

```

1  $G_{dag} \leftarrow G_{bi}$ 
2  $G_{dag}.SCC \leftarrow \text{GETSCC}(G_{dag})$ 
3 foreach iteration  $i=1$  to  $|G_{dag}.SCC|$  do
4    $V \leftarrow \text{INSERTNODE}(V, v_{meta:i})$ 
5    $X \leftarrow \text{GETNODESINSCC}(G_{dag}.SCC_i)$ 
6   foreach iteration  $j=1$  to  $|X|$  do
7      $N \leftarrow \text{GETNEIGHBOURSNOTINSCC}(X_j, X)$ 
8     foreach iteration  $k=1$  to  $|N|$  do
9        $E \leftarrow \text{REPLACEEDGE}(E, (X_j, N_k), (v_{meta:i}, N_k))$ 
10       $E \leftarrow \text{REPLACEEDGE}(E, (N_k, X_j), (N_k, v_{meta:i}))$ 
11       $V \leftarrow \text{REMOVENODE}(V, X_j)$ 

```

reachability check. In the bipartite graph ($G_{bi} = (V_{bi}, E_{bi})$) conversion phase (Line 1), we adopt the method in [102] which converts a hyperedge (U, W) to a set of bipartite edges $\{(u_i, q), \dots, (q, w_j)\} \subset E_{bi}$ where $u_i \in U$, $w_j \in W$, $V_{bi} = P \cup Q$, $P = U \cup W$ and $q \in Q$ by connecting the incoming node u_i and outgoing node w_j via node q . For example, in Figure 5.3a, hyperedge e_1 is converted to its bipartite form by adding a virtual node q_1 and connecting all its incoming and outgoing nodes to q_1 (See Figure 5.3b).

In the DAG conversion step (Algorithm 3), we adopt the approach in [351] to identify SCCs (Line 2) and replace each SCC with a meta node $v_{meta:i}$ (Lines 3 to 11). For example, in Figure 5.3, the $SCC = \{ERKPP, ERKP, ERK\}$ is replaced with $v_{meta:A}$ and the edges are reconfigured as $(MEKPP, v_{meta:A})$ and $(MKP3, v_{meta:A})$.

The DAG indexing step (Algorithm 4) facilitates efficient assessment of node reachability

Figure 5.4: Indexed directed acyclic graph representation of Figure 3.1 with strongly connected component and non-spanning tree edge table. Solid and dotted arrows correspond to edges in the spanning and non-spanning tree, respectively

and is used for filtering nodes to reduce the target search space. We adopt the indexing approach of [59]⁴ which performs depth-first traversal on the DAG G_{dag} . During the traversal, the preorder index of v is assigned (Line 2), and v 's descendants (Lines 6 to 7) and non-spanning tree (NST) children (Line 8) are identified. These are subsequently use to perform the reachability check in the ASSESSREACHABILITY procedure. Finally, the postorder index v is assigned (Line 9) when all of v 's descendants are visited. For example, in Figure. 5.4, the root nodes (E, PP2A, $V_{meta:1}$ and MKP3) are first identified and DAG indexing is performed on each of them, starting from E. The preorder and postorder values of E are set to 1 and 6, respectively while the non-spanning tree children and descendants are set to null and $\{V_{meta:1}, V_{meta:4}\}$, respectively. Repeating the DAG indexing process on each root node results in an indexed DAG.

In the ASSESSREACHABILITY procedure, only nodes having a path leading to the output node are added to the set of filtered target nodes $V_{candidate}$. An index-based reachability algorithm is used to determine if there exists a path from each node u to the output node x (denoted as $u \rightarrow x$). Given a node u and an output node x , let w be the non-spanning tree descendent of u and $u.preorder$ and $u.postorder$ denote the preorder and postorder indexes of u , respectively. A path $u \rightarrow x$ exists if any of the following conditions are satisfied [59]:

- (i) $u.preorder \leq x.preorder$ and $u.postorder \geq x.postorder$

⁴Note that more efficient approaches such as [373] and [356] can be adopted as well. However, this is orthogonal to the problem.

Algorithm 4: Algorithm INDEX**Input:** DAG $G_{dag} = (V_{dag}, E_{dag})$, child node u , parent node v .**Output:** DAG G_{dag} .

```

1 if  $v.preorder = null$  then
2    $v.preorder \leftarrow \text{SETTONEXTINDEX}(v.preorder)$ 
3   foreach iteration  $i=1$  to  $|V_{dag}|$  do
4      $w \leftarrow \text{GETCHILDNODE}(v)$ 
5      $G_{dag} \leftarrow \text{INDEX}(G_{dag}, w, v)$ 
6      $v.descendents \leftarrow \text{INSERTNODE}(v.descendents, w)$ 
7      $v.descendents \leftarrow v.descendents \cup w.descendents$ 
8      $v.NSTNodes \leftarrow v.NSTNodes \cup w.NSTNodes$ 
9    $v.postorder \leftarrow \text{SETTONEXTINDEX}(v.postorder)$ 
10 else if  $v.preorder \neq null$  and  $u \neq null$  then
11    $u.descendants \leftarrow \text{INSERTNODE}(u.descendants, v)$ 
12    $u.NSTNodes \leftarrow \text{INSERTNODE}(u.NSTNodes, v)$ 
13 if  $u \neq null$  then
14    $u.descendants \leftarrow u.descendants \cup v.descendants$ 
15    $u.NSTNodes \leftarrow u.NSTNodes \cup v.NSTNodes$ 

```

(ii) $w.preorder \leq x.preorder$ and $w.postorder \geq x.postorder$

Note that in a directed acyclic graph G_{dag} , u may be a SCC meta node. In which case, all the component nodes in this SCC will be retrieved and added to the set of filtered candidate targets $V_{candidate}$. For example, in Figure 5.4, if ERKPP (which belongs to $G.\text{SCC}_4$ represented by $v_{meta:4} \in V_{DAG}$) is chosen as x , then no nodes will be filtered as $v_{meta:4}$ is a leaf node and all nodes will be added to $V_{candidate}$.

LEARNFEATURESANDSETTINGS Procedure. Next, we describe Algorithm 5. Given a $|V_{candidate}| \times |X_{all}|$ matrix of feature values H , a target set T_x and a step-size of the misclassification cost s , the goal is to identify a set of predictive structural features $\mathcal{F}$ and the parameter configurations for the best prioritization model $\mathcal{M}$. This is achieved through the following steps.

First, the algorithm initializes the misclassification cost of the target class C^+ to a default value of 0.5 in Line 1. Then, it performs feature selection (Line 2) to obtain the predictive

Algorithm 5: Procedure LEARNFEATURESANDSETTINGS

Input: Matrix of structural feature values H , known target set T_x and step-size of the misclassification cost s .

Output: Predictive structural feature set $\mathcal{F}$ and prioritization model $\mathcal{M}$.

```

1  $C^+ \leftarrow \text{INITIALIZE}()$ 
2  $\phi_{best}, \mathcal{F}, \mathcal{M} \leftarrow \text{SELECTFEATURES}(H, T_x, C^+)$ 
3 foreach iteration  $i=1$  to  $\frac{1}{s} - 1$  do
4    $C^+ \leftarrow i \times s$ 
5    $\phi_{curr} \leftarrow \text{TUNESVC}(\mathcal{F}, T_x, C^+)$ 
6   if  $\phi_{curr} > \phi_{best}$  then
7      $\phi_{best} \leftarrow \phi_{curr}$ 
8      $\mathcal{M} \leftarrow \text{SETBESTPARAMWEIGHT}(\mathcal{M}, C^+)$ 

```

structural feature set $\mathcal{F}$. Next, it iterates through the range of misclassification cost ($0 - 1$) using the step-size s . In each iteration, it sets the misclassification cost of the target class C^+ (Line 4), then performs SVM training (Line 5) to obtain the parameter settings of the prioritization model $\mathcal{M}$ with the best accuracy (Lines 7 to 8). Note that in this thesis, we examine three different approaches for feature selection, namely, BSE, WRE and WRE-BSE. The BSE approach is a well-known greedy approach and interested readers may refer to [227]. The WRE and WRE-BSE approaches are outlined in Algorithms 6 and 7, respectively.

We shall now describe the structural feature selection algorithms (Algorithms 6 and 7) in turn. Given a matrix of feature values $\mathcal{H}$, a known target set T_x and misclassification cost for the target class C^+ , the goal of these algorithms is to find the accuracy ϕ , a predictive feature set $\mathcal{F}$ and the parameters of the prioritization model $\mathcal{M}$ that best prioritizes T_x .

Algorithm 6 outlines the WRE approach which performs statistical tests (Wilcoxon and ROC) and uses insights of the test results to eliminate features. Unlike the BSE approach which is integrated with the SVM, the WRE approach is like a wrapper layer that sits on top of the SVM. The wrapper layer identifies features with $p\text{-value} < 0.9$ and ROC AUC < 0.7 for removal (Lines 1 to 3). The best prioritization model is found by training the SVM using the remaining features (Line 4).

Algorithm 6: Algorithm WRE

Input: Matrix of structural feature values H , target set T_x and misclassification cost for target class C^+ .

Output: Prediction accuracy ϕ , predictive structural feature set $\mathcal{F}$ and prioritization model $\mathcal{M}$.

- 1 $\mathcal{R}_{Wilcoxon} \leftarrow \text{WILCOXONFILTER}(H, T_x)$
- 2 $\mathcal{R}_{ROC} \leftarrow \text{ROCFILTER}(H, T_x)$
- 3 $\mathcal{F} \leftarrow \mathcal{R}_{Wilcoxon} \cap \mathcal{R}_{ROC}$
- 4 $\phi, \mathcal{M} \leftarrow \text{TUNESVC}(\mathcal{F}, T_x, C^+)$

Algorithm 7 outlines the hybrid approach (WRE-BSE) which first performs WRE (lines 1 to 3), then BSE (lines 4 to 11). The steps for performing WRE have been described above in Algorithm 6. We shall now briefly describe the steps for performing BSE. First, the algorithm iteratively removes a feature from the remaining structural feature set $\mathcal{F}$ obtained from WRE and trains the optimal SVM model using the TUNESVC procedure (Line 7). Then, it identifies the structural feature $\mathcal{F}_i$ that when removed, results in a SVM model with the best prediction accuracy $\phi_{currBest}$ (Line 8). This feature is removed from $\mathcal{F}$ if $\phi_{currBest}$ is greater than $\phi_{prevBest}$ (Line 9). The algorithm stops the greedy structural feature elimination either when a single feature remains in $\mathcal{F}$ or when further feature removal did not produce a better model (Line 12).

5.5.3 Algorithm Complexity Analysis

Theorem 5.1 *Given a signaling network G , an output node x , a target set T_x , a feature set $\mathcal{X}_{all}$ and the step-size of the misclassification cost s , the Algorithm TENET has worst-case time complexity $O((|V| + |E|)^2 + O(\mathcal{G}(\mathcal{X}_{all})) + O(\mathcal{T}(\cdot)))$ where $\mathcal{G}(\mathcal{X}_{all})$ is the worst-case time complexity for extracting the features and $O(\mathcal{T}(\cdot))$ is the worst-case time complexity of the feature selection method used.*

Proof: In the FILTERCANDIDATE algorithm, the conversion of the input signaling network $G = (V, E)$ to a bipartite graph $G_{bi} = (V_{bi}, E_{bi})$ takes $O(|V_{bi}| + |E_{bi}|)$ time. In DAG conversion, $O(|V_{bi}| + |E_{bi}|)$ time is required for finding SCC using [351]. In the worst case, the

Algorithm 7: Algorithm WRE-BSE

Input: Matrix of structural feature values H , target set T_x , misclassification cost for target class C^+

Output: Prediction accuracy ϕ , predictive structural feature set $\mathcal{F}$, prioritization model $\mathcal{M}$.

```

1  $\mathcal{R}_{Wilcoxon} \leftarrow \text{WILCOXONFILTER}(H, T_x)$ 
2  $\mathcal{R}_{ROC} \leftarrow \text{ROCFILTER}(H, T_x)$ 
3  $\mathcal{F} \leftarrow \mathcal{R}_{Wilcoxon} \cap \mathcal{R}_{ROC}$ 
4  $\phi_{prevBest}, param_{prevBest} \leftarrow \text{TUNESVC}(\mathcal{F}, T_x, C^+)$ 
5 repeat
6   foreach iteration  $i=1$  to  $|\mathcal{F}|$  do
7      $\phi_i, param_i \leftarrow \text{TUNESVC}(\mathcal{F} - \mathcal{F}_i, T_x, C^+)$ 
8      $\mathcal{F}_i, \phi_{currBest}, param_{currBest} \leftarrow \text{GETFEATURETOREMOVE}(\phi_i, param_i)$ 
9     if  $\phi_{currBest} > \phi_{prevBest}$  then
10        $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{F}_i$ 
11        $\mathcal{M} \leftarrow param_i$ 
12 until  $|\mathcal{F}| = 1$  or  $\phi_{currBest} < \phi_{prevBest}$ ;
```

Structural Features	Time Complexity	Space Complexity
Degree centrality	$O(V)$	$O(V + E)$
Eigenvector centrality	$O(V ^2)$ [181]	$(O V ^2 + E)$
Closeness centrality	$O(V ^3)$ [174]	$(O V ^2 + E)$
Eccentricity centrality	$O(V E)$ [347]	$(O V ^2 + E)$
Betweenness centrality	$O(V E)$ [39]	$O(V + E)$ [39]
Bridging centrality	$O(V ^2 + E)$	$O(V + E)$
Bridging coefficient	$O(V ^2)$	$O(V + E)$
Clustering coefficient	$O(V ^{2.373})$ [384]	$O(V ^2 + E)$
Proximity prestige	$O(V ^2 + E)$	$O(V + E)$
Target downstream effect	$O(V ^2 + E)$	$O(V + E)$

Table 5.4: Time and space complexity for computing the different features. The proofs of those algorithm complexities that are provided without citation are reported in Appendix B.

signaling network is a complete directed graph and CONVERT2DAG takes $O(|V_{bi}|^2 + |E_{bi}|)$ time since $|V_{bi}| < |V_{bi}|^2$. In the indexing of the DAG graph $G_{dag} = (V_{dag}, E_{dag})$, the depth-first traversal requires $O(|V_{DAG}| + |E_{DAG}|)$ time while computing the set of nodes that can reach x takes $O(|V_{DAG}|)$ time. Hence, FILTERCANDIDATE algorithm takes $O(|V_{bi}|^2 + |E_{bi}|)$ time since $|V_{bi}| = |V| + |E|$, $|E_{bi}| = \sum_{(U,W) \in E} (|U| + |W|)$ and $(|V_{bi}| + |E_{bi}|) \geq (|V_{DAG}| + |E_{DAG}|)$.

The time complexity of the EXTRACTALLFEATURES procedure (denoted as $O(\mathcal{G}(\cdot))$) de-

depends on the features to be extracted. Table 5.4 contains the time and space complexity for computing the different structural features in this thesis. Amongst the features, closeness centrality has the highest time complexity ($O(|V|^3)$).

The time complexity of the `LEARNFEATURESANDSETTINGS` procedure is dependent on the time complexities of the feature selection approach (denoted as $O(\mathcal{T}(\cdot))$) and the training of the misclassification cost. Three feature selection approaches are explored in this thesis. In BSE, a greedy approach is used for selecting a feature for removal at each iteration and a new SVM model is trained and tuned accordingly using the `TUNESVC` procedure. `TUNESVC` uses the grid search approach described in Section 5.4.2 to tune the SVM parameters. The tuning process takes $O(i^p \times k)$ where i , p and k are the number of iterations⁵ required for the grid search, the number of parameters to be tuned, and the time complexity of training a SVM, respectively. According to [357], standard SVM training takes $O(m^3)$ time where m is the training set size. Hence, `TUNESVC` has $O(i^p \times |V|^3)$ time complexity since the training set size is approximately equal to the data set size ($|V|$). In the worst case, algorithm BSE removes all but one feature. This takes $O(|\mathcal{X}_{all}|^2 \times i^p \times |V|^3)$ time where $\mathcal{X}_{all}$ is the set of structural features. In WRE, the statistical-based filter requires two steps, namely, `WILCOXONFILTER` and `ROCFILTER` to find the predictive feature set. Performing the Wilcoxon test for a particular structural feature requires $O((gh)^2)$ time [253] where g and h are the target and non-target class sizes, respectively, and $|V| = g + h$. Generating ROC for a particular structural feature requires $O(|V|^2)$ time [110]. Hence, `WILCOXONFILTER` and `ROCFILTER` require $O((gh)^2|\mathcal{X}_{all}|)$ and $O(|V|^2|\mathcal{X}_{all}|)$ time complexities, respectively. The intersection of the two sets of features generated by `WILCOXONFILTER` and `ROCFILTER` takes $O(|V|)$ time in the worst case [93]. Hence, WRE requires $O(|V|^2|\mathcal{X}_{all}| + i^p \times |V|^3)$ time. This can be further simplified to $O(i^p \times |V|^3)$ since $|V| > |\mathcal{X}_{all}|$ in most signaling networks. The time complexity of WRE-BSE is $O(|\mathcal{X}_{all}|^2 \times i^p \times |V|^3)$, the maximum of the time complexities of BSE and WRE. The train-

⁵The number of iterations i needed for the grid search is dependent on the number of tuning level l , the range of the parameter to be searched r and the step size during the search s . Formally, it is defined as $i = l \times \frac{r}{s}$.

ing of the misclassification cost takes $O((\frac{1}{s} - 1)(i^p \times |V|^3))$. Hence, taken together, LEARN-
FEATURESANDSETTINGS procedure has time complexity of $O(O(\mathcal{T}(\cdot)) + (\frac{1}{s} - 1)(i^p \times |V|^3))$.

Taken together, the TENET algorithm requires $O(|V_{bi}|^2 + |E_{bi}| + O(\mathcal{G}(\mathcal{X}_{all})) + O(\mathcal{T}(\cdot) + (\frac{1}{s} - 1)(i^p \times |V|^3)))$ time for computation. In the worst case, the signaling network is a single strongly connected component with edges connecting every pair of nodes. Such a network implies that $O(|E_{bi}|) = O(|V_{bi}|^2)$. Hence, in the worst case, the time complexity of TENET is $O((|V| + |E|)^2 + O(\mathcal{G}(\mathcal{X}_{all})) + O(\mathcal{T}(\cdot)))$ since $|V_{bi}| = |V| + |E|$.

Theorem 5.2 *Given a signaling network G , an output node x , a target set T_x , a feature set $\mathcal{X}_{all}$ and the step-size of the misclassification cost s , the Algorithm TENET has worst-case space complexity $O((|V| + |E|)^2 + O(I(\mathcal{X}_{all})) + O(\mathcal{J}(\cdot)))$ where $I(\mathcal{X}_{all})$ is the worst-case space complexity for extracting the features and $O(\mathcal{J}(\cdot))$ is the worst-case space complexity of the feature selection method used.*

Proof: Computing $G_{bi} = (V_{bi}, E_{bi})$, the bipartite graph of the input signaling network $G = (V, E)$, requires $O(|V_{bi}| + |E_{bi}|)$ space to store the nodes and edges. *Convert2DAG* requires $O(|V_{bi}| + |E_{bi}|)$ and $O(|V_{DAG}| + |E_{DAG}|)$ space for SCC identification using [351] and for DAG conversion, respectively. Let D_v be the set of nodes such that there exists a path from $v \in V_{DAG}$ to $d \in D_v$ and let $|E_{v(non-tree)}|$ be the number of non-spanning tree edges of v . *Index* uses $O(|V_{DAG}| + \sum_{v \in V_{DAG}} |D_v| + \sum_{v \in V_{DAG}} |E_{v(non-tree)}|)$ space for storing the descendant nodes and non-spanning tree edges. In the worst case, the spanning tree of the DAG is a linear path and for every node $v \in V_{DAG}$, there is an edge to every descendant node of v so that *Index* requires $O(|V_{DAG}| + |V_{DAG}|^2)$ space. Computing the set of nodes that can reach x requires $O(|V|)$. Since $(|V_{bi}| + |E_{bi}|) \geq (|V_{DAG}| + |E_{DAG}|)$, $|V_{bi}| \geq |V|$ and $|E_{bi}| \geq |E|$, Algorithm 2 requires $O(|V_{bi}| + |E_{bi}| + |V_{DAG}|^2)$ space in the worst case.

The space complexity of the EXTRACTALLFEATURES procedure (denoted as $O(I(\cdot))$) depends on the features to be extracted. Amongst the features studied in this thesis, closeness centrality has the highest time complexity ($O(|V|^3)$) (Table 5.4).

The space complexity of the `LEARNFEATURESANDSETTINGS` procedure (denoted as $O(\mathcal{J}(\cdot))$) depends on the algorithm used. In BSE, the `TUNESVC` procedure requires $O(|V|^2)$ [357] for standard SVM assuming that we retain only the current best SVM model during the grid search process. In the worst case, which happens in the first while-loop iteration, we have to compute the SVM $|\mathcal{X}_{all}|$ times after the removal of individual structural features. Assuming that only the current best SVM model after feature removal is stored, the space complexity of BSE remains unchanged at $O(|V|^2)$. In WRE, the statistical-based filter requires two steps, namely, `WILCOXONFILTER` and `ROCFILTER` to find the predictive feature set. Performing the Wilcoxon test for a particular structural feature requires $O(g^2h)$ space [253] where g and h are the target and non-target class sizes, respectively; $|V| = g + h$ and $g < h$. Generating ROC for a particular structural feature requires $O(|V|)$ space using the approach in [110]. Hence, `WILCOXONFILTER` and `ROCFILTER` require $O(g^2h|\mathcal{X}_{all}|)$ and $O(|V||\mathcal{X}_{all}|)$ space complexities, respectively. The union of the two sets of features generated by `WILCOXONFILTER` and `ROCFILTER` requires $O(|V|)$ space using the data structure specified in [124]. Hence, WRE requires $O(g^2h|\mathcal{X}_{all}| + |V||\mathcal{X}_{all}| + |V| + |V|^2)$ space. This can be further simplified to $O(g^2h|\mathcal{X}_{all}| + |V|^2)$ since $|V| > |\mathcal{X}_{all}|$ in most signaling networks. In situation where $g \ll h$, the space complexity can be further reduced to $O(|V|^2)$ since h is approximately $|V|$ and $g^2 < |V|$. However, when $g \gg h$, the g^2 term dominates and the space complexity becomes $O(g^2h|\mathcal{X}_{all}|)$. The space complexity of WRE-BSE is $O(g^2h|\mathcal{X}_{all}| + |V|^2)$, the maximum of the space complexities of BSE and WRE.

Taken together, the TENET algorithm requires $O(|V_{bi}| + |E_{bi}| + |V_{DAG}|^2 + O(I(\mathcal{X}_{all})) + O(\mathcal{J}(\cdot)))$ space for computation. In the worst case, the signaling network is a single strongly connected component with edges connecting every pair of nodes. Such a network implies that $O(|E_{bi}|) = O(|V_{bi}|^2)$. Hence, in the worst case, the space complexity of TENET is $O((|V| + |E|)^2 + O(I(\mathcal{X}_{all})) + O(\mathcal{J}(\cdot)))$ since $|V_{bi}| = |V| + |E|$ and $|V_{bi}|^2 > |V_{DAG}|^2$.

5.6 Experiments

In this section, we perform various experiments to evaluate the performance of TENET. Note that in TENET, the prioritization model can be learnt using a training set that is obtained from the data of a single network or that of several networks *combined*. The *combined network* is created by integrating different networks. Note that in the networks we study, the node and edge sets are distinct. Hence, the combined network is a supergraph consisting of several disconnected subgraphs. For instance, for the `MAPK-PI3K` network, the data can be either an individual network (*i.e.*, `MAPK-PI3K` network alone) or a combined network consisting of all networks. That is, the combined network is a supergraph consisting of four disconnected subgraphs, each representing a network in Section 4.2. In the former case, the training and test sets are subset of nodes randomly selected from the `MAPK-PI3K` network and their associated structural features. In the latter case, the test set is the entire set of nodes in the `MAPK-PI3K` network and their associated structural features. The training set is obtained from the remaining networks. In the sequel, we shall explore the use of data of both individual network and combined network for generating prioritization models and their effects on the prioritization results.

Recall that in Algorithm 1, the `LEARNFEATURESANDSETTINGS` procedure learns the predictive feature set and parameter settings of the prioritization model that yields the best prediction accuracy on the validation data using a cross-validation approach. The experiments are designed to evaluate this procedure and to answer the following questions.

- (i) Q1: Is there a single structural feature that can characterize targets effectively in all signaling networks?
- (ii) Q2: Do single features characterize targets better than multiple features?
- (iii) Q3: Which feature selection approach performs the best in terms of predicting known targets?

- (iv) Q4: How does misclassification cost affect the prediction of known targets?
- (v) Q5: Which TENET variant performs the best?
- (vi) Q6: How does the best performing TENET variant compare with random prioritization and other state-of-the-art approaches?

Specifically, questions Q1 and Q2 are for justifying the TENET approach for target characterization.

5.6.1 Experimental Setting

TENET is implemented using Java. The Wilcoxon test and the ROC analysis are performed using *R*. In addition, we use several publicly available tools and libraries that are listed below:

- (i) *libSVM* library [51]: for training the SVM.
- (ii) JUNG library [267]: for creating the graph model.
- (iii) *libSBML* library [37]: for reading and processing the SBML files of the networks.
- (iv) *Copasi* [152]: for performing LSA [139].
- (v) *NetworkPrioritizer* [171] plugin for *Cytoscape* [320]: for comparison with TENET.

The experiments are performed on a computer system using a 64-bit operating system with 8GB RAM and a dual core processor running at 3.60GHz.

Data Set. In this chapter, we study four networks (Section 4.2). Recall that our experimental study will examine the effect of generating prioritization model from using individual networks and combined networks. Hence, a total of eight different networks (summarized in Table 5.5) will be studied, each having a different test set.

	I₁	I₂	I₃	I₄
Data set	MAPK-PI3K	glucose-stimulated insulin secretion	endomesoderm gene regulatory	glucose metabolism
Cross-validation used for model selection	8	5	10	10
Test set	Table 5.6 (I ₁)	Table 5.6 (I ₂)	Table 5.6 (I ₃)	Table 5.6 (I ₄)
Number of targets in test set	1	1	21	2
Percentage of targets in test set	25%	10%	34.4%	40%

	C₁	C₂	C₃	C₄
Data set	All networks	All networks	All networks	All networks
Cross-validation used for model selection	10	10	10	10
Test set	MAPK-PI3K	glucose-stimulated insulin secretion	endomesoderm gene regulatory	glucose metabolism
Number of targets in test set	10	6	206	16
Percentage of targets in test set	27.8%	10.2%	33.1%	34%

Table 5.5: Summary of data set. The network IDs are used for identifying the networks in the figures in Section 5.6. The test sets are listed in Table 5.6.

Network ID	Test Set
I₁	MEKPP [†] , RPI3K, internalization, PP2A
I₂	FBP [†] , OXA _{cyt} , Malate _{cyt} , Aspartate, Glutamate, Citrate, ATP, NADP ⁺ , NAD ⁺ , Ubiquinone
I₃	GENE_E_Otx [†] , GENE_E_Pmarl [†] , GENE_M_Eve [†] , GENE_P_Bra [†] , GENE_P_Eve [†] , PRE_E_UbiqSoxB1 [†] , PRE_P_CB [†] , PROTEIN_E_HesC [†] , PROTEIN_E_UVAotx [†] , PROTEIN_E_CB [†] , PROTEIN_E_nBTcf [†] , PROTEIN_M_Pmarl [†] , PROTEIN_M_SoxB1 [†] , PROTEIN_P_Delta2 [†] , PROTEIN_P_SoxB1 [†] , mRNA_E_Ets1 [†] , mRNA_M_HesC [†] , mRNA_M_SoxB1 [†] , mRNA_M_UbiqSoxB1 [†] , mRNA_P_Endol6 [†] , mRNA_P_UbiqEts1 [†] , GENE_E_Apobec, GENE_EES, GENE_E_Kakapo, GENE_E_OrCt, GENE_E_Sm50, GENE_E_SuTx, GENE_M_Alx1, GENE_MCAPK, GENE_M_FoxO, GENE_M_Nrl, GENE_M_VEGFR, GENE_P_OrCt, GENE_P_Sm27, PRE_E_VEGF, PROTEIN_E_Apobec, PROTEIN_E_Ficolin, PROTEIN_E_Hex, PROTEIN_E_OrCt, PROTEIN_E_Sm27, PROTEIN_M_FoxN23, PROTEIN_M_L1, PROTEIN_M_Sm27, PROTEIN_M_UbiqSoxC, PROTEIN_P_Dpt, PROTEIN_P_Dri, PROTEIN_P_FoxA, PROTEIN_P_Pks, mRNA_E_Dri, mRNA_E_ES, mRNA_E_Gcm, mRNA_M_Gcad, mRNA_M_Hex, mRNA_M_Not, mRNA_M_Snail, mRNA_M_Tbr, mRNA_M_Tel, mRNA_M_z13, mRNA_P_Apobec, mRNA_P_Dri, mRNA_P_OrCt
I₄	Isocitrate [†] , EI1A [†] , Emp, Enolase, Crp

Table 5.6: Test set of I₁ to I₄. Nodes marked with [†] are known targets.

Parameters. The ranges of the parameters that are tuned using the grid-search approach (described in Section 5.4.2) are listed in Table 5.7. We use a 3-level grid-search which progressively narrows down the search region. At each level, the region is divided into ten equal parts.

Parameters	Type	Range Tested
C	SVM parameter	$[2^{-12}-2^{12}]$
γ	kernel parameter	$[2^{-12}-2^{12}]$
d	kernel parameter	$[2-6]$
c_0	kernel parameter	$[2^{-12}-2^{12}]$

Table 5.7: SVM kernel types and their associated parameters [238].

For each parameter, 30 SVM models are trained and the prioritization model is the one that yields the best prediction accuracy for the cross-validation process. Interested readers may refer to [155] for details of the grid-search approach.

Variants of TENET. In this chapter, we study the use of three different feature selection methods and the weighed misclassification cost approach to address the challenges of training the SVM (Section 5.4.3). Henceforth, we shall refer to these approaches as SVM-training approaches. By varying the SVM-training approaches incorporated into the TENET variants, the effects of each training approach can be examined. Table 5.8 summarizes these different TENET variants. Note that the WMC training approach involves specification of the misclassification cost of the target class (denoted as C^+) and the non-target class (denoted as C^-). We vary the weights according to the constraints stated in Equation 5.10. Table 5.9 provides details of the weights ratio of C^+ to C^- used in the experiments where $C^+=W_1$ and $C^-=W_2$. The weight ratio IDs in this table are used later in Figures 5.9 to 5.12.

Approaches that were Compared with TENET. We compare the results of TENET against several other methods, namely, random prioritization (baseline approach), LSA [139] and *NetworkPrioritizer* [171]. In random prioritization, the nodes were randomly assigned a rank in the range $[1-|V|]$ where $|V|$ is the number of nodes in the signaling network and we assume that no ranking ties are present. LSA was performed using *Copasi* [310] with the following configuration: {task=sensitivities; subtask=time series; function=all variables of the model;

TENET Variant	BSE	WRE	WRE-BSE	WMC
TENET-naïve				
TENET-B	✓			
TENET-R		✓		
TENET-H			✓	
TENET-W				✓
TENET-WB	✓			✓
TENET-WR		✓		✓
TENET-WH			✓	✓

Table 5.8: The different variants of TENET studied. ✓ indicates the SVM-training approach(es) incorporated into the different variants. For instance, TENET-WB incorporates the BSE feature selection approach and the weighted misclassification cost.

Weights Ratio ID	W_1	W_2
1	0.1	0.9
2	0.2	0.8
3	0.3	0.7
4	0.4	0.6
5	0.5	0.5
6	0.6	0.4
7	0.7	0.3
8	0.8	0.2
9	0.9	0.1

Table 5.9: Weights ratio used in the experiments.

and variable=all parameter values}. *NetworkPrioritizer* allows prioritization using a variety of features⁶. The prioritization ranks are computed using user-configurable rank aggregation approaches, namely, *Weighted Borda Fuse* (WBF) and *Weighted AddScore Fuse* (WASF). The former aggregation approach is a generalization of the Borda count [85] while the latter approach is based on the weighted sum score of the different feature values. Note that for both approaches, we consider all features provided in *NetworkPrioritizer* and assign uniform weights to them for rank aggregation since we do not have prior knowledge of which feature or weight is better. Interested readers may refer to Chapter 2 for a review on LSA and *NetworkPrioritizer*.

⁶Degree centrality, shortest path betweenness, shortest path closeness, random walk betweenness, random walk transmitter closeness and random walk receiver closeness. Interested reader may refer to [171] for more information.

Performance Metrics. In a binary-classification problem, a confusion matrix is typically used to determine the performance of the classifier [111]. In particular, the performance of different variants of TENET is compared based on their prediction *accuracy*, *sensitivity*, *specificity* and *precision* of their prioritization models using the same training and test set.

The calculation of these metrics are as follows:

$$\text{Accuracy } (\phi) = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.11)$$

$$\text{Sensitivity (TPR)} = \frac{TP}{TP + FN} \quad (5.12)$$

$$\text{Specificity (TNR)} = \frac{TN}{FP + TN} \quad (5.13)$$

$$\text{Precision (PPV)} = \frac{TP}{TP + FP} \quad (5.14)$$

where TP, TN, FP and FN denote true positive, true negative, false positive and false negative prediction, respectively. Note that PPV is set to 0 when the classifier did not make any positive prediction.

In addition to these metrics, we include an additional metric called *feature reduction factor* to compare the performance of the feature selection methods:

$$\text{Feature reduction factor (FRF)} = 1 - \frac{|\mathcal{F}|}{\mathcal{X}_{all}} \quad (5.15)$$

where $\mathcal{X}_{all}$ is the entire set of features considered in the study and $\mathcal{F} \subset \mathcal{X}_{all}$ is the set of features selected using the feature selection method.

We use the linear sum of the performance metrics to generate a performance score $\mathcal{P}$ for comparing one prioritization model against another, and the best prioritization model for a particular network is the one having the highest $\mathcal{P}$.

$$\mathcal{P} = \sum_{m \in M} \text{value}_m \quad (5.16)$$

where $M = \{\bar{\phi}(val), \phi(test), \text{TPR}, \text{TNR}, \text{PPV}\}$ and value_m is the value of metric m .

5.6.2 Experimental Results

The experiments are organized as follows. First, we characterize targets in the networks in terms of structural features using the Wilcoxon test and ROC analysis. Then, we study the effect of structural feature selection and WMC. Note that the issue of SVM kernel selection will also be explored to determine the best kernel for our problem. Finally, we compare the structural feature-based ranks obtained using prioritization models created by TENET against ranks found using a random classifier and other state-of-the art approaches.

Structural Features-Based Target Characterization. In this set of experiments, we perform Wilcoxon test and ROC analysis on four individual networks (I_1 to I_4) and a combined network generated from I_1 to I_4 in order to characterize targets in these networks in terms of structural features. Tables 5.10 and 5.11 show the p -values and ROC AUC of the networks, respectively. Note that features with p -values lesser than 0.05 and ROC AUC greater than 0.7 are considered to be discriminative of targets and non-targets (referred to as *discriminative features*). These table entries are marked with *. From the tables, we note some general trends in the networks.

First, the statistical tests and the threshold ($p\text{-value} < 0.05$ and $\text{ROC AUC} > 0.7$) used to identify discriminative features directly affects the discriminative features selected. For instance, in the same network I_1 , total degree centrality θ_{total} , a discriminative feature based on the Wilcoxon test, does not distinguish targets from non-targets well enough based on ROC analysis. In Table 5.10, there were nine discriminative features in I_1 but none in I_4 based on the same threshold ($p\text{-value} < 0.05$). Note that the choice of the threshold is application-specific. Second, the number of discriminative features differed across the networks. In Table 5.11, some networks such as I_2 are characterized by as many as eleven features whereas other networks such as I_1 have only seven discriminative features. Note that it is also possible for a network to have no discriminative features (*e.g.*, I_4 when $\text{ROC AUC} > 0.7$). This implies that the structural feature set $\mathcal{X}_{all}$ may not be appropriate for characterizing targets in this network and additional

Structural Feature	I ₁	I ₂	I ₃	I ₄	Combined Network
Betweenness centrality δ	9.62×10^{-4} (0.999)*	0.934 (0.069)	1.77×10^{-5} (1)*	0.770 (0.237)	9.68×10^{-5} (1)*
Bridging centrality ζ	8.83×10^{-4} (0.999)*	0.173 (0.833)	3.34×10^{-11} (1)*	0.889 (0.115)	6.31×10^{-11} (1)*
Bridging coefficient π	0.095 (0.911)	0.018 (0.983)*	0.088 (0.912)	0.923 (0.078)	0.014 (0.987)*
Closeness centrality β	1.76×10^{-4} (1)*	0.993 (0.007)*	0.941 (0.059)	0.631 (0.377)	0.858 (0.143)
Cycle clustering coefficient κ_{cyc}	0.414 (0.601)	0.976 (0.025)*	0.764 (0.236)	0.397 (0.614)	0.802 (0.198)
Eccentricity centrality ϑ	0.015 (0.986)*	0.917 (0.088)	0.994 (0.006)*	0.399 (0.616)	0.986 (0.014)*
Eigenvector centrality α	0.058 (0.946)	0.999 (0.001)*	1.75×10^{-7} (1)*	0.790 (0.217)	9.04×10^{-6} (1)*
In clustering coefficient κ_{in}	0.297 (0.717)	0.876 (0.130)	0.942 (0.058)	0.461 (0.550)	0.927 (0.073)
In degree centrality θ_{in}	0.718 (0.295)	0.983 (0.018)*	0.062 (0.938)	0.731 (0.279)	0.277 (0.724)
Middleman clustering coefficient κ_{mid}	0.445 (0.571)	0.976 (0.025)*	$1 (2.80 \times 10^{-8})^*$	0.059 (0.944)	$1 (6.10 \times 10^{-6})^*$
Out clustering coefficient κ_{out}	0.545 (0.470)	0.764 (0.246)	0.682 (0.319)	0.489 (0.523)	0.518 (0.482)
Out degree centrality θ_{out}	0.006 (0.994)*	0.997 (0.003)*	5.54×10^{-5} (1)*	0.683 (0.326)	0.031 (0.969)*
Proximity prestige μ	0.011 (0.990)*	0.993 (0.007)*	0.549 (0.451)	0.832 (0.174)	0.892 (0.108)
Target downstream effect σ	0.961 (0.042)*	0.054 (0.949)	0.983 (0.017)*	0.071 (0.932)	0.909 (0.091)
Total degree centrality θ_{total}	0.033 (0.969)*	0.996 (0.004)*	3.64×10^{-4} (1)*	0.586 (0.424)	0.054 (0.946)
Undirected clustering coefficient κ_{undir}	0.994 (0.007)*	0.008 (0.993)*	$1 (1.84 \times 10^{-7})^*$	0.105 (0.899)	$1 (2.82 \times 10^{-4})^*$

Table 5.10: Wilcoxon test results of the networks. Each cell entry contains the p -value in the form of xxx(yyy) where xxx and yyy denote the p -values for the alternative hypotheses where the features of the known target nodes are smaller and greater than the non-target nodes, respectively.

features need to be considered. Third, different discriminative features performed differently on different networks. For instance, in Table 5.10, bridging centrality ζ , a feature found to yield good targets in a variety of PPI networks [160], performed well on I₁ and I₃, but poorly on I₂ and I₄. Fourth, the features found to be discriminative in the combined network does not necessarily perform well in all individual networks. For example, in Table 5.10, betweenness centrality δ was found to be discriminative for the combined network, I₁ and I₃, but not for I₂

Structural Feature	I_1	I_2	I_3	I_4	Combined Network
δ	0.838*	0.687	0.6	0.566	0.583
ζ	0.842*	0.619	0.662	0.609	0.646
π	0.644	0.763*	0.467	0.629	0.55
β	0.89*	0.807*	0.461	0.529	0.476
κ_{cyc}	0.523	0.717*	0.516	0.478	0.517
ϑ	0.729*	0.649	0.553	0.485	0.546
α	0.673	0.879*	0.626	0.572	0.597
κ_{in}	0.554	0.623	0.531	0.492	0.527
θ_{in}	0.556	0.755*	0.467	0.545	0.488
κ_{mid}	0.515	0.717*	0.628	0.629	0.593
κ_{out}	0.49	0.575	0.51	0.497	0.501
θ_{out}	0.758*	0.832*	0.587	0.54	0.46
μ	0.75*	0.807*	0.503	0.585	0.528
σ	0.688	0.701*	0.551	0.629	0.53
θ_{total}	0.696	0.825*	0.58	0.518	0.465
κ_{undir}	0.767*	0.802*	0.618	0.614	0.576

Table 5.11: ROC AUC of the networks.

and I_4 .

Hence, there is no single feature that can be used for distinguishing targets from non-targets in all signaling networks. In addition, the lack of discriminative features in certain networks (*e.g.*, I_4) implies that multiple features may be needed to distinguish targets from non-targets. Based on these results, we advocate that characterization of targets in signaling networks is highly dependent on network topology and demands analysis of multiple structural features. In subsequent experiments, we shall evaluate TENET, a SVM-based approach, which characterizes the targets of a given signaling network using multiple structural features to produce a prioritization model that can facilitate target prioritization in Chapter 6.

Kernel Selection. First we need to determine the SVM kernel. We experimented with several kernels: linear, radial basis function (RBF), sigmoid and polynomial. The parameters relevant to each kernel type and the ranges of these parameters that we tested are found in Tables 5.1 and 5.7, respectively. Figure 5.5 plots the results of TENET-naïve (which considers all structural features) using the various kernels. The choice of kernel did not affect the accuracy of the

validation and the test sets. This implies that the training data is likely to be linearly separable. The execution time, however, is affected by the number of parameters involved in the kernels and the training set size [281] (size of network). Henceforth, we shall use the linear kernel for the rest of the experiments since it yielded the same accuracy as other kernels but is faster in terms of training speed. The parameters for the best models in this experiment is reported in Appendix C. Note that in I_1 , the sensitivity (TPR) and precision (PPV) are zero irrespective of the kernels. This highlights a need to use additional techniques (*e.g.*, feature selection) to improve the prioritization models.

Feature Selection. Next, we examine the performance of different feature selection approaches (TENET-B, TENET-R and TENET-H) and compare it with TENET-naïve for different signaling networks. We compare the approaches based on the predictive feature sets, the performance metrics previously defined and the runtime performance. Note that this set of experiments, we study the effect of the feature selection approaches in isolation. The effect of incorporating WMC into the SVM will be studied later.

Table 5.12 summarizes the predictive feature sets for each network using different approaches. In total, 24 experiments were conducted since there are three feature selection methods and eight networks (I_1 to I_4 and C_1 to C_4). Amongst these 24 experiments, 25% of the predictive feature sets consist of only one feature while the remaining had multiple features (ranging from 4 to 15 features). This supports our previous observation that multiple features result in better prediction of known targets.

Figures 5.6 and 5.7 illustrate the performance of different approaches based on the performance metrics. We make several observations. First, no single approach performs consistently well on all performance metrics. For instance, TENET-H⁷ achieves the best average feature

⁷Note that TENET-H always has the same or better result than TENET-R because we first carry out TENET-R followed by TENET-B in TENET-H. Hence, the predictive feature set found using TENET-H must be a subset of that found using TENET-R.

Figure 5.5: Performance of different kernels using the TENET-naïve approach.

Figure 5.6: Performance of different feature selection approaches.

Figure 5.7: Boxplots of performance of different feature selection approaches.

		δ	ζ	π	β	κ_{cyc}	ϑ	α	κ_{in}	θ_{in}	κ_{mid}	κ_{out}	θ_{out}	μ	σ	θ_{total}	κ_{undir}
I ₁	TENET-B	✓		✓						✓			✓				
	TENET-R	✓	✓		✓		✓						✓	✓			✓
	TENET-H	✓	✓		✓		✓										
I ₂	TENET-B									✓							
	TENET-R	✓		✓	✓	✓		✓	✓	✓	✓		✓	✓		✓	✓
	TENET-H			✓	✓	✓											✓
I ₃	TENET-B	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
	TENET-R	✓	✓				✓	✓			✓		✓		✓	✓	✓
	TENET-H	✓	✓				✓	✓					✓			✓	✓
I ₄	TENET-B		✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
	TENET-R														✓		
	TENET-H														✓		
C ₁	TENET-B	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓		✓
	TENET-R	✓	✓	✓	✓		✓	✓			✓		✓				✓
	TENET-H		✓	✓			✓	✓					✓				✓
C ₂	TENET-B	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓				✓
	TENET-R	✓	✓					✓			✓		✓		✓	✓	✓
	TENET-H	✓	✓					✓			✓				✓		✓
C ₃	TENET-B									✓							
	TENET-R		✓														
	TENET-H		✓														
C ₄	TENET-B		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
	TENET-R	✓	✓	✓			✓	✓			✓		✓		✓	✓	✓
	TENET-H		✓	✓			✓	✓			✓		✓		✓	✓	✓

Table 5.12: Summary of structural features selected for different signaling networks using different approaches. ✓ indicates the feature selected for the approach. For instance, in I₄, TENET-R selected σ .

reduction (70.3%) while TENET-B has the best average prediction accuracy on the test set (75.6%). However, we note that TENET-B seems to outperform other approaches in several metrics. Second, a non-performing approach may still outperform a top-performing approach in specific instances. For example, TENET-H and TENET-R have the best test set prediction accuracy for I₃ (75.4%) and C₂ (78%), respectively. Taken together with the previous observation, the most appropriate feature selection approach is dependent on the signaling network. Third, feature selection does not necessarily lead to improvement in prediction accuracy. For instance, in C₄, the naïve approach has the best prediction accuracy (60.9%). The poor performance of the feature selection approaches could be due to imbalanced training and test data, and uninformative structural features. For instance, in I₂, the imbalanced training data tends to

Figure 5.8: Runtime performance of different feature selection approaches.

result in a model that is skewed towards correct classification mainly in the non-target class. Coupled with the fact that the test data has only one known target (Table 5.5), misclassification of this single target could result in a classifier that performs the same or worse than the naïve SVM. Indeed, the naïve SVM classifies all non-target class members correctly and misclassifies this single target. Hence, in order to perform better, a classifier must achieve 100% correct classification. Fourth, the models generally have high specificity. This is again partly due to the imbalanced data set with the majority class containing negative class response (*i.e.*, non-targets). The sensitivity and precision of the models are greatly affected by the target class size in the test data set as seen in I_1 and I_2 , which had only one target. Correctly classifying the target results in sensitivity and precision of 1 and misclassification results in sensitivity and precision of 0. Note that the problem of the imbalanced training set can be alleviated with WMC which we shall study later.

Next, we examine the runtime performance of different feature selection approaches. Figure 5.8 plots the runtime required to execute the `SELECTFEATURES` procedure. We make the following observations. First, TENET-R has the best runtime performance, followed by TENET-H and TENET-B. The poorer performance of TENET-B is due to the interaction of the feature selection approach with the classifier (classifier-aware approach) which is different

from TENET-R where the feature selection approach is a wrapper layer that sits on top of the classifier. Second, the size of the networks used for training affects the runtime performance. In general, larger size networks require longer runtime. For instance, I_3 (network ID 2) having 622 nodes requires longer runtime for all approaches compared to I_2 (network ID 1) with 59 nodes. Third, we note an unexpected dip in TENET-H for C_3 (network ID 6). Recall that the feature selection method in TENET-H first selects features using WRE, then performs BSE to further reduce the features. In C_3 , the feature set selected by WRE contains a single feature (bridging centrality) and no further reduction is allowed. Hence, TENET-H has the same runtime as TENET-R for C_3 .

Taken together, these observations imply that the choice and runtime performance of the feature selection approach are dependent on the signaling network.

Effect of Varying Misclassification Cost. Next, in this set of experiment, we examine the effect of incorporating WMC. Figures 5.9 to 5.16 illustrate the performance for I_1 to I_4 and C_1 to C_4 . Intuitively, we expect to observe the following trends in the figures as the target misclassification cost C^+ increases

- (i) The validation and test set prediction accuracy would display a negative skewed distribution.
- (ii) The test set sensitivity would display an increasing distribution.
- (iii) The test set specificity would display a decreasing distribution.
- (iv) The test set precision would display a positive skewed distribution.

This is because a large C^+ eventually results in a model that is likely biased towards classifying data as targets.

Figure 5.9: Performance of TENET variants incorporating feature selection approach and WMC for the `MAPK-PI3K` network.

We make the following observations from the figures. First, amongst the individual networks, only I_3 display the expected trends. This could be due to the extreme small target size (1 or 2) in the test set of the networks which resulted in extreme fluctuations in the performance metrics and deviation from the expected trends. Hence, the target size of the test set can have significant impact on the observed results. Second, the performance of the combined networks C_1 (Figure 5.13), C_2 (Figure 5.14) and C_4 (Figure 5.16) resembles that of I_3 . This could be due to the large size of I_3 which dominates over the rest of the networks used for training. This implies that when using a combined network for generating the prioritization model, the model can be easily influenced by dominating networks used for training. Third, sensitivity generally

Figure 5.10: Performance of TENET variants incorporating feature selection approach and WMC for the glucose-stimulated insulin secretion network.

improves while specificity generally deteriorates when the target misclassification cost is set higher than the non-target misclassification cost ($C^+ > C^-$). This observation is in line with our expected trend as is seen in networks such C_2 (Figure 5.14). It implies that sensitivity and specificity behave in an antagonistic manner with the change in the ratio of C^+ to C^- . The choice of an appropriate model depends on the application. For instance, in the case of prioritizing targets for drug development, choosing a less accurate but more sensitive model may be more appropriate for minimizing the likelihood of missing a real target. Fourth, the prediction accuracy tends to display a skewed distribution where accuracy initially increases (or remains constant) with increasing C^+ , and then decreases with increasing C^+ . Some networks

Figure 5.11: Performance of TENET variants incorporating feature selection approach and WMC for the endomesoderm gene regulatory network.

that display this trend are I_3 (Figure 5.11) and C_1 (Figure 5.13). Fifth, individual networks and combined networks behave differently. In individual networks, prediction accuracy, sensitivity and precision generally improve when C^+ is set larger than C^- . However, in combined networks, sensitivity improves whereas other performance criteria deteriorates when C^+ is set larger than C^- . Hence, there is no single universal best value of C^+ and the choice of C^+ depends on the network.

Best TENET Variant. Next, we identify the best TENET variant for the signaling networks. Recall that the linear sum of the performance metrics (referred to as performance score $\mathcal{P}$) is

Figure 5.12: Performance of TENET variants incorporating feature selection approach and WMC for the glucose metabolism network.

used to compare the different SVM model and the best variant yields a model with the highest performance score. Table 5.13 summarizes the best TENET variants of different networks. In the table, Δ_x denotes the change in performance metric x of the best TENET variant when compared to that of TENET-naïve. Specifically, it is defined as $\Delta_x = \frac{x_{best} - x_{naïve}}{x_{naïve}}$ where x_{best} and $x_{naïve}$ are the values of performance metric x of the best TENET variant and TENET-naïve, respectively. Note that, * marks instances where $x_{naïve} = 0$ and # marks the best models selected for configuring the SVM in the PRIORITIZETARGET procedure. In particular, the test set prediction accuracy, sensitivity, specificity and precision of TENET-naïve for I_1 are all 1. Hence, the change in performance metric remains at zero.

Figure 5.13: Performance of TENET variants incorporating feature selection approach and WMC for C_1 .

We make the following observations. First, the best TENET variant is network dependent. For instance, the best variants for I_2 and I_3 are TENET-WH ($C^+ = 0.9$) and TENET-WB ($C^+ = 0.7$), respectively. Second, variants incorporating both WMC and structural feature selection generally perform well. In particular, 14 out of the 17 best variants in Table 5.13 use both WMC and structural feature selection. Third, the best variants generally set misclassification cost of the target class higher than that of the non-target class (*i.e.*, $C^+ > C^-$). Observe that five out of six networks with best variants that incorporates WMC sets C^+ larger than C^- . Fourth, TENET variants based on individual networks outperforms that based on combined networks. In particular, the test set prediction accuracy for I_1 to I_4 is in the range $[0.803-1]$ whereas that

Figure 5.14: Performance of TENET variants incorporating feature selection approach and WMC for C_2 .

for C_1 to C_4 is in the range [0.609–0.78]. The poorer performance of C_1 to C_4 could be due to several reasons:

- (i) The small number of training networks do not provide sufficient information for a generalized model.
- (ii) The structural features selected for study are insufficient for characterizing a generalized model.
- (iii) Signaling networks by nature have distinct characteristics and it is just not possible to have a generalized model.

Figure 5.15: Performance of TENET variants incorporating feature selection approach and WMC for C_3 .

Hence, in general, TENET variants based on individual networks that incorporate structural feature selection and WMC with C^+ set larger than C^- tend to perform better.

Comparison of TENET with Other Approaches. Lastly, we compare TENET against other prioritization approaches, namely, random prioritization, LSA [139] and *NetworkPrioritizer* [171], in terms of the quality of prioritization and the runtime performance.

Recall that in Section 5.3.2, TENET-prioritized ranks can be obtained by solving the supervised learning problem as a regression problem. We use the ϵ -support vector regression

Figure 5.16: Performance of TENET variants incorporating feature selection approach and WMC for C_4 .

(ϵ -SVR) in *libSVM* for training the models and use the default ϵ value (1×10^{-3})⁸. The parameters of the SVM are set according to the best models for each network (Table 5.13). In cases where more than one best model exist, we preferentially select the approach with the largest C^+ when C^+ are set larger than C^- since large C^+ tends to improve the model. These model parameters are reported in Appendix C. Note that the nodes are ranked in decreasing order of the regression score. For other approaches, the steps for prioritizing the targets are described in Section 5.6.1.

Figures 5.17, 5.18 and 5.20 plots the *normalized ranks* of all nodes in the test sets of I_1 , I_2

⁸In ϵ -SVR, the error function is an ϵ -insensitive loss function and error smaller than ϵ is ignored [51]. Refer to Section 5.4.1.

Figure 5.17: Normalized rank of nodes in test set of I_1 .

Figure 5.18: Normalized rank of nodes in test set of I_2 .

Figure 5.19: Normalized rank of top 10 nodes ranked using TENET in test set of I_3 .

Figure 5.20: Normalized rank of nodes in test set of I_4 .

Figure 5.21: Normalized rank of targets in test set of C_1 .

Figure 5.22: Normalized rank of targets in test set of C_2 .

	I₁	I₂	I₃	I₄
Best Approaches	TENET-B [#] , TENET-WB (C ⁺ =0.1,0.2,0.3, 0.4)	TENET-WH (C ⁺ =0.9 [#])	TENET-WB (C ⁺ =0.7 [#])	TENET-WB (C ⁺ =0.2,0.3,0.4, 0.6,0.8 [#])
$\mathcal{P}$	4.935	4.109	3.86	4.9
$\bar{\phi}(val) [\Delta\bar{\phi}(val)]$	0.935 [0.16]	0.82 [-0.087]	0.747 [-0.02]	0.9 [0.268]
$\phi(test) [\Delta\phi(test)]$	1 [0]	0.9 [0]	0.803 [0.088]	1 [0.667]
TPR [Δ TPR]	1 [0]	1 [∞^*]	0.905 [0.462]	1 [1]
TNR [Δ TNR]	1 [0]	0.889 [-0.111]	0.75 [-0.063]	1 [0.499]
PPV [Δ PPV]	1 [0]	0.5 [∞^*]	0.655 [0.058]	1 [1]

	C₁	C₂	C₃	C₄
Best Approaches	TENET-WH (C ⁺ =0.6 [#])	TENET-R [#]	TENET-WR (C ⁺ =0.8 [#]), TENET-WH (C ⁺ =0.8),	TENET-naïve [#]
$\mathcal{P}$	3.08	3.022	3.268	2.917
$\bar{\phi}(val) [\Delta\bar{\phi}(val)]$	0.734 [-0.013]	0.711 [-0.052]	0.561 [-0.274]	0.757 [0]
$\phi(test) [\Delta\phi(test)]$	0.694 [0.136]	0.78 [0.070]	0.724 [0.097]	0.609 [0]
TPR [Δ TPR]	0.4 [0.333]	0.5 [0.502]	0.602 [∞^*]	0.313 [0]
TNR [Δ TNR]	0.808 [0.105]	0.811 [0.048]	0.788 [-0.212]	0.767 [0]
PPV [Δ PPV]	0.444 [0.48]	0.231 [0.615]	0.593 [∞^*]	0.471 [0]

Table 5.13: Summary of best TENET variant for different networks. The C⁺ values are provided in bracket besides approaches which uses WMC.

and I₄, respectively. For the remaining networks, due to the larger size of the test sets, we only plot the *normalized ranks* of selected nodes in the test sets in Figures 5.19 and 5.21 to 5.24. The *normalized rank* of a node u for a particular approach x is denoted as $\Psi_{norm(x):u}$ and defined as follows.

$$\Psi_{norm(x):u} = \frac{\Psi_{x,u}}{\text{MAX}_{i \in V}(\Psi_x : i)} \quad (5.17)$$

where $\Psi_{x,u}$ is the rank of u based on x , V is the set of nodes in the given signaling network and $\text{MAX}(\cdot)$ is the maximum operator. We use the normalized rank for comparison since the range of ranks for each approach is different. In these tables, we mark the known targets in the figures using (*) for ease of reading. Note that an ideal ranking approach should assign higher normalized ranks, represented by shorter histogram, to known targets. Interested readers

Figure 5.23: Normalized rank of top 15 nodes ranked using TENET in test set of C_3 . † *none* is the node used to represent degraded proteins in this network.

may refer to Appendix A for rankings of the entire test set. From these figures, we note that normalized ranks of a given node can vary widely using different prioritization approaches. For instance, in Figure 5.22, the normalized rank of $G3P$, a known target, is relatively high when prioritized using TENET, random prioritization and LSA. However, it is given relatively low ranking by WBF and WASF. In another network C_1 (Figure 5.21), WBF and WASF assigned higher ranks to Raf^* , another known target, compared to TENET and random prioritization. Hence, an approach that performs better for one particular network can perform poorly in another.

In order to compare the performance of different prioritization approaches, we perform

Figure 5.24: Normalized rank of targets in test set of C_4 .

ROC analysis based on the rankings on the nodes in the test set for each network. Figure 5.25 illustrates the performance of various approaches. In general, TENET outperforms the rest of the approaches for individual networks (network ID 0 to 3). In particular, the order of the methods in terms of average ROC AUC is $\text{TENET}(76.2\%, \text{s.d.}=0.19) \preceq \text{WBF}(69.9\%, \text{s.d.}=0.09) \preceq \text{WASF}(67.5\%, \text{s.d.}=0.17) \preceq \text{LSA}(64.1\%, \text{s.d.}=0.14) \preceq \text{Random}(62\%, \text{s.d.}=0.14)$. Note that the average ROC AUC is indicated in round brackets and s.d. represent standard deviation. Figures 5.26 and 5.27 illustrate the boxplot results of the ROC and AUPR analysis, respectively.

Finally, we analyze the runtime performance of TENET compared to other approaches. We note that in general, TENET and LSA perform more poorly than WBF and WASF. The poorer performance of TENET was largely due to the training of the prioritization model in the LEARNFEATURESANDSETTING procedure. When we exclude the execution time of this procedure, which can be performed offline, the performance of TENET (denoted as “TENET (Regression only)” in Figure 5.25) was comparable to that of WBF and WASF. For TENET,

Figure 5.25: Performance of different prioritization approaches on the signaling networks.

training networks based on individual networks performed better in terms of ROC AUC and AUPR. Henceforth, we shall only consider individual-based training networks for TENET.

Taken together with the previous findings, TENET outperforms other state-of-the-art ap-

Figure 5.26: ROC analysis on different approaches.

Figure 5.27: AUPR analysis on different approaches.

proaches in terms of the quality of the prioritization results and is comparable in terms of runtime performance.

5.7 Summary

In conclusion, TENET is a SVM-based approach that characterizes known targets in signaling networks using structural features. In particular, TENET identifies predictive structural features of known targets and uses these features to generate prioritization models that facilitate target prioritization in Chapter 6. It uses several approach such as structural feature selection and weighted misclassification cost to handle various issues during SVM training. Specifically, structural feature selection improves prediction accuracy of prioritization models by eliminat-

ing redundant features whereas weighted misclassification cost can be used to improve other performance criteria such as sensitivity. When we compare TENET with other state-of-the art approaches, we find that TENET outperforms other approaches in prioritizing several networks. In addition, we note that using individual networks for training the SVM resulted in better prioritization models compared to using combined networks. In summary, the contribution of this chapter is a machine learning-based framework that affords flexibility in characterizing signaling networks of different sizes and with different number of known targets.

Although TENET is evaluated on a small⁹ number of signaling networks, it can easily incorporate the study of more signaling networks. Note that no modification of the framework is required and there is no restriction on the size of networks, the number of known targets and the number of network features to be studied.

⁹Recall that manual target curation, a time-intensive process, is needed to identify known targets of signaling networks for validating our experimental results. Hence, we restrict our study to only 4 signaling networks.

Chapter 6

TAPESTRY: A Novel Algorithm for Fast Discovery of Putative Target Nodes

In Chapter 5, we presented TENET, the offline component of the *Target Prioritization* subsystem in TRECENTO, which learns a prioritization model and an associated set of structural features (referred to as predictive structural feature set) for a given network that can be used to generate a prioritization score. In this chapter, we present TAPESTRY (**T**Arget **P**rioritization using **n**Etwork **S**TReuctures and **d**Ynamics), the online component of *Target Prioritization*. Given a new unseen signaling network, TAPESTRY is responsible for selecting a prioritization model and the predictive structural feature set from the *Target Characterization Repository*, using the *Network Mapper Module*. The prioritization model is used to generate a rank (referred to as *structural feature-based (SFB) rank*) as described in Section 5.6.2, which is then *integrated* with a *dynamic feature-based (DFB) rank* generated by the *Dynamic Feature Ranker Module* to produce a final prioritization rank (referred to as *putative target score*) for each candidate target node. Note that TAPESTRY differs from TENET in two ways. First, the goal of TAPESTRY is to perform target prioritization using the prioritization models and predictive structural feature sets learnt from TENET. Second, we assume that each given network in TENET has an associated set of known targets. This assumption does not hold for TAPESTRY. For clarity, we refer to the networks given as input to TENET and to TAPESTRY as *training networks* and *unseen networks*, respectively.

The rest of this chapter is organized as follows. First, we briefly describe the challenges related to prioritization of targets in signaling networks in Section 6.1. Then, we discuss the motivation of TAPESTRY in Section 6.2. As we shall see later, the *target prioritization problem* that TAPESTRY addresses, involves two subproblems: *network matching problem* and *network feature-based ranking problem*. In Section 6.3, we discuss work related to these two subproblems. Next, in Section 6.4, we introduce the Ras activation signaling network that we use as an example of an unseen network. The *target prioritization problem* is formally introduced in Section 6.5. Then, we describe the TAPESTRY algorithm in Section 6.6 and discuss the experimental results in Section 6.7. Finally, Section 6.8 concludes the chapter. Preliminary work of this chapter have been published in [68, 69, 145].

6.1 Background

Target-based drug discovery took prominence after discovery of new drugs using *physiology-based* drug discovery had reached a plateau [312]. The former approach is the process of discovering new drugs that begins with the identification of novel targets [369] whereas the latter approach focusses on using *in vivo* phenotypic screening for drug discovery. However, to date, the shift in paradigm has not resulted in many successful new drugs. One key reason is that novel targets identified have low success rate¹ [312]. Coupled with the length of time needed for drug discovery and potential failure in clinical trials, these targets which have not been validated become unattractive to pharmaceutical companies. Hence, there is a dire need to develop new tools to improve the target identification process in terms of efficiency of target discovery and efficacy of targets identified. High-throughput screening (HTS) technology has been the predominant technology used for screening drug targets [369]. However, the use of HTS technology for combination drugs is still limited as exhaustive measurement of combinations with more than two drugs remains a challenge [410]. In this chapter, we propose

¹Only 3% of novel targets identified reach preclinical phase [369].

TAPESTRY, an *in silico* tool for target identification and prioritization. The targets prioritized by TAPESTRY can be used to guide target combination screening *in vivo* (using targeted gene knockdown in zebrafish [256]), *in vitro* (using HTS platform) or *in silico* (as we shall see in Chapter 7).

6.2 Motivation

In this section, we discuss the motivation of TAPESTRY. First, we justify why network-based approaches are more favourable compared to non-network-centric approaches for prioritization targets. Then, we discuss why supervised machine learning techniques are inappropriate for prioritizing targets. Finally, we discuss limitations of state-of-the-art network-based approaches such as *NetworkPrioritizer*.

Several recent work (reviewed in Chapter 2) have been proposed for target prioritization. Recall that in Chapter 5, we investigated two of these *in silico* approaches, namely, local sensitivity analysis (LSA) [139, 156] and *NetworkPrioritizer* [171], a network-based approach. LSA was found to perform poorly compared to *NetworkPrioritizer*. Briefly, LSA determines the importance of a parameter (*e.g.*, concentration of a particular target) based on its effect on the output of the system (*e.g.*, activity of the output node) when the parameter is perturbed. It is a non-network-centric approach. That is, it does not consider the network characteristics of the targets during analysis, a possible reason for its poor performance. For instance, where is the target in the network? How does the target interact with its neighbours? Signaling networks are by nature complex [214] and diverse [399]. Graphical models with defined kinetics (*e.g.*, ODE model) attempts to capture the complexity of signaling networks. Hence, we advocate that network-based approaches are more appropriate for prioritizing targets in signaling networks.

In the previous chapter, we proposed TENET, a structural feature-based target characterization approach based on SVM. TENET characterizes known targets in signaling networks using structural features and generates a prioritization model based on these features. Note

that although the primary goal of TENET is for target characterization, the supervised learning framework that it employs can also be used for target prioritization by generating a prioritization model for the given unseen network and using that model to prioritize targets for the network. However, known targets, a required input to the TENET algorithm, are often unavailable in unseen networks for two reasons.

- (i) No one has undertaken the task of identifying these targets, or
- (ii) The biological problem is new (*e.g.*, emerging infectious diseases [86, 229]) or understudied (*e.g.*, cellular glucose homeostasis in human [270]) and knowledge of targets is still lacking.

In the former case, there may be abundant information of known targets available in various literature and databases and a known set of targets can still be generated via curation effort that is often tedious and time consuming. For instance, manual curation described in Chapter 5 to generate the benchmark set of known targets for ovarian cancer documented in the *Clinical-Trials* repository [259]. In the latter case, there is simply no relevant information available. For instance, in the case of emerging infectious diseases such as MERS [86], research is still ongoing to learn about the disease and knowledge is limited. Hence, supervised learning based approaches may not be ideal for performing target prioritization.

Although network-based approaches such as *NetworkPrioritizer* have been proposed for target prioritization, these state-of-the-art approaches suffer from two limitations.

First, they require users to specify the appropriate network features and weights to use for target prioritization. An appropriate choice for which features and weights to use may not be immediately apparent and it often relies on an in-depth understanding of the given signaling network. We can relieve the burden of decision making from users by having the application learn the appropriate features and weights to use from examples of signaling networks (training networks) offline. Knowledge gained from the learning process can be used subsequently to

decide on suitable choices of features and weights when prioritizing targets of a given unseen network.

Second, they consider only structural features. Biological systems function through the dynamic interactions of thousands of proteins in a changing environment [358]. Hence, dynamics play an important role in understanding biology [7]. Although topological features are found related to dynamics and modularity in biological networks [285], they are unable to explain the temporal aspects of the networks [288]. For instance, which reactions are rate limiting? Therefore, the two types of features complement each other by providing different perspectives of the biological system [70]. It would be desirable to study a variety of dynamic features similar to our study for structural features in Chapter 5. However, work related to network dynamics are fewer compared to that related to network structure [284]. Moreover, these work focus on modeling the network dynamics and analysis of these models. To the best of our knowledge, there does not exist any appropriate network dynamic feature for use. Hence, we propose *profile shape similarity distance* (PSSD) (Section 6.6.1), a dynamic feature derived from ODEs time-series curves, for our study. In our initial investigations, incorporating PSSD into the prioritization score improves the ROC AUC for some networks modestly. For instance, in the endomesoderm gene regulatory network [196], target prioritization based on a weighted sum score involving both structural features and PSSD resulted in ROC AUC=0.83, a slight improvement over a score based on structural features alone (ROC AUC=0.827). This modest improvement was achieved only when we set a low weight for PSSD and diminished when PSSD was given higher weight (more importance). This implies that the dynamic feature can be used to improve the overall prioritization results and this effect is dependent on its weight which indicates the relative importance of dynamic feature compared to structural feature. The *network feature-based ranking problem* described in Section 6.5.2 addresses the issue of assigning appropriate weights to the structural and dynamic features.

6.3 Related Work

As we shall see later in Section 6.5, the *network matching problem* and the *network feature-based ranking problem* that TAPESTRY addresses can be modelled as the *network similarity ranking problem* and the *rank aggregation problem*, respectively. In this section, we discuss recent approaches pertaining to these two problems. Note that state-of-the-art approaches relating to target prioritization are reviewed in Chapter 2.

6.3.1 Network Similarity Ranking Approaches

First, we provide a brief description of the concept of *network similarity*. Given three signaling networks (MAPK-PI3K [143], endomesoderm gene regulatory [196] and glucose metabolism [192] networks) with known sets of targets, two networks are considered “similar” if they share “similar” targets. Targets are similar if they share commonality in terms of their network features. For instance, the targets in both the MAPK-PI3K and the endomesoderm gene regulatory networks have high eccentricity centrality. In comparison, targets in the glucose metabolism network have high target downstream effect. Hence, the MAPK-PI3K network is similar to the endomesoderm gene regulatory network, but not the glucose metabolism network. Note that in these networks, there is no overlap in the nodes and edges. Thus, distance such as Jaccard index [166] which is based on the overlap between the target sets cannot be used to identify networks with similar targets. Instead, *network similarity ranking* methods such as graphlet degree distribution (GDD) [287] and NETSIMILE [24] that measure network similarity in terms of other aspects of the networks have to be considered. In Section 6.7, we examine three *network similarity ranking* approaches to examine their effect on identifying the *best matched network*.

In the *network similarity ranking problem* [24, 335], given a network G and a set of networks of potentially different sizes and node or edge correspondences, the goal is to rank the set of networks in order of their similarity to G . Network similarity ranking approaches can be broadly classified into *global similarity* measures [128] and *local similarity* measures [287].

The former uses distance measures of the entire network (*e.g.*, degree distribution [128]) while the latter uses measures of portions (neighbourhoods) of the network (*e.g.*, graphlet distribution [287]). Local similarity measures have been proposed to be more appropriate for biological network comparison since these networks tend to be noisy and global distance measures applied onto these incomplete networks may be a distorted representation of the actual network [287]. In this chapter, we examine if given a network, local network similarity approaches can be leveraged to identify the *best matched network* for prioritizing targets. In particular, we study two local network similarity approaches, namely, graphlet degree distribution [287] and NETSIMILE [24], and compare them against our proposed approach FUSSY (FeatUre-based Statistical SimilaritY measure) (described in Section 6.6.3) in terms of their ability to identify the *best matched network* in Section 6.7.2.

Graphlet Degree Distribution (GDD). The graphlet degree distribution proposed by Pržulj and his colleagues in [287] is an extension of the degree distribution. The degree distribution measures the number of nodes having degree k for each value of k while the graphlet degree distribution counts the number of nodes touching a particular graphlet at a particular orbit. Figure 6.1 provides an illustration of the orbits for the different graphlets defined in [287]. The authors define a measure, GDD *agreement*, to determine similarity between two networks.

Given two signaling networks G and H , let $Y_G^j(k)$ denote the fraction of the total area under the GDD curve of network G relevant to automorphism orbit j for degree k . The **GDD distance** of G and H for the j^{th} distribution is defined as

$$D^j(G, H) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{\infty} (Y_G^j(k) - Y_H^j(k))^2}. \quad (6.1)$$

The corresponding **GDD agreement** for the j^{th} orbit is defined as

$$A^j(G, H) = 1 - D^j(G, H) \quad (6.2)$$

Figure 6.1: Automorphism orbits 0 to 72 defined in [287]. Nodes belonging to the same orbit in a graphlet are in the same shade.

where $j \in [0 - 72]$.

Note that a distance of 0 indicates that G and H have identical j^{th} GDD. The agreement between the two networks can be either arithmetic ($A_{arith}(G, H)$) or geometric ($A_{geo}(G, H)$) and are given as:

$$A_{arith}(G, H) = \frac{1}{73} \sum_{j=0}^{72} A^j(G, H) \quad (6.3)$$

$$A_{geo}(G, H) = \sqrt[73]{\prod_{j=0}^{72} A^j(G, H)} \quad (6.4)$$

NETSIMILE. The NETSIMILE algorithm introduced by Berlingerio et al. in [24] consists of three steps: *feature extraction*, *feature aggregation* and *comparison*. In the first step, NETSIMILE extracts 7 local and *egonet*-based features. Given a network $G = (V, E)$, let N_i be the neighbours of node $i \in V$, then the *egonet* of i is a subgraph $I = (V_I, E_I)$ where $V_I = i \cup N_i$ and $E_I \subseteq E$ is the set of all edges connecting nodes u and v such that $u, v \in V_I$. These features and their corresponding notations are listed in Table 6.1. In the next step, the features are aggregated into a “signature” vector by computing their mean, median, standard deviation,

Symbol	Description
d_i	Number of neighbours of node i . Defined as $d_i = N_i $ where N_i denotes the neighbours of i .
κ_i	Clustering coefficient of node i . The definition follows that of $\kappa_{undir(i)}$ in Section 4.1.7.
$\bar{d}_{N_i}$	Average number of node i 's two-hop away neighbours. Defined as $\bar{d}_{N_i} = \frac{1}{d_i} \sum_{j \in N_i} d_j$.
$\bar{\kappa}_{N_i}$	Average clustering coefficient of N_i . Defined as $\bar{\kappa}_{N_i} = \frac{1}{d_i} \sum_{j \in N_i} \kappa_j$.
$ E_{I_i} $	Number of edges in the egonet of node i (I_i).
$ E_{I_i}^o $	Number of outgoing edges from I_i .
$ N_{I_i} $	Number of neighbours of I_i .

Table 6.1: Features extracted by NETSIMILE and their notations.

skewness² and kurtosis³ values. Finally, it compares the “signature” vectors of two networks by computing the *Canberra distance*. Given two ranked lists P and Q of dimension f , with P_i being the i^{th} element of P . Then, the *Canberra distance* between the two lists is defined as

$$d_{Can}(P, Q) = \sum_{i=1}^f \frac{|P_i - Q_i|}{P_i + Q_i} \quad (6.5)$$

A smaller distance implies closer similarity between the two networks.

Note that GDD is proposed for comparing protein-protein interaction (PPI) networks whereas NETSIMILE is for general network comparison. In particular, GDD considers a single type of feature (graphlet degree distribution) whereas NETSIMILE uses seven features (Table 6.1).

Another related field of work is *network alignment* which focusses on the *subgraph isomorphism problem* [195]. That is, given two networks G and H , identify a mapping between a

²The skewness of a variable X , denoted as γ_1 , is the standardized third population moment about the mean and quantifies how symmetrical the distribution is. It is defined as

$$\gamma_1 = \frac{E(X - \mu)^3}{(E(X - \mu)^2)^{\frac{3}{2}}} = \frac{\mu_3}{\sigma^3}$$

where μ is the mean, σ is the standard deviation, μ_3 is the third moment about the mean and E is the expectation operator.

³The kurtosis of a variable X , denoted as β_2 , is the standardized fourth population moment about the mean and quantifies how closely the shape of the distribution matches the Gaussian distribution. It is defined as

$$\beta_2 = \frac{E(X - \mu)^4}{(E(X - \mu)^2)^2} = \frac{\mu_4}{\sigma^4}$$

where μ is the mean, σ is the standard deviation, μ_4 is the fourth moment about the mean and E is the expectation operator.

set of nodes in G and that in H . Network alignment has been applied to solve various problems including discovering biological function [328] and phylogeny [195]. Note that this is different from our focus in this chapter. Although both *network alignment* and *network matching* identify similar networks, the definition of what constitutes similarity is different. In *network alignment*, similarity is in terms of the extent in which nodes in one network can be mapped onto nodes in another network. However, in *network matching*, two given networks are similar as long as their targets share similar network features. A direct mapping between the network targets is not necessary.

6.3.2 Rank Aggregation Approach

A *rank aggregation* problem combines the rankings derived from various approaches to produce a “consensus” ranking. Two classes of approach exist to address this problem. One is based on majority votes while the other is based on some form of rank averaging [280]. The former class of approaches evaluates the ranks using the *Condorcet* criterion. Given a list of individuals containing “X” and “Y”, the *Condorcet* criterion states that if “X” is ranked higher than “Y” more often than not, then “X” should also be ranked higher than “Y” overall. That is, the final “consensus” ranking is based on the number of pairwise wins between the individuals. Some examples of this approach are Copeland’s method [73], Kemeny-Young method [176] and Schulze method [318]. A challenge for these majority vote approaches is the voting paradox where there is no Condorcet winner and tie-breaking rules are usually used to resolve this.

Copeland’s method. Copeland’s method [73] is a natural extension of the Condorcet approach. It ranks the individuals according to the *Copeland score* [306]. Given a set of individuals $C = \{c_1, \dots, c_N\}$, in a pairwise competition between $c_i \in C$ and $c_j \in C$, let the pairwise score be

$$p_{(i,j)} = \begin{cases} 1 & \text{if } c_i \text{ wins } c_j \\ \frac{1}{2} & \text{if } c_i \text{ ties with } c_j \\ 0 & \text{if } c_j \text{ wins } c_i \end{cases}$$

Then, the *Copeland score* of c_i is defined as

$$C(i) = \sum_{j \neq i} p_{(i,j)} \quad (6.6)$$

Kemeny-Young method. The Kemeny-Young method [176] ranks individuals in order to maximize the number of pairwise agreements with the votes. That is, the *Kemeny-Young ranking* should minimize the number of votes that disagree [72]. Given a set of individuals $C = \{c_1, \dots, c_N\}$, a ranking r and a vote v , let the pairwise score of any two distinct individuals c_i and c_j in C be

$$q_{i,j}(r, v) = \begin{cases} 1 & \text{if both } r \text{ and } v \text{ ranks } c_i \text{ higher than } c_j \\ 1 & \text{if both } r \text{ and } v \text{ ranks } c_j \text{ higher than } c_i \\ 0 & \text{otherwise} \end{cases}$$

Then, the *Kemeny-Young ranking* maximizes

$$\sum_v \sum_{c_i, c_j \in C} q_{i,j}(r, v) \quad (6.7)$$

Schulze method. The Schulze method [318] ranks individuals according to the *strength* of the voters' preferences. Given a set of individuals $C = \{c_1, \dots, c_N\}$, let $d[c_i, c_j]$ be the number of voters who strictly prefer $c_i \in C$ to $c_j \in C$, and $Path_{c_i, c_j} = (P(1), \dots, P(n))$ be a set of $|C|$ candidates ordered according to voter's preferences starting with c_i and ending with c_j where $P(i) \in C$, $c_i = P(1)$ and $c_j = P(n)$. The *strength* of $Path_{c_i, c_j}$ is defined as

$$s[c_i, c_j] = \max\{\min\{d[P(i), P(i+1)] - d[P(i+1), P(i)] \mid i = 1, \dots, (n-1)\} \mid P(1), \dots, P(n) \text{ is a path from } c_i \text{ to } c_j\}. \quad (6.8)$$

The latter class of approaches uses some form of rank averaging and includes approaches such as the Borda method [85] and weighted-sum [112].

Borda method. The Borda method [85] ranks individual according to the *Borda count* which is based on the score assigned to the individual according to its position in the ranked list. This

approach sometimes elects broadly acceptable individuals rather than those having the majority votes. Given a set of individuals $C = \{c_1, \dots, c_N\}$ and a set of ranked list $R = \{r_1, \dots, r_M\}$ associated to C , let $r_i^{c_j}$ denote the rank value of $c_j \in C$ based on the ranked list $r_i \in R$. Then, the *Borda count* of c_j is defined as

$$B_{c_j} = \sum_{i=1}^M r_i^{c_j} \quad (6.9)$$

Weighted-sum. The weighted-sum approach [112] is a popular rank averaging-based approach which ranks the individuals based on the *aggregated rank value*. Given a set of individuals $C = \{c_1, \dots, c_N\}$ and a set of ranked list $R = \{r_1, \dots, r_M\}$ associated to C , let $r_i^{c_j}$ denote the rank value of $c_j \in C$ based on the ranked list $r_i \in R$ and w_i denote the weight of the ranked list r_i . Then, the *aggregated rank value* of c_j is defined as

$$A_{c_j} = \sum_{i=1}^M w_i r_i^{c_j} \quad (6.10)$$

In TAPESTRY, two ranked lists have to be aggregated. These two lists are the SFB rank and DFB rank. Hence, the situation is analogous to a voting system with only two voters where no clear majority vote is likely. Thus, approaches based on rank averaging are more appropriate. In this chapter, we use the weighted-sum approach because it is commonly used, easy to implement and the weights reflects the relative importance of the ranked lists being aggregated [178].

6.4 Unseen Network Example: Ras Activation Network

In this section, we first describe the Ras activation signaling network which we use as an example of an unseen network. Then, we describe the curation process used to obtain its targets. Recall that the unseen network do not necessarily have known targets. The curated targets obtained for the Ras activation network are for the purpose of validating our approach later (Section 6.7).

The Ras activation network [105] (Figure 6.2) describes the phospholipase C(PLC)- ϵ -driven compartment switching of Ras activation at the plasma membrane and the Golgi. Ras activation mutation is found to be present in various cancers, including ovarian cancer [301]. The model in [105] is adapted from that in [26] which fitted well for experimental results in different cancer cell lines (PC12 and NIH 3T3 cells). Hence, the model can be considered a “general” Ras activation network in cancer cells. In particular, we associate this network with ovarian cancer so as to obtain an additional ovarian cancer-related network with different topology for our study in this thesis. We choose the guanosine triphosphate (GTP)-bound Ras located on the plasma membrane (RasGTP-PM) as the output node for this network. GTP binding activates Ras and Ras localization in the plasma membrane is necessary for activating its downstream effectors [2]. The ODE model (BIOMD000000161) of this network can be obtained from the *BioModel* repository [201]. This model contains 46 nodes and 43 hyperedges. Similar to other signaling networks, it contains feedback and feedforward loops. The characteristics of this network are summarized below:

- (i) Average number of neighbors for each node: 3
- (ii) Simple feedback loops (2 nodes): 1
- (iii) Simple feedback loops (3 nodes): 1
- (iv) Coupled feedback loops (>3 nodes): 1 (11-node), 1 (17-node)
- (v) Simple feedforward loops (3 nodes): 0
- (vi) Coupled feedforward loops (>3 nodes): 1 (30-node)

Note that the Ras activation network has exactly the same number of feedback and feedforward loops as the MAPK-PI3K network.

Recall that there is a lack of gold standard sets of benchmark targets relevant to the networks we study. Hence, we perform manual target curation to create this set of benchmark targets.

Ovarian Cancer Drugs in [259]	Mechanism of Action	Target
Carboxyamidotriazole (Phase II) [257]	Binds to and inhibits non-voltage-operated Ca^{2+} channels, blocking both Ca^{2+} influx into cells and Ca^{2+} release from intracellular stores	Ca^{2+}
Hydralazine (Phase III) [17, 327]	Increase Ca^{2+} inflow due partly to activation of beta-adrenoceptors, with resultant elevation of cyclic AMP, and partly to another mechanism	Ca^{2+}
EGFR antisense DNA (Phase I) [257]	A synthetic sequence of DNA constructed in the antisense orientation to a sequence of DNA in epidermal growth factor receptor (EGFR), a member of the erbB gene family. EGFR antisense DNA suppresses the expression of EGFR by tumor cells, thereby inhibiting tumor cell proliferation and decreasing tumor growth	EGFR
Lapatinib (Phase I) [289]	Bind to ATP-binding site of EGFR, preventing its autophosphorylation	activated EGFR
Matuzumab (Phase II) (EMD 72000) [257]	Binds to EGFR on tumor cells and blocks growth signals	EGF:EGFR
Erlotinib (Phase II) [257]	Competing with ATP, erlotinib reversibly binds to the intracellular catalytic domain of EGFR, thereby reversibly inhibiting EGFR phosphorylation and blocking the signal transduction events and tumorigenic effects associated with EGFR activation	EGF:EGFR
Gefitinib (Phase II) [257]	Inhibits the catalytic activity of numerous tyrosine kinases including EGFR, which may result in inhibition of tyrosine kinase-dependent tumor growth	active EGFR
Iressa (Phase II) [257]	Competes with the binding of ATP to the tyrosine kinase domain of EGFR, thereby inhibiting receptor autophosphorylation and resulting in inhibition of signal transduction	active EGFR
Tristantumumab (Phase II) [393]	Binds EGFR with high affinity and inhibits ligand-dependent receptor activation	EGF:EGFR
Vandetanib (Phase II) [367]	An ATP-competitive inhibitor of RET, EGFR, and VEGFR	active EGFR
Cetuximab (Phase II) [20]	EGFR inhibitor. Cetuximab is a human:mouse chimeric monoclonal antibody that binds with high specificity to the extracellular domain of EGFR and prevents receptor dimerization and signalling	dimerized EGFR
Ras peptide cancer vaccine (Phase II) [257]	Mutant Ras oncogenes produce novel proteins that are processed and displayed through HLA molecules on tumor cells. Vaccine is targeted at tumour cells carrying the Ras mutation	tumor cells expressing mutant Ras

Table 6.2: Ovarian cancer drugs in [259] and their targets in the Ras- activation network [105].

The target curation process for the Ras activation network is same as that for the MAPK-PI3K network (described in Section 4.3.1) since they are both implicated in ovarian cancer. Table 6.2 lists the targets that are relevant to this network.

6.5 Problem Statement

In this section, we shall describe the *target prioritization problem* which TAPESTRY addresses. This problem consists of two subproblems:

- (i) *Network matching problem*: for a given unseen network and a set of training networks, identify the *best matched network* from the training network set, and
- (ii) *Network feature-based ranking problem*: for a given unseen network with a set of SFB ranks and a set of DFB ranks, rank the targets in the unseen network.

Note that TAPESTRY uses the prioritization model of the best matched network found in (i) to generate SFB ranks for targets in the unseen network. The *network feature-based ranking problem* examines the issue of *integrating* SFB ranks with DFB ranks (Section 6.6.1) to prioritize targets in the unseen network. We shall now describe these two problems.

6.5.1 Network Matching Problem

Recall that the targets of the training network are known whereas those of the unseen network may not be known. Intuitively, the goal of the *network matching problem* is to find a training network (*best matched network*) whose targets have network features that are likely to be similar to those of the given unseen network. The network feature-based target prioritization model of the *best matched network* can then be used to prioritize the targets of the unseen network in an attempt to identify potential targets. Hence, the *network matching problem* is related to the network similarity ranking problem described in Section 6.3.1.

Definition 6.14 Given an unseen network G and a set of training networks $\mathcal{L} = \{L_1, \dots, L_N\}$, *network matching* calculates a **network matching distance** $D(G, L_i)$ between G and $L_i \in \mathcal{L}$. Comparing two networks, L_i and L_j where $L_i, L_j \in \mathcal{L}$, L_i is more similar to G if $D(G, L_i) < D(G, L_j)$. The **best matched network** $L_k \in \mathcal{L}$ of G is the network with the smallest network matching distance. That is, $\forall i : D(G, L_k) < D(G, L_i)$.

Ideally, the best matched network has a prioritization model that when applied onto the unseen network, yields a target prioritization result with the best ROC AUC. Different distance measures can be defined for determining network similarity. Examples are GDD agreement and Canberra distance of the NETSIMILE “signature” vectors. However, these state-of-the-art measures ignore dynamic features which form an important part of signaling networks. Hence, we propose FUSSY in Section 6.6.3 which uses structural features and a dynamic feature for measuring network similarity based on the similarity in network feature distribution (*feature distribution similarity*).

6.5.2 Network Feature-Based Ranking Problem

In the *network feature-based ranking problem*, the goal is to rank the nodes using structural and dynamic features. Hence, this problem is equivalent to a rank aggregation problem (Section 6.3.2) [98, 224] where SFB ranks and DFB ranks are aggregated to produce the *putative target score* for prioritizing targets. We use the weighted-sum approach to aggregate the ranks for reasons described in Section 6.3.2. The weighted-sum approach allows poor performance in one criterion to be compensated by good performance in other criteria, resulting in approximate ranking and hence, approximate prioritization. The approximate prioritization is good enough if it can prioritize a majority of known targets over non-target nodes.

Definition 6.15 Given a signaling network $G = (V, E)$ and an output node $x \in V$, let D and S denote the set of DFB ranks and SFB ranks of V , respectively. Then, the **putative target score** of a node $v_i \in V$ is defined as

$$\rho_{v_i} = \varpi_1 \frac{D_{v_i}}{\text{MAX}_{u \in V}(D_u)} + \varpi_2 \frac{S_{v_i}}{\text{MAX}_{u \in V}(S_u)} \quad (6.11)$$

where $\varpi_1 + \varpi_2 = 1$ and $\text{MAX}(\cdot)$ is the maximum operator. **Network feature-based ranking** orders the nodes in V based on decreasing ρ such that for two nodes $v_i, v_j \in V$, v_i is ranked higher than v_j if $\rho_{v_i} > \rho_{v_j}$.

The weights ($\overline{w}_1$ and $\overline{w}_2$) affect the putative target score and prioritization of targets and we study this effect in Section 6.7. Note that the ratio of weights provides an indication of the relative influence of structural and dynamic features in prioritizing targets.

6.6 Target Prioritization

Recall that in this chapter, we shall use *profile shape similarity distance* (PSSD), a dynamic feature, subsequently for target prioritization by TAPESTRY. We begin this section with an introduction to PSSD. Then, we provide an overview of TAPESTRY and describe FUSSY, our proposed network similarity approach that facilitates identification of a suitable prioritization model in the *Target Characterization Knowledge Base*. Recall that in Chapter 3 the prioritization model is used for generating SFB ranks of the unseen network. These SFB ranks shall be subsequently combined with the DFB ranks for target prioritization. Next, we describe the TAPESTRY algorithm and provide an illustrating example using the Ras activation network. Finally, we perform complexity analysis on the TAPESTRY algorithm.

6.6.1 Profile Shape Similarity Distance (PSSD)

We first describe PSSD in terms of its biological relevance in signaling networks. Then, we formally define PSSD and provide a small example to illustrate the calculation of this feature.

In signaling networks, signal responses to perturbation are typically measured in terms of the time-varying concentrations of signaling proteins such as kinases and phospho-proteins [194]. These time-varying concentrations are represented as concentration-time profiles (Figure 6.3). Time profile comparison is frequently used to identify node regulatory relationships for gene regulatory [374] and metabolic networks [302] for purpose such as elucidating network conductivities. There are certain considerations in comparing these phosphoprotein concentration-time profiles. In signaling networks, reactions occur at different and non-uniform rates [1] resulting in profiles with variable time delays. Hence, a distance measure

Figure 6.3: Some concentration-time profiles of the MAPK-PI3K network.

based on one-to-one alignment on a time axis (Euclidean) is ineffective at detecting similarity in these profiles. A non-linear measure, such as *dynamic time warping* (DTW) distance, allows a more intuitive alignment between profiles [177] and is more suitable for biological time series data such as gene and protein expression [1]. As we shall see later, DTW distance identifies the profiles of Akt and ERKPP in Figure 6.3 as similar whereas Euclidean distance did not (Section 6.7).

Definition 6.16 Given two discrete time series ϕ_u and ϕ_v , the *dynamic time warping distance* between them is defined recursively as:

$$\text{DTW}(\phi_u, \phi_v) = \xi(\text{First}(\phi_u), \text{First}(\phi_v)) + \text{Min} \begin{cases} \text{DTW}(\phi_u, \text{Rest}(\phi_v)) \\ \text{DTW}(\text{Rest}(\phi_u), \phi_v) \\ \text{DTW}(\text{Rest}(\phi_u), \text{Rest}(\phi_v)) \end{cases}$$

where $\text{First}(\phi_u) = \{\tau_{u[1]}\}$, $\text{Rest}(\phi_u) = \{\tau_{u[2]}, \tau_{u[3]}, \dots, \tau_{u[n]}\}$, $\xi(\tau_{u[i]}, \tau_{v[j]}) = (\tau_{u[i]} - \tau_{v[j]})^2$ and $\tau_{u[i]}$ is the value of ϕ_u at time point i .

Although DTW distance is robust to time warping, it can miss similar profiles that have undergone y-axis warping [177] and inversely similar profiles. These profiles are common

in signaling networks where signals may be amplified or attenuated as they pass through the signaling cascade, partly due to the relative levels of activators and inhibitors within the cascade [14]. Moreover, profiles of nodes tend to be similar to their activators and *inversely similar* to their inhibitors [339]. Given a concentration-time profile ϕ_u having n time points, denoted as $\phi_u = \{\tau_{u[0]}, \dots, \tau_{u[n]}\}$, let m be the median value of ϕ_u . The corresponding **inverted profile** is denoted as ϕ'_u where the i^{th} time point of ϕ'_u is defined as $\tau'_{u[i]} = 2 \times m - \tau_{u[i]}$. Hence, we extend DTW distance to address these limitations. The profiles are Z-normalized using $ZNormalize(\phi_u) = \frac{\tau_{u[i]} - \overline{\phi_u}}{\sigma(\phi_u)}$, $\forall \tau_{u[i]} \in \phi_u$ where $\sigma(\phi_u)$ and $\overline{\phi_u}$ are the variance and mean of ϕ_u , respectively, to minimize the effects of y-axis warping. For inversely similar profiles (ϕ'_u), $DTW(\phi'_u, \phi_v)$ will yield a lesser value than $DTW(\phi_u, \phi_v)$. Hence, both DTW distances will be computed and the lesser distance selected as the PSSD. We note that the profiles have to be non-stationary for meaningful DTW distance comparison. Since signaling networks pass along biological signals by modifying concentrations of phospho-proteins [194], we expect most of these profiles to be non-stationary.

Definition 6.17 Given a signaling network $G = (V, E)$, let ϕ_u be the Z-normalized concentration-time profiles of node $u \in V$. The **profile shape similarity distance** of between nodes u and $v \in V$ is defined as

$$\Phi_v^u = \text{Min}(DTW(\phi_u, \phi_v), DTW(\phi'_u, \phi_v)) \quad (6.12)$$

where $\text{Min}(\cdot)$ is the minimum operator and $DTW(\cdot)$ is the function for calculating the dynamic time warping distance.

For example, to calculate $\Phi_{\text{ERKPP}}^{\text{Akt}}$, ϕ_{ERKPP} and ϕ_{Akt} are Z-normalized using $ZNormalize(\phi)$ (Figure 6.3). Then, ϕ'_{Akt} is obtained by inverting ϕ_{Akt} . Finally, $DTW(\phi_{\text{Akt}}, \phi_{\text{ERKPP}})$ and $DTW(\phi'_{\text{Akt}}, \phi_{\text{ERKPP}})$ are calculated and the smaller value is assigned as $\Phi_{\text{ERKPP}}^{\text{Akt}}$. The *Dynamic Feature Ranker Module* uses the PSSD values to rank the nodes in the unseen signaling network. Nodes are ordered in increasing PSSD values since smaller PSSD imply closer similarity between the profiles.

6.6.2 Overview of TAPESTRY

TAPESTRY, our proposed approach for target prioritization, consists of three phases, namely, *candidate filtering*, *best matched network identification* and *target prioritization*.

During the *candidate filtering* phase, nodes that do not have paths leading to the output node are removed. This phase is intended to reduce the computation required by ensuring that analysis is performed only on potential targets.

Then, in the *best matched network identification* phase, a network similarity ranking approach is used to find the best matched network for the unseen network. The prioritization model of the best matched network is used for generating the SFB ranks. Recall that we consider a total of 16 structural features and using irrelevant features can impact the prioritization results negatively. This phase is intended to determine the most appropriate structural features and weights to use for computing SFB ranks for the given unseen network. Our strategy is to perform learning on examples of signaling networks with known targets offline to identify structural features and weights that best characterizes these known targets (captured in prioritization models generated by TENET in Chapter 5) and use these prioritization models to perform target prioritization. In Section 6.7, we examine three different network similarity ranking approaches, namely, GDD, NETSIMILE and FUSSY. The first two approaches are described in Section 6.3.1 and FUSSY is described in Section 6.6.3. We shall now discuss about the prioritization model before proceeding to describe the *target prioritization* phase.

Recall that in Chapter 5, we examine two ways of learning the prioritization model:

- (i) Approach 1: Learn a generalized prioritization model and associated predictive structural feature set from all training networks. That is, the training networks are combined as a single data set for training the prioritization model.
- (ii) Approach 2: Learn a prioritization model and associated predictive structural feature set for each training network. This generates a set of prioritization models and predictive structural feature sets.

In reality, a third approach involving categorizing the training networks and generating a category-based prioritization model can also be considered. However, in this thesis, we consider only four training networks⁴ and this small number of networks may not yield categories of reasonable size. In the end, each category may contain only one or two signaling networks. Hence, for small number of training networks, there is insignificant difference between using the category-based approach and Approach 2. Henceforth, we shall refer to the prioritization models derived from Approach 1 and Approach 2 as *combined network-based* (CNB) and *individual network-based* (INB) prioritization models, respectively.

In an ideal situation, it is desirable to have a single prioritization model which can be used for generating SFB ranks for all unseen networks in TRECENTO. This situation implies that targets in all signaling networks share the same network characteristics (*e.g.*, high bridging centrality). In particular, the CNB prioritization model in Approach 1 is akin to the single prioritization model in the ideal situation, which if it exists, should yield good prioritization results on the test set. However, our study in the preceding chapter reveals that CNB prioritization models perform poorer than INB prioritization models. Recall that across the four training networks we studied (Table 5.13), the test set prediction accuracy of the best approaches fluctuated in the ranges [0.803–1] and [0.609–0.78] for INB and CNB prioritization models, respectively. In particular, sensitivity of the CNB prioritization models ([0.313–0.602]) is significantly lower than that of the INB prioritization models ([0.905–1]). In other words, known targets are often missed by CNB prioritization models. Such poor sensitivity renders Approach 1 ineffective for applications such as drug target identification since random guessing may do better. Hence, we shall focus on Approach 2. That is, given an unseen network and a set of training networks, TAPESTRY shall identify the training network that is most “similar” to the unseen network (best matched network) and use its prioritization model for generating SFB ranks of the unseen

⁴Additional network is infeasible due to time constraint as target curation requires considerable amount of time.

network. In particular, the network matching problem described in Section 6.5.1 addresses the issue of identifying a similar network.

Finally, in the *target prioritization* phase, the SFB rank and DFB rank of the candidate targets are generated and integrated to obtain the putative target score that shall be used for prioritizing the targets. Note that besides structure, dynamics constitute an important part of signaling networks. Hence, we reason that incorporating dynamic features can improve target prioritization. The end result of TAPESTRY is a list of potential targets that are prioritized based on structural and dynamic features.

6.6.3 Proposed Network Similarity Ranking: FUSSY

Feature Distribution Similarity. TRECENTO uses several network features for prioritizing targets in the signaling network. It will be interesting to find out if these features can also be used to determine the similarity between networks in order to identify the best matched network. Intuitively, two networks are similar if they have similar network features (*e.g.*, exhibit similar distributions). FUSSY uses the notion of *feature distribution similarity* to assess network similarity. Given a network G and a set of reference network $\mathcal{L} = \{L_1, \dots, L_N\}$, G is more similar to L_i than L_j if G exhibits greater similarity to L_i than to L_j in more network features. One method of assessing feature similarity is to examine the feature distribution, which allows networks of different sizes to be compared. Statistical measures that can be used for assessing such similarity are Wilcoxon Rank-Sum (Wilcoxon) [226] and Kolmogorov-Smirnov (KS) [341]. These tests are nonparametric. Hence, they can be applied to features with distribution that are unknown a priori. In these tests, the null hypothesis that the distributions are the same is tested against the alternative hypothesis that the distributions are different. These statistical tests yield a set of p -values which can be aggregated into a combined p -value using *Fisher's inverse χ^2 method* (Fisher's) [279] or *Stouffer's method* (Stouffer's) [343]. The former approach is based on the χ^2 distribution while the latter approach is based on the standard normal curve. Given K p -values, the combined p -value obtained using the Fisher's and Stouffer's

Notation	Baseline Approach	Features Considered	Distance $D(\cdot)$
GDD_A	GDD	73 automorphism orbits	$A_{arith}(\cdot)$
GDD_G	GDD	73 automorphism orbits	$A_{geo}(\cdot)$
NETSIMILE	NETSIMILE	Features in Table 6.1	$d_{Can}(\cdot)$
$FUSSY_{(A,F)}$	FUSSY	All features in Section 4.1 and PSSD	$p_f(\cdot)$
$FUSSY_{(A,S)}$	FUSSY	All features in Section 4.1 and PSSD	$p_s(\cdot)$
$FUSSY_{(S,F)}$	FUSSY	All features in Section 4.1	$p_f(\cdot)$
$FUSSY_{(S,S)}$	FUSSY	All features in Section 4.1	$p_s(\cdot)$
$FUSSY_{(D,F)}$	FUSSY	PSSD	$p_f(\cdot)$
$FUSSY_{(D,S)}$	FUSSY	PSSD	$p_s(\cdot)$

Table 6.3: Summary of the various network similarity approaches and variants studied.

approaches, denoted as p_f and p_s , respectively, are defined as:

$$p_f = Pr[\chi^2(2K) \geq -2 \sum_{k=1}^K \log(p_k)] \quad (6.13)$$

$$p_s = Pr[Z \leq \frac{1}{\sqrt{K}} \sum_{k=1}^K \Gamma^{-1}(p_k)] \quad (6.14)$$

where p_k is the p -value of the k^{th} test, $\Gamma^{-1}(\cdot)$ is the inverse of the standard normal cumulative distribution function and $Z \sim N(0, 1)$. The combined p -value is the distance used for measuring network similarity in FUSSY.

We generate different variants of FUSSY (summarized in Table 6.3) for our study in this chapter using various network features and methods for combining the p -values. Note that FUSSY is different from state-of-the-art approaches (GDD and NETSIMILE). These state-of-the-art approaches rank all available networks. However, this may be undesirable in the context of target prioritization if none of the networks are similar enough to produce good prioritization in the given network. In addition, these approaches generally use features based on network topology and not dynamics. Recall that both structural and dynamic features are important for understanding the biological systems. Hence, ignoring one or the other may result in a suboptimal network similarity comparison. FUSSY addresses these limitations by filtering out networks that are different and by considering both structural and dynamic features. As we shall see in Section 6.7, profile shape similarity distance (PSSD), a dynamic feature, plays an

Algorithm 8: Algorithm FUSSY**Input:** Unseen network G , set of training networks $\mathcal{L}$ and relaxation parameter r (optional).**Output:** Best matched network G_{best} and network ranked list R .

```

1  $G_{best} \leftarrow \text{null}$ 
2  $\mathcal{X}_{best}, P_{best}, pFunc, H, Truth \leftarrow \text{LEARNBESTFEATURES}(\mathcal{L});$  /* Phase 1.1 */
3  $p_t \leftarrow \text{LEARNPTHRESHOLD}(Truth, P_{best}, r);$  /* Phase 1.2 */
4  $G_{best}, R \leftarrow \text{GETBESTNETWORK}(G, \mathcal{L}, \mathcal{X}_{best}, pFunc_{best}, p_t, H);$  /* Phase 2 */

```

important role in identifying signaling networks with similar targets.

The FUSSY Algorithm. Algorithm 8 outlines FUSSY which extracts the structural and dynamic features we study in this paper to generate a vector “signature” for comparing networks. Given an unseen network G , a set of training networks $\mathcal{L}$ and a relaxation parameter r , FUSSY returns the best matched network G_{best} through the following steps. First, it initializes the best matched network G_{best} (Line 1). Then, it proceeds with the two phases, namely, the *learning phase* (Lines 2 and 3) and the *network ranking phase* (Line 4) which we elaborate in turn. Note that r is an optional input and is used for relaxing the criteria for filtering out dissimilar networks in the second phase.

Phase 1: Learning. The learning phase is split into two stages (LEARNBESTFEATURES (Algorithm 9) and LEARNPTHRESHOLD procedures) and we shall describe them in turn.

The goal of Algorithm 9 is to learn which type of features (structural, dynamic, or both) and which method for combining p -values (FISHER(.) or STOUFFER(.)) are the most relevant for finding the best matched network. First, the algorithm initializes a vector $\mathcal{A}$ which contains all the variants of FUSSY (Line 1). For instance, the first element of $\mathcal{A}$ ($\mathcal{A}_0$) contains the list of all features $\mathcal{X}_{all}$ and FISHER(.) whereas the fourth element of $\mathcal{A}$ ($\mathcal{A}_3$) contains the list of all the structural features $\mathcal{X}_{structure}$ and STOUFFER(.). Then, it generates the true ranking order (referred to as ground truth) of the training networks (Line 3) which is subsequently used to find the best variant in $\mathcal{A}$ for the given set of training networks. For each training network $\mathcal{L}_i$, the prioritization model of the remaining training networks $\mathcal{L} \setminus \mathcal{L}_i$ are applied to

Algorithm 9: Procedure LEARNBESTFEATURES (Phase 1.1)**Input:** Set of training networks $\mathcal{L}$.**Output:** Best feature set $\mathcal{X}_{best}$; $|\mathcal{L}| \times |\mathcal{L}| - 1$ matrix of p -values for training networks using the best feature set P_{best} ; best function for combining p -values $pFunc$; $|\mathcal{X}_{all}| \times |\mathcal{L}|$ matrix of all features values in all training networks H ; and the ground truth $Truth$.

```

1  $\mathcal{A} \leftarrow \text{INITIALIZE}(\{\mathcal{X}_{all}, \mathcal{X}_{structure}, \mathcal{X}_{dynamic}\}, \{\text{FISHER}(\cdot), \text{STOUFFER}(\cdot)\})$ 
2  $H \leftarrow \text{EXTRACTFEATURES}(\mathcal{L}, \mathcal{X}_{all})$ 
3  $Truth \leftarrow \text{GETGROUNDTRUTH}(\mathcal{L}, H)$ 
4 foreach iteration  $i=1$  to  $|\mathcal{L}|$  do
5   foreach iteration  $j=i+1$  to  $|\mathcal{L}|$  do
6     foreach iteration  $k=1$  to  $|\mathcal{A}|$  do
7        $M \leftarrow \text{GETRELEVANTFEATURES}(H, \mathcal{A}_k)$ 
8        $cFunc \leftarrow \text{GETRELEVANTPCOMBINEMETHOD}(\mathcal{A}_k)$ 
9        $P_W \leftarrow \text{WILCOX}(\mathcal{L}_i, \mathcal{L}_j, M)$ 
10       $P_{KS} \leftarrow \text{KS}(\mathcal{L}_i, \mathcal{L}_j, M)$ 
11       $PValue_i(k, \mathcal{L}_j) \leftarrow \text{COMBINEP}(P_W, P_{KS}, cFunc)$ 
12 foreach iteration  $i=1$  to  $|\mathcal{A}|$  do
13   foreach iteration  $j=1$  to  $|\mathcal{L}|$  do
14      $RelativeRank_{i,j} \leftarrow \text{RANK}(PValue, i, j)$ 
15    $Distance_i \leftarrow \text{SUM}(\text{SPEARMAN}(RelativeRank, i), \text{KENDALL}(RelativeRank, i), Truth_i)$ 
16  $\mathcal{X}_{best}, P_{best}, pFunc, H, Truth \leftarrow \text{GETBESTFEATURES}(Distance, PValue, \mathcal{A})$ 

```

$\mathcal{L}_i$ to obtain the prioritized node set of $\mathcal{L}_i$. ROC analysis is performed on the prioritized node set and networks in $\mathcal{L} \setminus \mathcal{L}_i$ are ordered in decreasing ROC AUC to produce the ground truth. Next, the procedure extracts all the features for all training networks into a $|\mathcal{X}_{all}| \times |\mathcal{L}|$ matrix H (Line 2). Following which, it computes the combined p -value for the Wilcoxon and KS tests for each FUSSY variant (Lines 7 to 11). Then, for each training network $\mathcal{L}_i$, six ranked lists (*RelativeRank*), each corresponding to one variant⁵ of FUSSY, are obtained by ordering the remaining training networks in decreasing⁶ order of the combined p -value (Line 14). For instance, for the MAPK-PI3K network, the order of the remaining training networks is: $I_4 (0.128) \succ I_2 (6.52 \times 10^{-4}) \succ I_3 (2.2 \times 10^{-16})$ for dynamic feature using FISHER($\cdot$) for combining the p -values. The combined p -values are indicated in round brackets. Next, the disagreement between *RelativeRank* and *Truth* is measured using the *Spearman footrule distance*

⁵The different variants of FUSSY are based on the feature set and method for combining p -values and are listed in Table 6.3.

⁶Smaller p -value implies a higher probability that the null hypothesis (same feature distribution) is false.

Algorithm 10: Procedure GETBESTNETWORK (Phase 2)

Input: Unseen network G , a set of training networks $\mathcal{L}$, best feature set $\mathcal{X}_{best}$, best function for combining p -values $pFunc$, threshold p -value p_t and $|\mathcal{X}_{all}| \times |\mathcal{L}|$ matrix of all features values in all training networks H .

Output: Best matched network G_{best} and network ranked list R .

```

1  $H \leftarrow \text{EXTRACTFEATURES}(G, \mathcal{X}_{best})$ 
2 foreach iteration  $i=1$  to  $|\mathcal{L}|$  do
3    $P_W \leftarrow \text{WILCOX}(G, \mathcal{L}_i, H)$ 
4    $P_{KS} \leftarrow \text{KS}(G, \mathcal{L}_i, H)$ 
5    $P_{(G, \mathcal{L}_i)} \leftarrow \text{COMBINEP}(P_W, P_{KS}, pFunc)$ 
6  $G_{best}, R \leftarrow \text{GETBESTTRAININGNETWORK}(\mathcal{L}, P, p_t)$ 

```

(SPEARMAN procedure) [91] and *Kendall-Tau distance* (KENDALL procedure) [98], denoted as $\Upsilon(\cdot)$ and $\Lambda(\cdot)$, respectively. Given two complete rankings r_1 and r_2 of a set of N individuals, let $r_1(i)$ be the rank of $i \in N$ in the ranked list r_1 , the distances are calculated as:

$$\Upsilon(r_1, r_2) = \sum_{i \in N} |r_1(i) - r_2(i)| \quad (6.15)$$

$$\Lambda(r_1, r_2) = |\{i, j\} : r_1(i) < r_1(j) \text{ and } r_2(i) > r_2(j)| \quad (6.16)$$

For each variant of FUSSY, the Spearman footrule and Kendall-Tau distances for all the training networks are summed to obtain the overall distance (Line 15). The best variant is the one yielding the least overall distance (Line 16). The corresponding feature set, method for combining p -values and set of combined p -values are then selected as $\mathcal{X}_{best}$, $pFunc$ and P_{best} , respectively.

The LEARNPTHRESHOLD procedure (Line 3 in Algorithm 8) learns p_t which is used to determine dissimilar networks which are filtered off in the next phase. First, it obtains a mapping between the combined p -values and “poor” ROC AUC. In this chapter, we deem ROC AUC < 0.5 as “poor” since it indicates performance worse than random prioritization. Then, the combined p -values in this mapping are averaged to obtain p_t . Finally, p_t is set as the minimum of p_t and r (the optional relaxation parameter) since combined p -values less than p_t are filtered off.

Phase 2: Network Ranking. In this phase (Algorithm 10), the best matched network G_{best} is returned together with the ranking of the training networks R with combined p -values above p_t . First, values of the features in $\mathcal{X}_{best}$ is extracted for G and stored in H (Line 1). Then, for each pair of $(G, \mathcal{L}_i)$, the Wilcoxon and KS tests are performed for each feature in $\mathcal{X}_{best}$ (Lines 3 to 4). The resulting p -values are combined using $pFunc$ (Line 5). Finally, in the GETBESTTRAININGNETWORK procedure (Line 6), the training networks with combined p -values greater than or equal to p_t are ordered in decreasing combined p -values to obtain R . The top rank network is G_{best} . Note that, it is possible for FUSSY to return no best matched network if all the combined p -values are less than p_t . This indicates that none of the given training networks are similar. Then, we can either explore additional training networks to identify G_{best} or relax p_t using r to obtain a suboptimal G_{best} .

Theorem 6.3 *Given an unseen network $G = (V, E)$ and a set of reference networks $\mathcal{L}$, the time complexity of FUSSY is $O(|\mathcal{L}|(|\mathcal{L}| - 1)(O(\mathcal{G}(\mathcal{X}_{all})) + |\mathcal{A}|(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2))$ time in the worst case, where $\mathcal{X}_{all}$ is the set of network features, $|V_{\mathcal{L}[i]}|$ is the number of nodes of the i^{th} network in $\mathcal{L}$ and $|\mathcal{A}|$ is the number of FUSSY variants.*

Proof: The Algorithm 8 is made up of three main procedures LEARNBESTFEATURES, LEARNPTHRESHOLD and GETBESTNETWORK.

In procedure LEARNBESTFEATURES, $O(|\mathcal{L}|(|\mathcal{L}| - 1)(O(\mathcal{G}(\mathcal{X}_{all}))))$ time is required for extracting all features in $\mathcal{L}$ where $O(\mathcal{G}(\mathcal{X}_{all}))$ is the worst time complexity for extracting all features. Next, to obtain the ground truth, it takes $O(|\mathcal{L}|(|\mathcal{L}| - 1)(O(\mathcal{R}(.))))$ time to generate the rankings using support vector regression, $O(|\mathcal{L}||V_{\mathcal{L}[i]}|^2)$ time to perform ROC ANALYSIS [110] and $|\mathcal{L}|(|\mathcal{L}| - 1)\log(|\mathcal{L}| - 1)$ time to obtain the ground truth for every training network. Note that $O(\mathcal{R}(.))$ is the time complexity of computing the support vector regression and is equals to the number of support vector ($O(|\mathbf{v}|)$) [25]. In the worst case, $|\mathbf{v}| = |V_{\mathcal{L}[i]}|$. Given that each Wilcoxon test takes $O((|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2)$ time [253]; each KS test takes $O(|V_{\mathcal{L}[i]}|)$

[132] time and the calculation of the combined p -values requires $O(|\mathcal{X}_{all}|)$ time in the worst case, the computation of $PValue$ requires $O(|\mathcal{A}||\mathcal{L}|(|\mathcal{L}| - 1)((|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2 + |\mathcal{X}_{all}| + |V_{\mathcal{L}[i]}|))$ time in the worst case. This can be reduced further to $O(|\mathcal{A}||\mathcal{L}|(|\mathcal{L}| - 1)(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2)$ since $O(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2 \gg O(|V_{\mathcal{L}[i]}|)$ and for real networks, $O(|V_{\mathcal{L}[i]}|) \gg O(|\mathcal{X}_{all}|)$. Then, to obtain *RelativeRank* for all variants of FUSSY, $O(|\mathcal{A}||\mathcal{L}|(|\mathcal{L}| - 1)\log(|\mathcal{L}| - 1))$ time is needed assuming that heapsort [297] is used for ranking. Next, $O(|\mathcal{A}||\mathcal{L}|)$ and $O(|\mathcal{A}|(|\mathcal{L}| - 1)(|\mathcal{L}| - 2))$ time is required to compute the Spearman footrule distance and Kendall-Tau distance, respectively. Finally, in GETBESTFEATURES, $O(|\mathcal{A}|\log(|\mathcal{A}|))$ time is needed to find the best FUSSY variant. Taken together, LEARNBESTFEATURES procedure requires $O(|\mathcal{L}|(|\mathcal{L}| - 1)(O(\mathcal{G}(\mathcal{X}_{all})) + |\mathcal{A}|(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2))$ time since in real networks $(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2 \gg \log(|\mathcal{L}| - 1)$ and $(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2 \gg \log(|\mathcal{A}|)$.

In procedure LEARNPTHRESHOLD, $O(|\mathcal{L}|(|\mathcal{L}| - 1))$ time is needed to examine the $|\mathcal{L}| \times |\mathcal{L}| - 1$ P_{best} matrix. Finally, in procedure GETBESTNETWORK, computation of the combined p -values takes $O((|\mathcal{L}| - 1)(|V||V_{\mathcal{L}[i]}|)^2)$ time, filtering for dissimilar networks takes $O(|\mathcal{L}|)$ time and ranking the similar networks takes $O(|\mathcal{L}|\log(|\mathcal{L}|))$ time in the worst case when all the training networks are similar to the unseen network. Taken together, GETBESTNETWORK procedure requires $O((|\mathcal{L}| - 1)(|V||V_{\mathcal{L}[i]}|)^2)$ time since for real networks, $O((|\mathcal{L}| - 1)(|V||V_{\mathcal{L}[i]}|)^2) \gg O(|\mathcal{L}|\log(|\mathcal{L}|)) \gg O(|\mathcal{L}|)$.

Hence, FUSSY requires $O(|\mathcal{L}|(|\mathcal{L}| - 1)(O(\mathcal{G}(\mathcal{X})) + |\mathcal{A}|(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2))$ time. Note that the learning phase in FUSSY can be performed offline and this reduces the worst time complexity to $O((|\mathcal{L}| - 1)(|V||V_{\mathcal{L}[i]}|)^2)$.

Theorem 6.4 *Given an unseen network $G = (V, E)$ and a set of reference networks $\mathcal{L}$, the space complexity of FUSSY is $O((|\mathcal{L}| + 1)(\text{MAX}(|E_{\mathcal{L}[i]}|, |E|) + \text{MAX}(|V_{\mathcal{L}[i]}|, |V|)) + |\mathcal{A}||\mathcal{L}|^2)$ space in the worst case, where $|V_{\mathcal{L}[i]}|$ and $|E_{\mathcal{L}[i]}|$ are the number of nodes and edges of the i^{th} network in $\mathcal{L}$, respectively, $\text{MAX}(\cdot)$ is the maximum operator and $|\mathcal{A}|$ is the number of FUSSY variants.*

Proof: First, FUSSY requires $O((|\mathcal{L}| + 1)(\text{MAX}(|E_{\mathcal{L}[i]}|, |E|) + \text{MAX}(|V_{\mathcal{L}[i]}|, |V|)))$ to store the given unseen network and the given set of training networks where $\text{MAX}(\cdot)$ is the maximum operator. In the first procedure of FUSSY (`LEARNBESTFEATURES`), $O(|\mathcal{A}|)$, $O(|\mathcal{L}||\mathcal{X}_{all}|)$ and $O(|\mathcal{L}|(|\mathcal{L}| - 1))$ space is needed to store the various variants, to store the extracted features, and to store the ground truth, respectively. In addition, $O(|\mathcal{A}||\mathcal{L}|(|\mathcal{L}| - 1))$ space is needed for storing *PValue*, *RelativeRank* and *Distance*. Hence, taken together, `LEARNBESTFEATURES` requires $O(|\mathcal{A}||\mathcal{L}|(|\mathcal{L}| - 1) + |\mathcal{L}||\mathcal{X}_{all}|)$ space. In `LEARNPTHRESHOLD` procedure, $O(1)$ is needed to store p_t . Finally, in `GETBESTNETWORK` procedure, $O(|\mathcal{X}_{best}|)$ space is needed to store the extracted features and $O(|\mathcal{L}|)$ space is needed to store the combined p -value and the ranks of the similar training networks. Taken together, `GETBESTNETWORK` requires $O(|\mathcal{X}_{best}| + |\mathcal{L}|)$ space. Hence, FUSSY requires $O((|\mathcal{L}| + 1)(\text{MAX}(|E_{\mathcal{L}[i]}|, |E|) + \text{MAX}(|V_{\mathcal{L}[i]}|, |V|))) + |\mathcal{A}||\mathcal{L}|^2$ space since for real networks $|E_{\mathcal{L}[i]}| + |V_{\mathcal{L}[i]}| \gg |\mathcal{X}_{all}|$. Note that the learning phase in FUSSY can be performed offline and this reduces the worst space complexity to $O((|\mathcal{L}| + 1)(\text{MAX}(|E_{\mathcal{L}[i]}|, |E|) + \text{MAX}(|V_{\mathcal{L}[i]}|, |V|)))$.

6.6.4 The Algorithm TAPESTRY

The TAPESTRY algorithm is outlined in Algorithm 11. Given an unseen network $G = (V, E)$, an output node x , a set of training networks $\mathcal{L}$ and three optional parameters⁷ r , t_w and $step_w$, TAPESTRY obtains a $|V_{candidate}| \times |N|$ matrix of prioritized targets $\mathcal{P}$ where $V_{candidate}$ is the set of candidate targets in G and $|N|$ is the number of ranked list of $V_{candidate}$. As we shall see later, $|N|$ depends on the weight learning process in `PRIORITIZETARGETS`. We shall now elaborate on the steps in the TAPESTRY algorithm.

First, in the *candidate filtering* phase (Line 2), potential candidate nodes are found using Algorithm 2 in Chapter 5. Then, in the *best matched network identification* phase (Line 3), the

⁷The optional parameter r is subsequently used for learning by the FUSSY algorithm while t_w and $step_w$ are for learning by the `PRIORITIZETARGETS` procedure.

Algorithm 11: Algorithm TAPESTRY

Input: Unseen network G , output node x , set of training networks $\mathcal{L}$, relaxation parameter r (optional), weights learning parameter t_w (optional) and weight range step-size $step_w$ (optional).

Output: $|V_{candidate}| \times |N|$ matrix of TAPESTRY-prioritized node $\mathcal{P}$.

- 1 $\mathcal{P} \leftarrow \text{INITIALIZE}(\mathcal{P})$
- 2 $V_{candidate} \leftarrow \text{FILTERCANDIDATE}(G, x)$
- 3 $G_{best} \leftarrow \text{FINDBESTMATCHEDNETWORK}(G, \mathcal{L}, r)$
- 4 $\mathcal{P} \leftarrow \text{PRIORITIZETARGETS}(G, \mathcal{L}, G_{best}, V_{candidate}, x, t_w, step_w)$

best matched network $G_{best} \in \mathcal{L}$ of G is identified using an appropriate network similarity ranking approach such as FUSSY (Algorithm 8). Finally, in the *target prioritization* phase (Line 4), SFB ranks $\mathcal{S}$ of the unseen network are generated using G_{best} and DFB ranks $\mathcal{D}$ are obtained according to Definition 6.17. These ranks are combined using a weighted sum approach to produce the putative target score that is used for target prioritization.

PRIORITIZETARGETS Procedure. We shall now describe Algorithm 12 which consists of the *weight learning* phase and the *target ranking* phase. The *weights learning* phase is made up of three subphases: *initialization* (Phase 1.1), *preliminary rank computation* (Phase 1.2) and *weight exploration* (Phase 1.3). Likewise, the *target ranking* phase consists of three subphases, namely, *preliminary rank computation* (Phase 2.1), *weight retrieval* (Phase 2.2) and *target prioritization* (Phase 2.3).

First, the procedure initializes t_w and $step_w$ to the default values⁸ of 0.8 and 0.1, respectively if they are not provided in the inputs (Line 1). Then, in Phase 1.1, it obtains the output node and candidate node set of every training network $\mathcal{L}_i$ (Lines 3 to 4). Next, in Phase 1.2 (Lines 5 to 6), the structural and dynamics based ranks are obtained using the GETSTRUCTURALRANK (Algorithm 13) and GETDYNAMICSRANK (Algorithm 14) procedures, respectively. In Phase 1.3, the procedure explores weights in the range (0–1). The putative target score is calculated

⁸Recall that ROC AUC in the range of [0.7, 0.8] and [0.8, 0.9] indicate acceptable and excellent performances, respectively [153]. We select $t_w = 0.8$ as the default to ensure excellent performance of the prioritization model.

Algorithm 12: Procedure PRIORITIZETARGETS

Input: Unseen network G , set of training networks $\mathcal{L}$, best matched network G_{best} , set of candidate targets $V_{candidate}$, output node x , weights learning parameter t_w (optional) and weight range step-size $step_w$ (optional).

Output: $|V_{candidate}| \times |N|$ matrix of TAPESTRY-prioritized node $\mathcal{P}$.

```

1  $t_w, step_w \leftarrow \text{INITIALIZE\_THRESHOLD}(t_w, step_w)$ 
2 foreach iteration  $i=1$  to  $|\mathcal{L}|$  do
3    $x_{\mathcal{L}[i]} \leftarrow \text{GETOUTPUTNODE}(\mathcal{L}_i)$ 
4    $Y_{\mathcal{L}[i]} \leftarrow \text{FILTERCANDIDATE}(\mathcal{L}_i, x_{\mathcal{L}[i]})$ 
5    $\mathcal{S}_{\mathcal{L}[\cdot]} \leftarrow \text{GETSTRUCTURALRANK}(\mathcal{L}_i, \mathcal{L}_i, Y_{\mathcal{L}[i]})$ 
6    $\mathcal{D}_{\mathcal{L}[\cdot]} \leftarrow \text{GETDYNAMICSRANK}(\mathcal{L}_i, x_{\mathcal{L}[i]}, Y_{\mathcal{L}[i]})$ 
7   foreach iteration  $j=1$  to  $\frac{10}{step_w}$  do
8      $\rho_{\mathcal{L}[i]} \leftarrow \text{GETPUTATIVETARGETSCORE}(\mathcal{L}_i, \mathcal{D}_{\mathcal{L}[\cdot]}, \mathcal{S}_{\mathcal{L}[\cdot]}, j \times step_w, 1 - j \times step_w)$ 
9      $\mathcal{P}_{\mathcal{L}[i]} \leftarrow \text{RANK}(\rho_{\mathcal{L}[i]})$ 
10     $\text{ROC}_{\mathcal{L}[i]} \leftarrow \text{ROCANALYSIS}(\mathcal{P}_{\mathcal{L}[i]}, \text{GETKNOWNTARGETS}(\mathcal{L}_i))$ 
11    if  $\text{ROC}_{\mathcal{L}[i]} \geq t_w$  then
12       $W_{\mathcal{L}[i]} \leftarrow \text{APPENDTOWEIGHTLIST}(j \times step_w)$ 
13  $\mathcal{S} \leftarrow \text{GETSTRUCTURALRANK}(G, G_{best}, V_{candidate})$ 
14  $\mathcal{D} \leftarrow \text{GETDYNAMICSRANK}(G, x, V_{candidate})$ 
15  $WeightList \leftarrow \text{GETWEIGHTLIST}(W, G_{best})$ 
16 foreach iteration  $i=1$  to  $|WeightList|$  do
17    $\rho_i \leftarrow \text{GETPUTATIVETARGETSCORE}(G, \mathcal{D}, \mathcal{S}, WeightList_i, 1 - WeightList_i)$ 
18    $\mathcal{P}_i \leftarrow \text{RANK}(\rho_i)$ 

```

using Definition 6.15 and used for ranking the targets in the training network (Lines 8 to 9). Then, it performs ROC analysis on the rankings and those weight resulting in ROC AUC greater than or equals to t_w are stored in the weight array W (Lines 10 to 12).

In the *target ranking* phase, the structural and dynamics based ranks are first obtained using the GETSTRUCTURALRANK and GETDYNAMICSRANK procedures, respectively (Lines 13 to 14). Then, in Phase 2.2, the relevant list of weight *WeightList* for the best matched network is retrieved from W (Line 15). Finally, for each weight in *WeightList*, the putative target score is computed and used for producing a TAPESTRY-prioritized rank list $\mathcal{P}_i$ (Lines 17 to 18).

GETSTRUCTURALRANK Procedure. The computation of the SFB ranks outlined in Algorithm 13 consists of three steps. First, the relevant prioritization model and predictive struc-

Algorithm 13: Procedure GETSTRUCTURALRANK

Input: Unseen network G , best matched network G_{best} , and candidate target node set $V_{candidate}$.

Output: Structural feature-prioritized node set $\mathcal{S}$

```

1  $\mathcal{X}_{pred} \leftarrow \text{GETPREDICTIVEFEATURESET}(G_{best})$ 
2  $Q_{pred} \leftarrow \text{GETPREDICTIVEMODEL}(G_{best})$ 
3 foreach iteration  $i=1$  to  $|\mathcal{X}_{pred}|$  do
4    $H^i \leftarrow \text{GETFEATURES}(G, V_{candidate})$ 
5  $\mathcal{S} \leftarrow \text{RANK}(\text{PERFORMSVR}(Q_{pred}, H, V_{candidate}))$ 

```

tural feature set of the best matched network are retrieved from the *Target Characterization Knowledge Base* (Lines 1 and 2), respectively. Then, the predictive structural feature sets of G are extracted (Line 4) and used as inputs to the prioritization model Q_{pred} to obtain the set of SFB rank $\mathcal{S}$ (Line 5).

GETDYNAMICSRANK Procedure. Algorithm 14 outlines the computation of the PSSD. The concentration-time profiles used to compute the PSSD may be obtained from experiments or *in silico* simulations of biological models. In this thesis, we use the latter approach. For instance, the MAPK-PI3K ODE model [143] was simulated in *Copasi* [310] using parameters: $\{duration=1800 \text{ seconds}, intervals=6 \text{ seconds}\}$ ⁹ to obtain the time-series profiles. Each concentration time-series profile ϕ_i is first Z-normalized using the ZNORMALIZE function (Line 3). Then, the inverted profile ϕ'_u is obtained according to Section 6.6.1 (Line 4). Following which, the DTW distances (Line 5) between the two pairs of profiles (ϕ_i, ϕ_x) and (ϕ'_i, ϕ'_x) are computed using the FASTDTW algorithm in [311]. The lesser of the two distances is selected as Φ_x^i which is used to rank the candidate nodes $V_{candidate}$ in decreasing order (Line 6).

⁹The actual CPU time required for the simulation is about 1 second using a 32-bit operating system with 2GB RAM and a dual core processor at 1.86GHz. The simulation time is unrelated to the duration parameter which intuitively, corresponds to the range of ϕ and is related to $|\phi| (\frac{duration}{interval} = |\phi|)$.

Algorithm 14: Procedure GETDYNAMICSRANK

Input: Unseen network G , output node x , and candidate target node set $V_{candidate}$.

Output: Dynamic feature-prioritized-prioritized node set $\mathcal{D}$

```

1  $\phi_x \leftarrow \text{ZNORMALIZE}(G, x)$ 
2 foreach iteration  $i=1$  to  $|V_{candidate}|$  do
3    $\phi_i \leftarrow \text{ZNORMALIZE}(G, V_{candidate}[i])$ 
4    $\phi'_i \leftarrow \text{COMPUTEINVERTEDPROFILE}(\phi_i)$ 
5    $\Phi_x^i \leftarrow \text{MIN}(\text{DTW}(\phi_i, \phi_x), \text{DTW}(\phi'_i, \phi_x))$ 
6  $\mathcal{D} \leftarrow \text{RANK}(\Phi, V_{candidate})$ 

```

6.6.5 An Example

Given an unseen network, Ras activation network, with output node `RasGTP_PM` and a set of training networks consisting of `MAPK_PI3K` (I_1), glucose-stimulated insulin secretion (I_2), endomesoderm gene regulatory (I_3) and glucose metabolism (I_4) networks, TAPESTRY first identifies the set of candidate targets $V_{candidate}$ using the `FILTERCANDIDATE` procedure. Note that all nodes are considered as candidate targets since they all contain at least one path to `RasGTP_PM`.

Then, we apply a network similarity ranking approach to identify the best matched training network using the `FINDBESTMATCHEDNETWORK` procedure. Assuming `FUSSY` is the network similarity ranking approach used, the `LEARNBESTFEATURES` procedure in Algorithm 8 performs learning on the set of training networks as described in Section 6.6.3 to identify the best `FUSSY` variant for performing network similarity ranking. The algorithm iterates through each training network. For each iteration, it prioritizes the targets in the training network using the prioritization model of the remaining networks and then perform ROC analysis to establish the ground truth for each training network. Then, it ranks the remaining networks using all `FUSSY` variants and uses the sum of the Spearman footrule distance and the Kendall-Tau distance to determine which `FUSSY` variant yields a ranking closest to the ground truth. `FUSSY(D,F)` which uses only dynamic features and `FISHER(·)` for combining p -values was found to be the best variant. Next, the `LEARNPTHRESHOLD` procedure identifies p -values

corresponding to “poor” ROC AUC (*i.e.*, ROC AUC < 0.5). The average of these p -values (*e.g.*, 0.01) is set as the threshold p_t for filtering dissimilar networks.

Following which, the best matched network of the Ras activation network is identified using the GETBESTNETWORK procedure. Specifically, the dynamic feature of the Ras activation network and the training networks are extracted. Then, Wilcoxon and KS tests are performed to determine the similarity of the dynamic feature distribution between the Ras activation network and each training network. The p -values obtained are combined using FISHER($\cdot$) and the training networks with combined p -values above p_t are ranked in decreasing order of the combined p -values. In particular, the ranking of the training networks is $I_4 \prec I_1 \prec I_2$ and I_3 is found to be a dissimilar network. Hence, I_4 is the best matched network.

Next, the PRIORITIZETARGETS procedure performs weight learning on training networks to identify appropriate setting of weights for SFB and DFB ranks to be used for generating TAPESTRY-prioritized ranks for each training network. An appropriate weight is one that yields a TAPESTRY-prioritized rank with ROC AUC¹⁰ greater than or equal to 0.8. Note that the appropriate weight (*e.g.*, $\varpi_1 = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$) of the best matched network I_4 will be used subsequently for prioritizing targets in the unseen network. Following which, the prioritization model of I_4 and its associated set of predictive structural features (*e.g.*, closeness centrality, eigenvector centrality and bridging centrality) are retrieved using the GETSTRUCTUREBASEDRANK procedure. These centrality values are extracted from the Ras activation network and then input to the prioritization model to obtain SFB ranks using support vector regression (see Section 5.3.2). The GETDYNAMICSBASEDRANK procedure then generates the PSSD ranks using the concentration-time series of nodes in the Ras activation network. The concentration-time series are first Z-normalized and then vertically inverted. The PSSD of each node u is computed as the minimum of two dynamic time distances, namely $\text{DTW}(\phi_u, \phi_{\text{RASGTP_PM}})$ and

¹⁰Since we did not provide t_w in the inputs, it is set to 0.8 by default (Section 6.6.4).

$\text{DTW}(\phi'_u, \phi'_{\text{RASGTP_PM}})$, where ϕ_u and ϕ'_u denote the concentration-time series and vertically-inverted concentration-time series of u , respectively and $\text{DTW}(\cdot)$ is the dynamic time warping distance function. DFB ranks are obtained from ranking the nodes in increasing PSSD.

Finally, SFB and DFB ranks are combined based on Definition 6.15 using the weights ($\varpi_1 = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$) relevant to I_4 to generate TAPESTRY-prioritized ranks. Note that 6 sets of TAPESTRY-prioritized ranks, each corresponding to one ϖ_1 value, are found for the Ras activation network. An example of using the ranked lists for guiding wet lab experiments is to prioritize experiments involving top-ranking targets in these 6 ranked lists over other targets.

6.6.6 Algorithm Analysis

Theorem 6.5 *Given an unseen network G and a set of training networks $\mathcal{L}$, let $O(\mathcal{H}(\cdot))$ and $O(\mathcal{G}(\cdot))$ be the worst time complexity of the given network similarity ranking approach and the feature extraction function. Then, Algorithm TAPESTRY takes $O((|\mathcal{L}| + 1)(|V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|)^2 + O(\mathcal{G}(X_{\text{all}})) + |\phi_{\mathcal{L}[i]}||V_{\mathcal{L}[i]}| + O(\mathcal{H}(G, \mathcal{L})))$ time in the worst case, where X_{all} is the set of features, $|V_{\mathcal{L}[i]}|$ and $|E_{\mathcal{L}[i]}|$ are number of nodes and edges of $\mathcal{L}_i$ (the i^{th} network in $\mathcal{L}$), respectively, and $|\phi_{\mathcal{L}[i]}|$ is the number of time points in the time series of $\mathcal{L}_i$.*

Proof: Following from Section 5.5.3, the FILTERCANDIDATE algorithm takes $O(|V_{bi}|^2 + |E_{bi}|)$ time where $G_{bi} = (V_{bi}, E_{bi})$ is the bipartite representation of the given signaling network $G = (V, E)$, $|V_{bi}| = |V| + |E|$ and $|E_{bi}| = \sum_{(U,W) \in E} (|U| + |W|)$.

The time complexity of the FINDEQUIVALENTNETWORK procedure (denoted as $O(\mathcal{H}(\cdot))$) depends on the algorithm that is used for network similarity ranking. The GDD approach has complexity of $O(|V|^5)$ since the graphlet size is limited to 5 nodes [239]. The NETSIMILE approach has complexity of $O(\sum_{j=1}^k (|\mathcal{X}||V_{\mathcal{F}[j]}| + |\mathcal{X}||V_{\mathcal{F}[j]}|\log(|V_{\mathcal{F}[j]}|)))$ [24] where $|\mathcal{X}|$ is the number of structural features, and $|V_{\mathcal{F}[j]}|$ is the number of nodes in the j^{th} network in $\mathcal{F}$. According to Berlingerio et al. [24], $|\mathcal{X}| \ll |V_{\mathcal{F}[j]}| \ll |E_{\mathcal{F}[j]}|$ and $|V_{\mathcal{F}[j]}|\log(|V_{\mathcal{F}[j]}|) \approx |E_{\mathcal{F}[j]}|$ in

real-world networks where $|E_{\mathcal{F}[j]}|$ is the number of edges in the j^{th} network in $\mathcal{F}$. Recall that FUSSY has complexity of $O(|\mathcal{L}|(|\mathcal{L}| - 1)(O(\mathcal{G}(\mathcal{X}_{all})) + |\mathcal{A}|(|V_{\mathcal{L}[i]}||V_{\mathcal{L}[j]}|)^2))$ time in the worst case, where $\mathcal{X}_{all}$ is the set of network features, $|V_{\mathcal{L}[i]}|$ is the number of nodes of the i^{th} network in $\mathcal{L}$ and $|\mathcal{A}|$ is the number of FUSSY variants. Hence, the order of the three approaches in terms of increasing time complexity is $\text{NETSIMILE} \succ \text{FUSSY} \succ \text{GDD}$.

In Algorithm 12, `GETOUTPUTNODE`; `FILTERCANDIDATE`; `GETPUTATIVETARGETSCORE`; `RANK`; `ROCANALYSIS`; `APPENDTOWEIGHTLIST` and `GETWEIGHTLIST` procedures require $O(1)$; $O(|V_{bi,\mathcal{L}[i]}|^2 + |E_{bi,\mathcal{L}[i]}|)$; $O(|V_{\mathcal{L}[i]}|)$; $O(|V_{\mathcal{L}[i]}|\log(|V_{\mathcal{L}[i]}|))$; $O(|V_{\mathcal{L}[i]}|^2)$ [110]; $O(1)$ and $O(1)$ time, respectively, where $|V_{bi,\mathcal{L}[i]}|$ and $|E_{bi,\mathcal{L}[i]}|$ are the number of nodes and edges of the bipartite graph of $\mathcal{L}_i$ (the i^{th} network in $\mathcal{L}$) and $|V_{\mathcal{L}[i]}|$ is the number of nodes in the $\mathcal{L}_i$. In `GETSTRUCTUREBASEDRANK` procedure, feature extraction takes $O(\mathcal{G}(\mathcal{X}_{all}))$; generation of the SFB scores takes $O(|\mathbf{v}|)$ [25] where $\mathbf{v}$ is the number of support vectors (in the worst case, $|\mathbf{v}| = |V_{\mathcal{L}[i]}|$); and ranking using heapsort takes $O(|V_{\mathcal{L}[i]}|\log(|V_{\mathcal{L}[i]}|))$ time [297]. Hence, the time complexity of `GETSTRUCTUREBASEDRANK` is $O(O(\mathcal{G}(\mathcal{X}_{all})) + |V_{\mathcal{L}[i]}|\log(|V_{\mathcal{L}[i]}|))$. In `GETDYNAMICSBASEDRANK` procedure, in the worst case, $O(|\phi_{\mathcal{L}[i]}||V_{\mathcal{L}[i]}|)$ time is needed for Z-normalization, for performing profile inversion, and for DTW computation using `FASTDTW` [311], where $|\phi_{\mathcal{L}[i]}|$ is the number of points in the time-series profile of $\mathcal{L}_i$. The heapsort used for ranking in the `RANK` procedure takes $O(|V_{\mathcal{L}[i]}|\log(|V_{\mathcal{L}[i]}|))$ time. Hence, the time complexity of `GETSTRUCTUREBASEDRANK` is $O(|\phi_{\mathcal{L}[i]}||V_{\mathcal{L}[i]}| + |V_{\mathcal{L}[i]}|\log(|V_{\mathcal{L}[i]}|))$. Taken together, the time complexity of Algorithm 12 is $O((|\mathcal{L}| + 1)(|V_{bi,\mathcal{L}[i]}|^2 + |E_{bi,\mathcal{L}[i]}| + |V_{\mathcal{L}[i]}|\log(|V_{\mathcal{L}[i]}|) + O(\mathcal{G}(\mathcal{X}_{all})) + |\phi_{\mathcal{L}[i]}||V_{\mathcal{L}[i]}|))$. This can be further reduced to $O((|\mathcal{L}| + 1)((|V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|)^2 + O(\mathcal{G}(\mathcal{X}_{all})) + |\phi_{\mathcal{L}[i]}||V_{\mathcal{L}[i]}|))$ since the signaling network is a single strongly connected component in the worst case and $O(|E_{bi,\mathcal{L}[i]}|) = O(|V_{bi,\mathcal{L}[i]}|^2)$ and $|V_{bi,\mathcal{L}[i]}| = |V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|$.

Hence, the `TAPESTRY` algorithm requires $O((|\mathcal{L}| + 1)((|V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|)^2 + O(\mathcal{G}(\mathcal{X}_{all})) + |\phi_{\mathcal{L}[i]}||V_{\mathcal{L}[i]}|) + O(\mathcal{H}(\mathcal{F})))$ time.

Theorem 6.6 *Given an unseen network G and a set of training networks $\mathcal{L}$, let $O(\mathcal{K}(\cdot))$ be the worst space complexity of the given network similarity ranking approach. Then, Algorithm TAPESTRY requires $O(|\mathcal{L}|(|V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|)^2 + O(\mathcal{K}(G \cup \mathcal{L})))$ space in the worst case, where $|V_{\mathcal{L}[i]}|$ and $|E_{\mathcal{L}[i]}|$ are number of nodes and edges of $\mathcal{L}_i$ (the i^{th} network in $\mathcal{L}$), respectively.*

Proof: Following from Section 5.5.3, the FILTERCANDIDATE algorithm requires $O(|V_{bi}| + |E_{bi}| + |V_{DAG}|^2)$ space in the worst case, where $G_{bi} = (V_{bi}, E_{bi})$ is the bipartite representation of the given signaling network $G = (V, E)$; $|V_{bi}| = |V| + |E|$; $|E_{bi}| = \sum_{(U,W) \in E} (|U| + |W|)$ and $G_{DAG} = (V_{DAG}, E_{DAG})$ is the corresponding directed acyclic graph (DAG) of G . Hence, the complexity can be further reduced to $O((|V| + |E|)^2)$.

The space required for FINDEQUIVALENTNETWORK procedure (denoted as $O(\mathcal{K}(\cdot))$) depends on the algorithm used. In GDD, $O(\sum_{i=1}^{|\mathcal{F}|} (|V_{\mathcal{F}[i]}| + |E_{\mathcal{F}[i]}| + |\Upsilon|))$ space is needed for storing the graph information and automorphism orbits details where $|\Upsilon|$ is the number of automorphism orbits being considered. NETSIMILE has space complexity of $O(\sum_{i=1}^{|\mathcal{F}|} |\mathcal{X}_{all}| |V_{\mathcal{F}[i]}|)$ while FUSSY has space complexity of $O(|\mathcal{F}|(|E_{\mathcal{F}[i]}| + |V_{\mathcal{F}[i]}|) + |\mathcal{A}||\mathcal{L}|^2)$ (Theorem 6.6.3).

In Algorithm 12, GETOUTPUTNODE; FILTERCANDIDATE; GETPUTATIVETARGETSCORE; RANK; ROCANALYSIS; APPENDTOWEIGHTLIST and GETWEIGHTLIST procedures require $O(1)$; $O(|V_{bi,\mathcal{L}[i]}| + |E_{bi,\mathcal{L}[i]}| + |V_{DAG,\mathcal{L}[i]}|^2)$; $O(|V_{\mathcal{L}[i]}|)$; $O(|V_{\mathcal{L}[i]}|)$; $O(|V_{\mathcal{L}[i]}|)$; $O(|M|)$ and $O(|M|)$ space, respectively, where $|V_{bi,\mathcal{L}[i]}|$ and $|E_{bi,\mathcal{L}[i]}|$ are the number of nodes and edges of the bipartite graph of $\mathcal{L}_i$ (the i^{th} network in $\mathcal{L}$); $|V_{\mathcal{L}[i]}|$ is the number of nodes in the $\mathcal{L}_i$; $|V_{DAG,\mathcal{L}[i]}|$ is the number of nodes in the DAG graph of $\mathcal{L}_i$ and $|M|$ is the number of weight ratio considered. The GETSTRUCTUREBASEDRANK procedure requires $O(|V_{\mathcal{L}[i]}||\mathcal{X}_{all}| + |E_{\mathcal{L}[i]}|)$ space for storing $\mathcal{L}_i$ and its network feature values. In GETDYNAMICSBASEDRANK procedure, suppose only two time-series profiles (ϕ_v and ϕ_x) are stored at any time, then $O(|\phi_{\mathcal{L}[i]}| + |V_{\mathcal{L}[i]}|)$ space is needed. For real networks, $|M| \leq |V_{\mathcal{L}[i]}|$, $|V_{\mathcal{L}[i]}| \geq |\phi_{\mathcal{L}[i]}|$ and $|V_{\mathcal{L}[i]}| \geq |V_{\mathcal{L}[i]}||\mathcal{X}_{all}|$. Taken together, Algorithm 12 requires $O(|\mathcal{L}|(|V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|)^2)$ space in the worst case since $|V_{bi,\mathcal{L}[i]}| = |V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|$ and $|E_{bi,\mathcal{L}[i]}| = |V_{bi,\mathcal{L}[i]}|^2$.

Hence, the TAPESTRY algorithm requires $O(|\mathcal{L}|(|V_{\mathcal{L}[i]}| + |E_{\mathcal{L}[i]}|)^2 + O(\mathcal{K}(G \cup \mathcal{L})))$ space.

6.7 Experiments

In this section, we describe the experimental settings and discuss the experimental results obtained.

Recall that in Algorithm 11, the `FINDBESTMATCHEDNETWORK` procedure identifies the best matched network and then uses its prioritization model to rank the targets of the given unseen network using the `PRIORITIZETARGETS` procedure. The experiments are designed to evaluate these two procedures and to answer the following questions.

- (i) Q1: Are the prioritization models of the training network appropriate for prioritizing targets in the unseen network?
- (ii) Q2: What is the best matched network of the Ras activation network?
- (iii) Q3: Which FUSSY variant best identifies the best matched network?
- (iv) Q4: How does FUSSY compare with GDD [287] and NETSIMILE [24]?
- (v) Q5: Is DTW better than Euclidean distance for PSSD?
- (vi) Q6: What is the effects of varying the weights of SFB ranks and DFB ranks on target prioritization?
- (vii) Q7: How does TAPESTRY compare with random prioritization and other state-of-the-art prioritization approaches?

In particular, question Q1 justifies the use of prioritization models of training network for target prioritization in TAPESTRY whereas question Q2 helps to establish the ground truth for the Ras activation network and provide a basis for comparing different network similarity ranking approaches. Question Q5, on the other hand, is for justifying the use of the DTW measure in PSSD.

6.7.1 Experimental Setting

TAPESTRY is implemented using Java and the Wilcoxon and KS tests are performed using *R*. In addition, we use several publicly available tools and libraries that are listed below:

- (i) *FastDTW* library [311]: for performing DTW calculation on the time series profiles (DTW in Algorithm 14).
- (ii) *Jung* library [267]: for creating the graph model.
- (iii) *libSBML* library [37]: for reading and processing the SBML files of the networks.
- (iv) *libSVM* library [51]: for performing the regression learning to obtain the SFB ranks $\mathcal{S}$ (PERFORMSVR in Algorithm 13).
- (v) *GraphletCounter* [380] plugin for *Cytoscape* [320]: for computing GDD.
- (vi) *Copasi* [152]: for generating the concentration-time series profiles and performing LSA.
- (vii) *NetworkPrioritizer* [171] plugin for *Cytoscape* [320]: for comparison with TAPESTRY.

The experiments are performed on a computer system using a 64-bit operating system with 8GB RAM and a dual core processor running at 3.60GHz.

Data Set and Variants of Network Similarity Ranking Methods. In this chapter, our data set includes the training networks in Chapter 5 and the Ras activation network (unseen network). The description and notations of the training networks follow that in Chapter 5. Recall that in Section 6.6.4 we study different variants of the network similarity ranking approaches. These approaches are summarized in Table 6.3.

Performance Metrics. Two types of rankings are being performed in TAPESTRY, ranking of networks in `FINDBESTMATCHEDNETWORK` and ranking of target nodes in `PRIORITIZETARGETS`.

In network ranking, it is possible to establish the true order of the similarity ranking (ground truth) of the different training networks with respect to the Ras activation network (Section 6.7.2). Given an unseen network G and two training networks L_1 and L_2 , L_1 has a more “appropriate” prioritization model for G if the prioritization model of L_1 produces prioritization of targets in G with larger ROC AUC than that produced by the prioritization model of L_2 . Hence, by ordering the training network based on decreasing ROC AUC, we can obtain the ground truth. The different network similarity ranking approaches in Table 6.3 can then be compared based on this ground truth using a distance value derived from the sum of the Spearman footrule distance [91] and the Kendall-Tau distance [98]. The best approach is the one having minimal disagreement (*i.e.*, smallest distance) with the ground truth.

In target ranking, a ground truth similar to that of the network ranking is not available. However, rank comparison can still be performed using a set of known targets which can be found using the curation approach described in Section 6.4. In particular, we use ROC analysis to determine how well a list of target ranks P_1 identifies the set of known targets as compared to another list P_2 . Note that P_1 is a better prioritization of the given unseen network G if its ROC AUC is larger than that of P_2 . Note that the ROC AUC metric can also be replaced with the AUPR (area under precision-recall curve) metric to obtain an alternative view of the ground truth. AUPR has been suggested as a better measure in situations where the target class contain few true instances since it penalizes false positive among the best ranked prediction in a greater extend [83].

6.7.2 Experimental Results

Ground Truth. Recall that the goal of identifying the best matched network is to use its prioritization model to prioritize targets of the Ras activation network. Before we proceed

Figure 6.4: ROC AUC analysis of target ranking on the Ras activation network using the prioritization model of different training networks.

further, it is important to first establish if prioritization model of the training networks can indeed prioritize targets in the Ras activation network. It is also necessary to determine the true ranking order of the different training networks to provide a basis for comparing different network similarity ranking approaches.

We apply the prioritization model of each training network to generate the target ranks of the Ras activation network. This results in four prioritized target lists of the Ras activation network, each one corresponding to target ranking based on the prioritization model of each training network. Then, ROC analysis is applied to each of the prioritized target list to determine its ability in prioritizing known targets of the Ras activation network. Figure 6.4 plots the ROC AUC of using prioritization models of different training networks. I_4 (ROC AUC=0.832) produced the best prioritization model.

The Best FUSSY Variant. In this set of experiments, we identify the FUSSY variant that is most effective in identifying the best matched network for the unseen network and analyse the runtime performance of different FUSSY variants.

Figure 6.5 summarizes the ranking using different FUSSY variants while Figure 6.6 reports

Figure 6.5: Summary of the ranks of I₁ to I₄ obtained using various FUSY variants.

Figure 6.6: Distance between the ground truth and the ranks obtained using various FUSY variants.

the distances between the ground truth and the ranks obtained using various FUSY variants. We make the following observations. First, two FUSY variants (FUSY_(D,F) and FUSY_(D,S)) tie in the top spot. Both variants use only dynamic features. Second, three FUSY variants including FUSY_(D,F) correctly identifies the best matched network I₄ (Figure 6.5). For the remaining variants, I₄ was placed second in the ranking. Third, the method for combining the p -values does not seem to affect the network rankings in general. For instance, FUSY_(D,F) and FUSY_(D,S) have the same network rankings, as do FUSY_(S,F) and FUSY_(S,S). Hence, the best FUSY variant (FUSY_(D,F)) found using the training network performs well for the unseen

Figure 6.7: Runtime performance of various FUSSY variants.

network and correctly identifies I_4 as the best matched network. In subsequent experiments, we shall set $\mathcal{X}_{best} = \mathcal{X}_{dynamic}$ and $pFunc = \text{FISHER}(\cdot)$.

Figure 6.7 plots the runtime performance of various FUSSY variants. We make the following observations. First, the learning process of FUSSY takes up the bulk of its execution time. For instance, in the network comparison of I_0 and I_1 , $\text{FUSSY}_{(S,F)}$ required 2.271 minutes and 0.021 minutes when learning (LEARNBESTFEATURES and LEARNPTHRESHOLD procedures) was performed online and offline, respectively. Second, the size of the networks affects the runtime performance of FUSSY variants, particularly if the learning is performed offline. In general, larger networks (*e.g.*, I_3) require longer runtime compared to smaller networks (*e.g.*, I_0). Third, the types and number of features affect the runtime performance. For instance, in the network comparison of I_0 and I_2 , $\text{FUSSY}_{(D,F)}$, which uses only the dynamic feature (PSSD), is about an order of magnitude slower than $\text{FUSSY}_{(S,F)}$, which uses 16 structural features. Hence, the runtime performance of FUSSY is dependent on the size of the network, the types and number of features used by the variant. In particular, learning can be performed offline to improve the runtime performance of FUSSY. The improvement is up to 2 orders of magnitude in certain FUSSY variants (*e.g.*, $\text{FUSSY}_{(S,F)}$ for comparison of networks with fewer than 100 nodes).

Threshold p -value. In this set of experiments, we identify the threshold p -value p_t learnt from the training networks. Table 6.4 shows the combined p -values of each pair of training

	I₁ Model	I₂ Model	I₃ Model	I₄ Model
I₁	-	6.52×10^{-4} [0.554]	2.2×10^{-16} [0.473 [†]]	0.128 [0.642]
I₂	6.52×10^{-4} [0.619]	-	2.2×10^{-16} [0.544]	0.045 [0.425 [†]]
I₃	2.2×10^{-16} [0.654]	2.2×10^{-16} [0.596]	-	2.2×10^{-16} [0.479 [†]]
I₄	0.128 [0.606]	0.045 [0.514]	2.2×10^{-16} [0.55]	-

Table 6.4: Summary of the LEARNPTRESHOLD procedure. The $(i,j)^{th}$ cell entry is of the form $x[y]$ where x indicates the combined p -value for the (I_i, I_j) pair and y indicates the ROC AUC when prioritization model of I_j is applied to I_i . [†] mark entries with ROC AUC < 0.5.

networks using $\text{FUSSY}_{(D,F)}$ and the corresponding ROC AUC when applying the prioritization model of one training network to another. p_t is the average combined p -value of the cells marked with [†]. That is, $p_t = \frac{2.2 \times 10^{-16} + 2.2 \times 10^{-16} + 0.045}{3} = 0.01$. When we apply $p_t = 0.01$ to the ranked list of I_1 to I_4 obtained using $\text{FUSSY}_{(D,F)}$, we note that I_3 (ROC AUC=0.632, combined p -value= 2.2×10^{-16}) is considered as a dissimilar network and filtered off. Note that p_t affects the number of networks being ranked, but not the actual rank of the network. Hence, it is possible to have no best matched network if the given set of training networks is considered dissimilar to the unseen network.

Performance of Network Similarity Ranking Approaches. In this set of experiments, we compare the performance of the best FUSSY variant against GDD [287] and NETSIMILE [24] in terms of the quality of the network ranking and runtime performance. As observed in Figures 6.8 and 6.9, the best FUSSY variant outperforms both state-of-the-art approaches in terms of quality of ranking. In particular, $\text{FUSSY}_{(D,F)}$ is the only approach that correctly identifies I_4 as the best matched network. Note that both GDD and NETSIMILE consider only topological features and this could be the reason for the poor performance. The finding highlights the importance of using the dynamic feature for identifying the best matched network. It is interesting to note that although no dynamic feature was used for generating the prioritization model¹¹ of the best matched network, it is crucial for selecting the best matched network. This

¹¹The prioritization model obtained from TENET uses only topological features.

Figure 6.8: Summary of the ranks of I_1 to I_4 obtained using various approaches.

Figure 6.9: Distance between the ground truth and the ranks obtained using various approaches.

underlies a relationship between structural and dynamic features. It is widely acknowledged that a complex interplay exists between the structural and dynamic characteristics of a network but the exact relationship is still not clear [22].

Figure 6.10 illustrates the runtime performance of the various approaches. We make the following observations. First, the size of the networks affect all approaches. In particular, the most significant impact is experienced by the GDD-based approaches. Specifically, the runtime performance of GDD deteriorate by about 3 orders of magnitude when applied to I_3 (622 nodes) as compared to I_4 (47 nodes). Second, amongst the approaches, NETSIMILE performs the best and scales well even for larger networks. In particular, the order of the approaches in

Figure 6.10: Runtime performance of various approaches.

terms of runtime performance is $\text{NETSIMILE} \prec \text{FUSSY}_{(D,F)} \text{ (offline learning)} \prec \text{FUSSY}_{(D,F)} \prec \text{GDD}$. Hence, FUSSY has moderate runtime performance and outperforms other state-of-the-art approaches in terms of prioritization results.

Justification of Using DTW Distance over Euclidean Distance for PSSD. In this experiment we analyze the benefit of deploying a non-linear (DTW distance) measure over a linear distance (*e.g.*, Euclidean) measure for computing PSSD on the `MAPK-PI3K` network. Table 6.5 reports the distance between the time-series profiles of `ERKPP` and nodes in the `MAPK-PI3K` network based on DTW and Euclidean distances. The distances have low correlation (Spearman's coefficient=0.554, p -value=0.00045). We visually compare the profile of `ERKPP` with profiles of nodes that are ranked significantly different by DTW and Euclidean distances. A node is considered to have significantly different ranking if it has a rank difference of $\lfloor \frac{|V_{\text{candidate}}|}{2} \rfloor$ or more since this ranking difference is large enough to result in a node being assigned a high rank in DTW distance and low rank in Euclidean distance, and vice versa. We use the inverse of the distance for ranking purpose as the ranking function assigns higher rank to larger values and larger distances in this case imply greater dissimilarity. In Table 6.5, `Akt` and `PI` have

Node	$D_{DTW}(\cdot)$	$D_{Eucl.}(\cdot)$	Node	$D_{DTW}(\cdot)$	$D_{Eucl.}(\cdot)$	Node	$D_{DTW}(\cdot)$	$D_{Eucl.}(\cdot)$
ERKPP	1	1	PIP3	13	5	PP2A	23	15
RP	2	10	ERKP	14	17	RShGS	24	29
RHRG2	3	8	MEK	15	18	RShP	25	31
RPI3K*	4	9	MEKP	16	14	GS	26	19
PI	5	33	ERK	17	2	RShc	27	30
Akt	6	34	MEKPP	18	16	ShGS	28	27
AktPIP	7	3	AktPIP3	19	12	RasGDP	29	22
AktPIPP	8	13	internalization	20	32	RasGTP	30	25
RHRG	9	7	Raf*	21	28	Shc	31	26
PI3K	10	11	Raf	22	20	HRG	32	24
RPI3K	11	6	E	23	15	R	32	23
PI3K*	12	4	MKP3	23	15	ShP	33	21

Table 6.5: Ranking based on inverse of the DTW and Euclidean distances with respect to ERKPP, denoted as $D_{DTW}(\cdot)$ and $D_{Eucl.}(\cdot)$, respectively. Table is read from top to bottom and from left to right.

large ranking differences. Euclidean distance assigned higher rank to MKP3 even though Akt and PI looked visually more similar to ERKPP (Figure 6.3). Hence, DTW distance is a more suitable measure for comparing PSSD for signaling networks since these networks often have node profiles which exhibit non-linear relationship [1].

Effect of Varying Weights in Training Networks. In this set of experiments, we vary the weights of PSSD ranks and SFB ranks (denoted as ϖ_1 and ϖ_2) to investigate the effect on final prioritization ranks in networks I₁ to I₄. In particular, ϖ_1 and ϖ_2 are set according to Table 6.6 where ϖ_1 and ϖ_2 corresponds to W_1 and W_2 , respectively. The results are shown in Figure 6.11 (left) and a decreasing trend is observed in all four networks. Table 6.7 lists the values of ϖ_1 that yield prioritization results with ROC AUC greater than or equals to 0.8. Except for I₂, the rest of the networks had the best ROC AUC when $\varpi_1 = 0.1$ ¹². These observations imply that the structural features play a more important role than PSSD in terms of the prioritization results.

TAPESTRY-prioritization of the Ras Activation Network. In this set of experiments, we

¹²The best ROC AUC for I₂ occurs when $\varpi_1 = 0$. Note that in I₁ and I₄, the best ROC AUC occurs at $\varpi_1 = 0$ and $\varpi_1 = 0.1$. In I₃, the best ROC AUC at $\varpi_1 = 0.1$ (0.83) is a modest improvement over that at $\varpi_1 = 0$ (0.827).

Weights Ratio ID	W_1	W_2
0	0	1
1	0.1	0.9
2	0.2	0.8
3	0.3	0.7
4	0.4	0.6
5	0.5	0.5
6	0.6	0.4
7	0.7	0.3
8	0.8	0.2
9	0.9	0.1
10	1	0

Table 6.6: Weights ratio used in the experiments.

Training Network	ϖ_1
I_1	0, 0.1, 0.2, 0.3, 0.4
I_2	0, 0.1, 0.2, 0.3
I_3	0, 0.1, 0.2, 0.3
I_4	0, 0.1, 0.2, 0.3, 0.4, 0.5

Table 6.7: Weight ϖ_1 of training networks that yields prioritization results having ROC AUC greater than or equals to 0.8.

apply the weight setting learnt from training networks to prioritize targets in the Ras activation network and investigate if they indeed perform better than other weight setting. The right plot in Figure 6.11 illustrates the ROC AUC of prioritization results of this network using different weight ratio based on the prioritization model of I_4 (best matched network). Note that $\varpi_1 = 0$ implies that target prioritization is performed based on structural features alone (using the prioritization model generated by TENET). We observe that the plot displays a downward trend like those observed in the training networks. In particular, the best ROC AUC is 0.849 using $\varpi_1 = 0.3$. Interestingly, the range of ϖ_1 ([0–0.5]) yielding an ROC AUC greater than or equals to 0.8 correspond to that of I_4 , the best matched network. Hence, although the best weight setting differs for different networks, the weight setting of the best matched network provides a bound on the weight setting to be used on the unseen network.

Comparison of TAPESTRY with Other Target Prioritization Approaches. In this set of

Figure 6.11: Effect of varying the weights of SFB rank and PSSD rank on final prioritization ranks for I_1 to I_4 (left) and the Ras activation network I_0 (right).

experiments, we compare TAPESTRY against other approaches, namely, random prioritization, LSA [139] and *NetworkPrioritizer* [171], in terms of the quality of prioritization and the runtime performance. In random prioritization, we randomly assign a rank to the target nodes of the Ras activation network, assuming that there is no ties in the ranking. LSA was performed using *Copasi* [310] using the following configuration: {task=sensitivities; sub-task=time series; function=all variables of the model; and variable=all parameter values.}. *NetworkPrioritizer* allows prioritization using a variety of features¹³ that are computed using user-configurable rank aggregation approaches. In particular, we use the *Weighted Borda Fuse* (WBF) and *Weighted AddScore Fuse* (WASF) provided by *NetworkPrioritizer* to perform target prioritization. The former aggregation approach is a generalization of the Borda count while the latter approach is based on the weighted sum score of the different feature values. Note that for both approaches, we consider all features in *NetworkPrioritizer* and assign uniform weights to them for rank aggregation since we do not have prior knowledge of which feature or weight is better.

¹³Degree centrality, shortest path betweenness, shortest path closeness, random walk betweenness, random walk transmitter closeness and random walk receiver closeness. Interested reader may refer to [171] for more information.

Figure 6.12: Normalized ranks of targets in I_0 . TAPESTRY₀ and TAPESTRY₃ indicates the TAPESTRY approach with ω_1 set to 0 and 0.3, respectively.

Figure 6.13: Performance of various approaches. TAPESTRY₀ and TAPESTRY₃ indicates the TAPESTRY approach with ω_1 set to 0 and 0.3, respectively.

Figure 6.12 plots the normalized ranks¹⁴ of targets in I_0 using various approaches whereas Figure 6.13 illustrates the performance (AUC ROC and AUPR) of various approaches. TAPESTRY₀ and TAPESTRY₃ have the best AUC ROC and AUPR, respectively among different TAPESTRY variants. In particular, both TAPESTRY₀ and TAPESTRY₃ outperform all the other approaches

¹⁴Recall that in Chapter 5, normalized rank of a node u for a particular approach x is denoted as $\Psi_{norm(x):u}$ and defined as follows.

$$\Psi_{norm(x):u} = \frac{\Psi_{x:u}}{\text{MAX}_{i \in V}(\Psi_x : i)} \quad (6.17)$$

where $\Psi_{x:u}$ is the rank of u based on x , V is the set of nodes in the given signaling network and $\text{MAX}(\cdot)$ is the maximum operator.

Figure 6.14: Runtime performance of various approaches.

in terms of AUC ROC and TAPESTRY₀ is ranked second in terms of AUPR. When we examined the ranks given to known targets (Figure 6.12), TAPESTRY₃ is the only approach that ranked all known targets in the top-50% ranked nodes. Other approaches such as WBF and WASF miss certain targets (EGFR_{PM}, Ca and RasGDP_{PM}) in their top-50% ranked nodes. In addition, from Figure 6.14, we note that TAPESTRY has moderate runtime performance compared to other approaches.

Effect of Using Subset of Predictive Features of Best Matched Network I₄. In this set of experiments, we explore using a subset of the predictive structural features of the best matched network to prioritize the targets of the Ras activation network. Our goal is to find out if the set of predictive structural features can be further optimized leading to a better prioritization model (referred to as new prioritization model) than that of the best matched network. We use the feature selection approach in Chapter 5 for optimizing the predictive structural feature set. Specifically, we use the backward stepwise elimination (BSE) approach as it had been found to perform well in the training networks. Recall that the BSE approach greedily removes a feature in each iteration and stops when there is no further improvement in the prioritization model based on the cross-validation process. Hence, it produces the smallest optimal feature set. In particular, we use Algorithm 1 to perform the optimization and provide the following inputs to

	I₄ Prioritization Model	New Prioritization Model
Features Used for Model	$\zeta, \beta, \kappa_{cyc}, \vartheta, \alpha, \kappa_{in}, \kappa_{mid}, \kappa_{out}, \theta_{out}, \mu, \sigma, \theta_{total}, \kappa_{undir}$	$\theta_{out}, \sigma, \vartheta$
Best Model C Parameter	11.2	6
ROC AUC of SFB Rank	0.832	0.915

Table 6.8: Properties of prioritization models. Refer to Table 4.1 for descriptions of the symbols of the features.

Figure 6.15: Effect of varying the weights of SFB rank and PSSD rank on final prioritization ranks for I₀

the algorithm: Ras activation network as the signaling network G , RasGTP_{PM} as the output node x , the set of curated targets in Table 6.2 as the target set T_x , predictive structural feature set of I₄ as the feature set $\mathcal{X}_{all}$. In addition, we set a fixed misclassification cost for the target class at $C^+ = 0.8$ according to the prioritization model of I₄.

Table 6.8 summarizes the properties of the new prioritization model obtained. Compared to the prioritization model of I₄, the new prioritization model uses less features (out degree θ_{out} , target downstream effect σ and eccentricity centrality ϑ) but gives a better ROC AUC (based on SFB rank) of 0.915.

Next, we examine the effect of using the new prioritization model on TAPESTRY-prioritization rank and the runtime performance. Figure 6.15 plots the ROC AUC using various weight ratios based on the prioritization model of I₄ (best matched network) and the new prioritization

Figure 6.16: Ground truth found using AUPR.

Figure 6.17: Distance between the ground truth (AUPR) and the ranks obtained using various approaches.

model. In general, the new prioritization model produces better ranking. In terms of execution time, prioritization based on I_4 's prioritization model requires 4.35 seconds as compared to 12.21 seconds when the new prioritization model is used. Note that the additional time taken is for training the new prioritization model.

Hence, a subset of the predictive structural features of the best matched network may produce a better prioritization model for the unseen network. However, optimization of the predictive structural features demands knowledge of a set of known targets which may not always be readily available in unseen networks.

Alternative Performance Metric: AUPR. Finally, we examine the use of an alternative performance metric, AUPR. We make several observations. First, the ground truth established

Figure 6.18: ROC performance of different TAPESTRY variants using best matched model selected with different methods.

using AUPR (Figure 6.16) differs from that found using ROC AUC (Figure 6.4). Second, the best performing FUSSY variant ($\text{FUSSY}_{(A,S)}$) is different from that found using ROC AUC analysis. Note that the best FUSSY outperforms both GDD and NETSIMILE in terms of the distance from the ground truth (Figure 6.17). We compare the performance of the best matched models selected using both ROC AUC and AUPR. Specifically, the best match models found using each approach was used to generate the SFB ranks of the nodes in I_0 which are then combined with the DFB ranks to produce the putative target score used to prioritize the nodes in I_0 . Note that two sets of prioritization ranks are obtained, one for the best matched model selected using ROC AUC and the other for the model selected using AUPR. We then perform ROC and AUPR analysis using the two sets of prioritization ranks. The results are shown in Figures 6.18 to 6.21. From the figures, we observed that ROC AUC-based model selection for the best matched network produced superior prioritization results in terms of both ROC (Figures 6.20) and AUPR (Figures 6.21) analysis.

Figure 6.19: AUPR performance of different TAPESTRY variants using best matched model selected with different methods.

Figure 6.20: Boxplot of results in Figure 6.18.

Figure 6.21: Boxplot of results in Figure 6.19.

6.8 Summary

In conclusion, TAPESTRY is a network feature-based prioritization approach which performs target prioritization of signaling network using the following steps. First, it identifies a best matched network using network similarity ranking. Then, it obtains SFB ranks using the prioritization model of the best matched network. Finally, SFB ranks are aggregated with PSSD ranks using a weighted-sum approach. Amongst the three network similarity ranking approaches we study, FUSSY performs the best. We tested with two performance metrics (ROC AUC and AUPR) and found that the best matched model selected by ROC AUC yields better prioritization results. In particular, PSSD, a dynamic feature, plays an important role in identifying the best matched network. This highlights the importance of considering dynamic features when analyzing networks. In addition, we found that when compared to structural features, PSSD (our proposed dynamic feature) plays a less important role in target prioritization. However, this finding does not undermine the importance of dynamic features since we study more structural features (total of 16) in this thesis and the relative importance of the features can be affected by both the choice of the features studied and the number of features studied. Hence, it would be worthwhile to extend this study to include more dynamic features. However, at present, the lack of available dynamic features makes such study challenging. In the next chapter, we shall focus on the *Target Combination Finder* which uses prioritized ranks to identify target combinations.

Chapter 7

STEROID: Towards *In Silico* Target Combination Identification

In Chapters 5 and 6, we presented our work on target prioritization. In this chapter, we shall present STEROID (HeuriSTic-Based SynErgistic TaRget COmbination IDentification), the *Target Combination Finder* component in TRECENTO. Given a signaling network and a desired *therapeutic effect*, STEROID identifies target combinations that satisfy the *therapeutic effect* and minimize the *off-target effects* on the network. The rest of this chapter is organized as follows. First, we briefly discuss challenges related to the design of combination therapy in Section 7.1. Then, in Section 7.2, we present the motivation of STEROID and how it can address some of these challenges. In Section 7.3, we formally introduce the terminologies used. Next, we define the *target combination search problem* in Section 7.4. The Algorithm STEROID is presented in Section 7.5 and the empirical analysis of our approach is discussed in Section 7.6. The last section concludes this chapter. Preliminary version of this chapter has been published in [67].

7.1 Introduction

The lack of effective mono-therapy treatment for diseases (*e.g.*, cancer) which implicate multiple pathways has driven the effort to explore combination therapies. However, studies such

as [216] reveal that combination therapies are not always more effective than mono-therapies. Hence, it is important to devise strategies for designing good combination therapies. An important step of combination therapy design is to screen for potential drug combinations. Several approaches (reviewed in Chapter 2) including *in silico* model-based [11, 46, 144] screening, model-free screening [47, 161, 386] (*in vitro* and *in vivo*) and *in vitro* high-throughput screening [31, 290, 403] have been proposed. Compared to *in silico* approaches, *in vitro* and *in vivo* approaches are expensive and limited to the screening of a few drug pairs (typically two) [410].

Recall that a good drug combination maximizes the overall *therapeutic effect* while minimizing the *off-target effects*. Informally, the *therapeutic effect* and the *off-target effects* are measures of the intended and the unintended response, respectively, of a biological signaling network to the drug combination. Each drug effect can be simulated *in silico* by modifying appropriate signaling network model parameters. The intended response is the resulting changes to the concentration of the output node, while the unintended response is the resulting changes to the sum of the concentration of the rest of the nodes in the network. For instance, the therapeutic effect of a drug combination for the MAPK-PI3K network (Fig. 3.1), AZD6244 (MEK inhibitor) and sorafenib (Raf inhibitor), can be described as reducing $[ERKPP]$, the node concentration we seek to decrease [82, 405].

Recall that in Chapter 1, identification of such a drug combination typically involves two key steps, namely identification of good target combinations and identification of an appropriate set of drugs hitting these targets. In this thesis, we address the *target combination search problem*, which is complex and non-linear. Informally, this problem involves finding suitable sets of *drug targets* and the required *target activities* (type and extent of perturbations) for these targets for a given signaling network and a therapeutic goal. The complexity of the biological network (numerous potential *drug targets* and wide range of *target activities*) makes performing exhaustive search for sets of targets technically challenging, expensive and time consuming since the number of testable combinations increases exponentially with the number of variables

associated with the network. Tools that can facilitate early detection of inefficacious and/or toxic target combinations *in silico* can serve as a powerful discovery and pre-screening platform when coupled with other complementary technologies such as high-throughput screening.

7.2 Motivation

Few designs of combination therapies are automated. These state-of-the-art *in silico* methods are based on sequential decoding (SD) algorithms [46] or Monte Carlo simulated annealing (MCSA) [161, 392]. Stochastic search algorithms such as MCSA are expected to perform better than SD for non-linear problems [46]. MCSA modifies a candidate solution stochastically and the proposed modification is accepted or rejected using the Metropolis criterion [237]. Although stochastic candidate modification effectively covers the search space by producing a wide variety of candidates, it has two key limitations when used for identifying target combinations. First, drug targets in real signaling networks influence the *therapeutic* and *off-target effects* differently, due to one or more downstream nodes' involvement in other protein-protein interactions [68]. Disregarding this consideration can result in target combinations that satisfy the user-desired *therapeutic effect*, but have excessive *off-target effects*. Second, although the *target activity* affects the combination effects, it is chosen randomly in MCSA. A judicious selection process can provide us an opportunity to improve efficiency of the overall process.

STEROID, a simulated annealing-based approach that leverages heuristics, is designed to address these limitations. The key difference between STEROID and MCSA is in the candidate modification step of the simulated annealing (SA). MCSA modifies the *drug target* and *target activity* stochastically while STEROID uses heuristics to guide the modification. In particular, STEROID leverages on target prioritization methods (*e.g.*, sensitivity analysis and TAPESTRY (described in preceding chapter)) which prioritize potential targets in a given disease-associated network to select more effective targets (better *therapeutic effect* and/or lower *off-target effects*); and on *Loewe additivity theory* (referred to as LOEWE) which assesses the non-additive

effects in a combination drug treatment to select *synergistic*¹ *target activities*. A particularly challenging aspect of analyzing signaling networks is the large number of possible targets and target activities which can make exhaustive testing of combinations computationally intractable. These heuristics help to narrow the vast search space to a manageable region that is potentially enriched with good target combinations. As we shall see later in Section 7.6, the use of heuristics results in faster and more effective solutions.

Note that the goal of STERIOD is to identify *synergistic* combinations of targets with reduced *off-target effects* and is different from the goal of target prioritization which seeks to rank individual potential targets in the network.

7.3 Background

In this section, we briefly discuss two concepts (target prioritization and *Loewe additivity isobologram analysis*) which are exploited in STERIOD. We use the MAPK-PI3K signaling network (Figure 3.1) as a running example in this chapter. In particular, the desired therapeutic effect was set to 50% ERKPP down-regulation. Note that in practice, the therapeutic effect is dependent on the stage of the disease and is typically measured as inhibition of certain phenotypic response (*e.g.*, cell growth) which may not be linearly correlated with the inhibition of the output node concentration. In the sequel, we assume an ordinary differential equation (ODE) model of the signaling network is available.

7.3.1 Target Prioritization

Target prioritization methods assign prioritization rank to individual target nodes based on certain criteria (*e.g.*, the sensitivity of the output node to each target node [362]). Several target prioritization approaches such as LSA [362] and *NetworkPrioritizer* [171] have been proposed

¹Synergy occurs when a pair of targets exerts effects that are greater than the sum of their individual effects, and is generally a beneficial strategy for maximizing effect while minimizing toxicity.

recently. These approaches are reviewed in Chapter 2 and Chapter 6 elaborates on our proposed target prioritization approach TAPESTRY.

Briefly, sensitivity analysis assigns node rank according to the sensitivity value which is the extent of the output node perturbation divided by the extent of parameter (*e.g.*, kinetic rate constant) perturbation. In particular, LSA measures sensitivity by varying a single parameter at a time [362]. *NetworkPrioritizer* and TAPESTRY, on the other hand, are network-based approaches. Unlike *NetworkPrioritizer* which considers only topological features, TAPESTRY uses both topological and dynamic features for target prioritization. A unique aspect of TAPESTRY is its ability to leverage knowledge gained from learning of training networks with known targets to first identify predictive features that characterize targets and then use these predictive features to identify targets of the given network. Note that this is different from *NetworkPrioritizer* which requires users to specify the features to use and the weights of these features. Our study in the preceding chapter reveals that network-based approaches, particularly TAPESTRY, perform better in prioritizing known targets. Hence, we reason that these prioritized targets are likely to form safer and more efficacious combinations.

In this chapter, we shall compare the use of three target prioritization approaches, namely, LSA [362], *NetworkPrioritizer* [171] and TAPESTRY, to identify *synergistic* combinations of targets with reduced *off-target effects*. Note that the goal of this chapter is different from that of the aforementioned target prioritization techniques which is to rank individual target nodes.

7.3.2 Loewe Additivity Theory (LOEWE)

LOEWE assumes that drugs act without self-interaction and determines drug interaction in a combination using the *combination index* [113]. Given a set of drugs X and therapeutic effect T , let D_x and d_x be the doses of drug $x \in X$ required to achieve effect T when used alone and in combination, respectively. Then, the *combination index* is defined as

$$CI = \sum_{x \in X} \frac{d_x}{D_x}. \quad (7.1)$$

Figure 7.1: Isobologram. D_1 and D_2 are the dose of each drug that achieves the desired therapeutic effect if administered alone. d_1 and d_2 together can achieve the same effect.

The combination is *synergistic*, *additive* or *antagonistic* if $CI < 1$, $CI = 1$ or $CI > 1$, respectively. The isobologram (Figure 7.1) provides a visual interpretation of LOEW. It is a graph with the individual drug doses (D_1 and D_2) as its axes. The “line of additivity” is used to interpret the drug interaction. Synergistic and antagonistic combinations are represented by drug doses that fall below and above the line of additivity, respectively [406]. As we shall see later in Section 7.5.2, this theory can be adapted to guide selection of synergistic targets.

7.4 Problem Statement

In this section, we formally define the problem of *target combination search*. We begin by introducing several concepts related to *drug target*. Henceforth, we shall use the notations given in Table 7.1.

7.4.1 Drug Target and Target Activity

First, we present the concept of a *target* and its *activity* (referred to as *target activity*) in the context of signaling networks. A drug asserts its effect on a network through the *target* and the *target activity* is a variable related to the extent of *target* perturbation. The drug effect is typically modeled *in silico* as modulation of the node concentration. The modulation is

Symbol	Description
Γ_c	Target activity of target c .
RPE_u	Reactant-product edge set of node u .
ϕ_u	Concentration-time series profile of node u .
$\text{AUC}(\phi_u)$	Area under the concentration-time series profile curve of node u .
Ψ_X	Ranked list $\{\psi_{X:u_1}, \psi_{X:u_2}, \dots, \psi_{X:u_i}\}$ based on score X where $\psi_{X:u_i}$ is the rank of node u_i .
$\zeta_{th}(c)$	Therapeutic effect of target c .
$\zeta_{off}(c)$	Off-target effects of target c .
λ^+	Selective pressure in the range [1–2].
sp_c	Selection probability of target c .
ϵ	Adjustment factor for relaxing conditions.
t	Temperature parameter of simulated annealing.

Table 7.1: Description of symbols used in this chapter.

achieved by modifying a network parameter that controls the concentration of the node associated with the *target*. This parameter can either be the node's edges (typically represented as ODE reactions) [392] or the node itself (initial concentration) depending on whether the node concentration varies with time. We now formally define these two concepts. We first introduce the notion of *reactant-product edge set* to facilitate exposition. Given a signaling network $G = (V, E)$ and a node $u \in V$, the *reactant-product edge set* of u is defined as $\text{RPE}_u = R_u \cup P_u$ where $R_u \subset E$ and $P_u \subset E$ are the edge sets involving u as reactants and products, respectively.

Definition 7.18 Given a signaling network $G = (V, E)$, and node $u \in V$ with concentration time-series profile ϕ_u and reactant-product edge set RPE_u , the **drug target** of a node u is

$$\begin{cases} c_{fix} = u & \text{if } \phi_u \text{ is constant} \\ c_{var} \in \text{RPE}_u & \text{otherwise} \end{cases}$$

Definition 7.19 Given a drug target c perturbed by drug D with dissociation constant K_D , the **target activity** of c is defined as

$$\Gamma_c = \frac{[D]}{K_D} \quad (7.2)$$

where $[D]$ is the concentration of D .

The ODE modification varies according to the drug type (*e.g.*, activators or inhibitors) and the mechanism of action. Note that there are several different types of activators and inhibitors. In this thesis, we model activation using *nonessential activation* [63], and inhibition using *competitive inhibition* [392] for the following reasons. *Nonessential activator* affects the rate of reaction but does not stop the reaction from happening when it is absent. Since drugs are exogenous to the biological system, we expect them to affect the system only when they are present. *Competitive inhibitors*, on the other hand, is the most common type of inhibition and there are various tools (*e.g.*, HADDOCK [95]) that facilitate the design of these drugs. We assume that a target is druggable by both activators and inhibitors due to the lack of readily available information on the type of action (activation or inhibition) that is valid for each target. Note that violation of this assumption will result in an invalid target combination. However, since STERIOD returns a set of acceptable solutions instead of a single solution, invalid solutions can be discarded in favor of the next valid solution. Note that each reversible reaction is made up of an equivalent pair of irreversible reactions (forward and backward reactions). We explicitly convert all reversible reactions to irreversible reactions using [310] in order to clearly distinguish whether it is the forward or the backward reactions that is activated or inhibited.

Given two nodes u and v , an inhibitor I and an activator A where u and v have constant and variable concentration time-series profiles, respectively, let RPE_v be the reactant-product edge set of v . Then, the targets of u and v , denoted as c_{fix} and c_{var} , respectively, are defined as

$$c_{fix} = u \quad (7.3)$$

$$c_{var} = \frac{V_{max}[S]}{K_m + [S]} \in \text{RPE}_v \quad (7.4)$$

where V_{max} is the maximum velocity; K_m is the Michaelis-Menten constant; and $[S]$ is the concentration of the substrate S .

The *competitive inhibition* of c_{fix} and c_{var} are given by the following equations:

$$I(c_{fix}) = \frac{[u]_0}{\frac{[I]}{K_I}} \quad (7.5)$$

$$I(c_{var}) = \frac{V_{max}[S]}{K_m(1 + \frac{[I]}{K_I}) + [S]} \quad (7.6)$$

In the above equations, $[u]_0$ is the initial concentration of u and K_I is the dissociation constant of I . Similarly, let K_A be the dissociation constant of A . The *nonessential activation* of c_{fix} and c_{var} are defined as follows.

$$\mathcal{A}(c_{fix}) = \frac{[A]}{K_A}[u]_0 \quad (7.7)$$

$$\mathcal{A}(c_{var}) = \frac{V_{max}[S](1 + \frac{[A]}{K_A})}{K_m + [S]} \quad (7.8)$$

For example, the reaction $P \rightarrow P + Akt \xrightarrow{Akt} Akt \rightarrow P \rightarrow P$ whose original ODE is $k[P \rightarrow P][Akt]$ becomes $k[P \rightarrow P][Akt](1 + \frac{[A]}{K_A})$ and $\frac{k[P \rightarrow P][Akt]}{1 + \frac{[I]}{K_I}}$ when modified to simulate nonessential activation and competitive inhibition, respectively.

7.4.2 Target Effects

Next, we formally define the notions of *therapeutic effect* and *off-target effects*. Given a signaling network $G = (V, E)$, a drug target c and the desired therapeutic effect ς_{th} , let $u \in V$ be the node (output node) associated with effect ς_{th} , and $AUC(\phi_u^-)$ and $AUC(\phi_u^+)$ be the area under the concentration-time series profile curve of node u before and after c is perturbed, respectively. Then, the *therapeutic effect* $\varsigma_{th(c)}$ and *off-target effects* $\varsigma_{off(c)}$ of c are given by the following equations.

$$\varsigma_{th(c)} = \frac{|AUC(\phi_u^-) - AUC(\phi_u^+)|}{AUC(\phi_u^-)} \quad (7.9)$$

$$\varsigma_{off(c)} = \sum_{v \in V \setminus u} \left(\frac{|AUC(\phi_v^-) - AUC(\phi_v^+)|}{AUC(\phi_v^-)} \right) \quad (7.10)$$

Note that $\zeta_{th(c)}$ and $\zeta_{off(c)}$ can be determined from *in silico* simulation using *Copasi* [310]. The combination effects are defined similarly whereas $AUC(\phi_u^+)$ and $AUC(\phi_u^-)$ can be estimated using the linear trapezoidal rule method [61].

For example, the therapeutic effect and off-target effects of Akt are given as $\zeta_{th(Akt)} = \frac{|AUC(\phi_{ERKPP}^-) - AUC(\phi_{ERKPP}^+)|}{AUC(\phi_{ERKPP}^-)}$ and $\zeta_{off(Akt)} = \sum_{v \in V \setminus u} (\frac{|AUC(\phi_{ERKPP}^-) - AUC(\phi_{ERKPP}^+)|}{AUC(\phi_{ERKPP}^-)})$, respectively.

7.4.3 Problem Definition

Intuitively, the goal of the *target combination identification problem* is to identify targets and their activities that achieve a user-specified therapeutic effect (*e.g.*, to achieve 50% inhibition of $ERKPP$) while minimizing the off-target effects. Hence, the problem can be modeled as the optimization of a *constraint satisfaction problem* (CSP) which is NP-hard [120]. The CSP is represented as a triple (X, D, C) , where X , D and C represent the set of variables, the variables' domain and the set of constraints, respectively. The element X represents the set of drug targets and target activities; D represents the set of candidate targets in a given disease-related network and the target activity range; and C represents the condition that the combination therapeutic effect must match the desired therapeutic effect. The objective of the *target combination identification problem* is to minimize the combination off-target effects.

Definition 7.20 Given a set of target combination $C = \{C_1, \dots, C_N\}$ and a desired therapeutic effect ζ_{th} , let $C_i = \{c_1, \dots, c_m\}$ where $c_j \in C_i$ is the j^{th} target in the i^{th} combination. Let $\zeta_{off(C_i)}$ and $\zeta_{th(C_i)}$ be the off-target effects and therapeutic effect of combination C_i , respectively. Then, the *target combination identification problem* is defined as

$$C_i = \min\{\zeta_{off(C_i)} | \zeta_{th(C_i)} = \zeta_{th}\} \quad (7.11)$$

7.5 Identifying Target Combination

In this section, we describe the heuristic algorithm STERIOD for identifying target combinations. We begin by providing an overview of STERIOD and discussing the rationale behind key steps of our approach. Then, we present the two heuristics that we shall exploit for modifying candidate solutions. Next, we describe the STERIOD algorithm. Finally, we conclude this section by performing a complexity analysis on the algorithm.

7.5.1 Overview of STERIOD

STERIOD is a heuristics-based simulated annealing algorithm for identifying target combinations. Recall from Chapter 1 that target combination identification requires searching for appropriate targets and their activities from a vast search space that makes exhaustive search impractical. We use simulated annealing, a local search meta-heuristic algorithm, to address the *target combination identification problem* for the following reasons. It is easy to implement, converges and can escape from local optima [147].

Briefly, STERIOD consists of two phases, namely, *network preprocessing* and *heuristics-based simulated annealing*.

In the *network preprocessing* phase, all reversible reactions in the network are converted to equivalent irreversible pairs of reactions and then the reactions are modified to simulate actions of nonessential activators and competitive inhibitors. Recall that in Section 7.4.1, such preprocessing allow us to disambiguate the reaction that shall be perturbed and the type of perturbation required for our predicted target combination solutions. Note that clarity of our target combination solutions can facilitate drug design activity that follows by pinpointing the exact reactions that the drugs should target and how the drugs should affect the reactions.

In the *heuristics-based simulated annealing*, *target prioritization-based heuristic* and *LOEWE-based heuristic* are used for guiding the selection of candidate target combination solutions. In particular, target prioritization-based heuristic leverages on target prioritization rank to select

for more effective targets for the combination whereas LOEWE-based heuristic is used for selecting targets with synergistic activities. Unlike state-of-the-art MCSA-based approaches that generates random candidate solutions, STERIOD uses heuristics to effectively reduce the solution search space leading to efficient discovery of potential solutions. Note that the heuristics are chosen based on their relevance to efficacious therapies. Specifically, in Chapter 5, we demonstrate that network-based target prioritization approaches such as TAPESTRY identified known drug targets and we advocate that preferentially selecting for prioritized targets in the combinations can lead to more effective target combinations. In addition, LOEWE ensures synergistic interaction of targets. That is, it identifies target combinations that achieve therapeutic effects that are greater than the sum of the effects of individual targets in the combination. Synergy is generally a beneficial strategy for maximizing therapeutic effects while minimizing toxicity. Note that in STERIOD, candidate solutions have to satisfy the desired therapeutic conditions and are optimized for off-target effects. Instead of choosing specific off-target effects (*e.g.*, ratio between two proteins), we consider off-target effects in a general sense. That is, deviation of activities of all nodes except the output node before and after perturbation of the target combination. The rationale behind this is that importance of specific off-target effects are dependent on signaling networks and their associated disease and users may not know at the onset which particular off-target effects are more important.

7.5.2 Heuristics

Target prioritization-based heuristics. The goal of using the target prioritization heuristic for target selection is to improve the average solution quality by choosing more effective targets with higher probability, thereby minimizing off-target effects. To achieve this, we first translate the node prioritization rank (Recall from Section 7.3.1) to an equivalent *target rank*, then convert the rank to a *selection probability* value which is used to decide if the target will be accepted. We now introduce these two concepts.

Given a signaling network $G = (V, E)$ and a target prioritization method P , let c_{fix} and c_{var} be the targets of nodes u and v , respectively where u and v have constant and variable concentration-time series, respectively, and $u, v \in V$. Then, the *target ranks* of c_{fix} and c_{var} , denoted as $\Psi_{c_{fix}}$ and $\Psi_{c_{var}}$, respectively, are defined as

$$\Psi_{c_{fix}} = \Psi_{P:u} \quad (7.12)$$

$$\Psi_{c_{var}} = \sum_{w \in W} \Psi_{P:w} \quad (7.13)$$

where $\Psi_{P:u}$ is the rank of u based on P , $W = X \cup Y$, $X, Y \subset V$, and $c_{var} = (X, Y)$.

The *selection probability* (sp) of a target is the likelihood of selecting the target. We use the rank-based fitness function in [32] to obtain a target's selection probability. The fitness function is based on the individual target ranks and avoids scaling problems associated with using actual objective values. It is defined as

$$sp = \frac{2 - \lambda^+}{|\mathcal{T}|(2 - \lambda^+) + 2(\lambda^+ - 1)} \quad (7.14)$$

where $\mathcal{T}$ is the set of individual targets in the signaling network and λ^+ is the parameter (referred to as selective pressure) used to control the expected sampling rate of the individual target [382]. Note that λ^+ is typically in the range [1–2] [32].

Observe that the aforementioned heuristic is independent of any specific target prioritization method. However, as we shall see in Section 7.6, TAPESTRY-based target combination identification typically generates superior quality results compared to MCSA-based techniques as the former exploits structural and dynamics features of the signaling network to improve prioritization of targets.

LOEWE-based heuristic. The effects (Section 7.4.2) resulting from a drug combination can be interpreted as drugs at particular dosages hitting their targets resulting in certain target activities causing a particular response of the network. Hence, an interaction of multiple targets in a combination can be assessed the same way as drug interactions (Section 7.3.2) by replacing

the drug doses with target activities. A target combination is guaranteed to be synergistic if its target activities are chosen from values below the line of additivity. Following from Section 7.3.2, we define the *target interaction* as follows.

Definition 7.21 *Given a therapeutic effect ζ_{th} and a target combination $C = \{c_1, \dots, c_m\}$, let $\Gamma_{0(c_i)}$ and $\Gamma_{(c_i)}$ be the target activities of the i^{th} target in C that achieve ζ_{th} when targeted alone and in combination, respectively. Then, the **target combination index** of C is defined as*

$$TCI_C = \sum_{c_i \in C} \frac{\Gamma_{(c_i)}}{\Gamma_{0(c_i)}} \quad (7.15)$$

*The combination is **synergistic**, **additive** or **antagonistic** if $TCI_C < 1$, $TCI_C = 1$ or $TCI_C > 1$, respectively.*

Following from Definition 7.21, for a 2-target combination $C = \{c_1, c_2\}$, the *synergistic ranges* of targets c_1 and c_2 , denoted as sr_{c_1} and sr_{c_2} , respectively are defined as $sr_{c_1} = [0 - \Gamma_{0(c_1)})$ and $sr_{c_2} = [0 - \Gamma_{(c_2)})$, where $\Gamma_{(c_2)} \in [0 - \Gamma_{0(c_2)})$ and $TCI < 1$. Graphically, these synergistic ranges can be visualized in Figure 7.1 (rightmost isobologram) as “synergistic range₁” and “synergistic range₂”.

7.5.3 The Algorithm STERIOD

Systematic algorithms (*e.g.*, backtracking) that rely on partial instantiation of the candidate solutions to eliminate candidates that violate the constraints have been proposed for solving CSP [38]. In the target combination identification problem, full instantiation of the candidate solution is needed to find the combination effects. Hence, these systematic algorithms cannot be applied effectively. Metaheuristics (*e.g.*, simulated annealing (SA)) are used instead to find approximate solutions as they can achieve good performance results for large combinatorial optimization problems [261]. MCSA (a variant of SA) has been proposed for finding drug target combinations [161, 392], but suffer from certain limitations as highlighted in Section 7.1.

Algorithm 15: Algorithm STERIOD

Input: Signaling network G , set of prioritized node rank Ψ , therapeutic effect ς_{th} and combination size $\mathcal{S}$.

Output: Solution set $\mathcal{R}$.

```

1  $\mathcal{R} \leftarrow \text{INITIALIZE}(\mathcal{R})$ 
2  $(\lambda^+, \epsilon_t, \epsilon_a, N, t_0, i_{max}) \leftarrow \text{SETTODEFAULTS}(\lambda^+, \epsilon_t, \epsilon_a, N, t_0, i_{max})$ 
3  $t \leftarrow t_0$ 
4  $(G', \mathcal{T}, \Gamma_0) \leftarrow \text{PREPROCESSINPUT}(G, \varsigma_{th}, \epsilon_t, t_0, i_{max})$  /*Phase 1*/
5 while  $t \geq 0$  and  $|\mathcal{R}| \leq N$  do
6   foreach iteration  $i=1$  to  $i_{max}$  do
7      $\mathcal{X} \leftarrow \text{GETCOMBI}(\mathcal{T}, \lambda^+, \Psi, \epsilon_a, \Gamma_0, \mathcal{R}, \mathcal{S})$  /*Phase 2.1*/
8      $(\varsigma_{th}(\mathcal{X}), \varsigma_{off}(\mathcal{X})) \leftarrow \text{GETEFFECT}(G', \varsigma_{th}, \mathcal{X})$  /*Phase 2.2*/
9      $\mathcal{R} \leftarrow \text{ACCEPTCOMBI}(\varsigma_{th}(\mathcal{X}), \varsigma_{off}(\mathcal{X}), \varsigma_{th}, \epsilon_a, \epsilon_t, t, \mathcal{R})$  /*Phase 2.3*/
10   $t \leftarrow t - 1$ 

```

We shall now present a novel algorithm called STERIOD (outlined in Algorithm 15) that addresses these limitations by leveraging on target prioritization and LOEWE-based heuristics for modifying drug target and target activity of candidate solutions.

Given a signaling network G , a set of prioritized node rank Ψ , a desired therapeutic effect ς_{th} and the required combination size $\mathcal{S}$, STERIOD identifies a set of synergistic target combination $\mathcal{R}$ which satisfies ς_{th} and has minimal off-target effects ς_{off} . The inputs G and Ψ are used to modify the drug targets and target activities. In addition, G is also used to simulate the target combination effects. Note that the user can specify his preferred target prioritization method for finding Ψ . Several other parameters $(\lambda^+, \epsilon_t, \epsilon_a, N, t_0$ and $i_{max})$ that are required by STERIOD are set to default values, but can be modified if required (Line 2). The parameter λ^+ is used to compute the selection probability of the target. In practice, it is difficult to achieve the therapeutic effect exactly and additive target combinations are generally close to the line of additivity, but seldom “sit” exactly on it. Hence, we specify adjustment factor parameters ϵ_t and ϵ_a to relax the condition for therapeutic effect and additive combination into bound conditions, respectively (e.g., 49.5% to 50.5% inhibition of ERKPP and additive if $0.95 \leq \text{TCI} \leq 1.05$). Finally, the parameters N , t_0 and i_{max} are used to configure the SA and they control when the SA terminates: when N solutions are found or when $t_0 \times i_{max}$ iterations are completed.

Algorithm 16: The PREPROCESSINPUT Procedure (Phase 1)

Input: Signaling network $G = (V, E)$, therapeutic effect ζ_{th} , adjustment factor for therapeutic effect ϵ_t , initial temperature t_0 and maximum number of iterations per temperature cycle i_{max} .

Output: Modified signaling network G' , set of candidate target $\mathcal{T}$ and set of individual target activity Γ_0 .

```

1   $(\mathcal{T}, \Gamma_0) \leftarrow \text{INITIALIZE}(\mathcal{T}, \Gamma_0)$ 
2  foreach  $e \in E$  do
3      if  $\text{ISREVERSIBLE}(e) = \text{true}$  then
4           $(e_{forward}, e_{backward}) \leftarrow \text{GETIRREVERSIBLERXNPAIR}(e)$ 
5           $G \leftarrow \text{REPLACEREVERSIBLEREACTION}(e_{forward}, e_{backward}, e, G)$ 
6  foreach  $v \in V$  do
7      if  $\text{HASCONSTANTTIMESERIESPROFILE}(v) = \text{true}$  then
8           $\mathcal{T} \leftarrow \mathcal{T} \cup v$ 
9      else
10          $\text{RPE} \leftarrow \text{GETREACTANTPRODUCTEDGESET}(v)$ 
11          $\mathcal{T} \leftarrow \mathcal{T} \cup \text{RPE}$ 
12 foreach  $c \in \mathcal{T}$  do
13      $G' \leftarrow \text{MODIFYTARGETREACTION}(G, c)$ 
14      $\Gamma_{0(c)} \leftarrow \text{GETINDIVIDUALTARGETACTIVITY}(G, c, \zeta_{th}, \epsilon_t, t_0, i_{max})$ 
15      $\Gamma_0 \leftarrow \Gamma_0 \cup \Gamma_{0(c)}$ 

```

STEROID consists of two phases, namely, the *preprocessing phase* and the *simulated annealing with heuristics phase*.

Phase 1: Preprocessing. In this phase (Algorithm 16), the reversible reactions in G are first converted into pairs of irreversible reactions using [310] (Lines 4 to 5). Then, the drug targets c_{fix} (Line 8) and c_{var} (Lines 10 to 11) of nodes with constant and variable concentration-time series profiles, respectively are found based on Definition 7.18. Next, the reactions are modified to simulate the effects of the targets when modulated by non-competitive inhibitors and essential activators (Line 13) according to Equations 7.6 to 7.8. Finally, the individual target activities (ITA) Γ_0 required to achieve the desired therapeutic effect (50% down-regulation of ERKPP , $\epsilon_t=5\%$) are found using MCSA configured with the parameters t_0 and i_{max} . Details of the ITA of the MAPK-PI3K network are found in Table 7.2. Targets not listed in Table 7.2 cannot achieve the desired therapeutic effect alone and are deemed to have $\Gamma_0 = \infty$. In Table 7.2,

Target	Modification	Γ_0	Target	Modification	Γ_0
Reaction 1 ^f	inhibition	42.975	Reaction 11	inhibition	180.193
Reaction 1 ^b	activation	5187.249	Reaction 12	activation	0.273
Reaction 2 ^f	inhibition	125.529	Reaction 13	inhibition	7.772
Reaction 2 ^b	activation	29.69	Reaction 14	activation	1.012
Reaction 3 ^f	inhibition	33.62	Reaction 15	inhibition	4.149
Reaction 3 ^b	activation	5857.89	Reaction 16	activation	217.788
Reaction 4	activation	47.246	Reaction 17	inhibition	2.092
Reaction 5 ^f	inhibition	2.081	Reaction 18	activation	1.376
Reaction 5 ^b	activation	10.622	Reaction 19	inhibition	1.746
Reaction 6 ^f	inhibition	8.957	Reaction 20	activation	13.033
Reaction 6 ^b	activation	16.933	Reaction 21	inhibition	2.367
Reaction 7 ^f	inhibition	8.144	Reaction 22	activation	2.344
Reaction 7 ^b	activation	8.295	Reaction 34 ^f	activation	3750.449
Reaction 8 ^f	inhibition	2.231	MKP3	activation	2.755
Reaction 8 ^b	activation	2.243	E	activation	2.294
Reaction 9 ^f	activation	0.381	PP2A	activation	2.315

Table 7.2: Details of individual drug targets in the MAPK-PI3K network achieving the desired therapeutic effect. Each reversible reaction is converted into a pair of equivalent irreversible reactions where ^f and ^b indicate the forward and the backward reactions in the irreversible reaction pair, respectively. Table is read from top to bottom and from left to right.

the targets MKP3, E and PP2A are associated with nodes with constant concentration-time profiles (c_{fix}) while the remaining are associated with nodes with variable concentration-time profiles (c_{var}).

For instance, $PIP3 + Akt \rightleftharpoons Akt + PIP3$ is converted into a pair of irreversible reactions using the procedures GETIRREVERSIBLERXNPAIR and REPLACEREVERSIBLEREACTION. This pair of reactions ($Akt + PIP3 \rightarrow PIP3 + Akt$ and $PIP3 + Akt \rightarrow Akt + PIP3$) are targets (c_{var}) associated with $Akt + PIP3$, Akt and $PIP3$. Then, the reactions are then modified using the MODIFY-TARGETREACTION procedure to $\frac{k[Akt + PIP3](1 + \frac{[A]}{K_A})}{1 + \frac{[I]}{K_I}}$ and $\frac{k[PIP3][Akt](1 + \frac{[A]}{K_A})}{1 + \frac{[I]}{K_I}}$ where k is the affinity constant; A and I are the activator and inhibitor with dissociation constants K_A and K_I , respectively. Finally, the ITA (Γ_0) are found using the GETINDIVIDUALTARGETACTIVITY procedure. Note that these targets have $\Gamma_0 = \infty$ which imply that the MCSA process used to identify the ITA complete the maximum number of iterations without finding any target activity that can

Algorithm 17: The GETCOMBI Procedure (Phase 2.1)

Input: Set of candidate target $\mathcal{T}$, selective pressure λ^+ , set of prioritized node rank Ψ , adjustment factor for target interaction ϵ_a , set of individual target activity Γ_0 , solution set $\mathcal{R}$ and combination size S .

Output: Combination candidate $\mathcal{X} = \{(x_1, \Gamma_1), \dots, (x_S, \Gamma_S)\}$.

```

1  $\mathcal{X} \leftarrow \emptyset$ 
2 foreach combination candidate component  $(x_i, \Gamma_i) = (x_1, \Gamma_1)$  to  $(x_S, \Gamma_S)$  do
3   while ISNULL( $x_i$ ) is TRUE do
4      $\mathcal{A} \leftarrow \text{SELECTRANDOMTARGET}(\mathcal{T} / \mathcal{X}, \mathcal{R})$ 
5      $x_i \leftarrow \text{ACCEPTTARGET}(\mathcal{A}, \mathcal{T} / \mathcal{X}, \Psi, \lambda^+)$ 
6    $\Gamma_i \leftarrow \text{SELECTACTIVITY}(x_i, \mathcal{X}, \Gamma_0, \epsilon_a, \mathcal{R})$ 
7    $\mathcal{X} \leftarrow \mathcal{X} \cup (x_i, \Gamma_i)$ 

```

Algorithm 18: The GETEFFECT Procedure (Phase 2.2)

Input: Signaling network G , therapeutic effect ζ_{th} and candidate combination $\mathcal{X}$.

Output: Combination therapeutic effect $\zeta_{th}(\mathcal{X})$ and combination off-target effects $\zeta_{off}(\mathcal{X})$.

```

1  $(\text{AUC}(\phi^-), \text{AUC}(\phi^+)) \leftarrow \text{SIMULATECOMBINATION}(G, \mathcal{X})$ 
2  $\zeta_{th}(\mathcal{X}) \leftarrow \text{GETTHERAPEUTICEFFECT}(\text{AUC}(\phi^-), \text{AUC}(\phi^+), \zeta_{th})$ 
3  $\zeta_{off}(\mathcal{X}) \leftarrow \text{GETOFFTARGETEFFECTS}(\text{AUC}(\phi^-), \text{AUC}(\phi^+), \zeta_{th})$ 

```

achieve the desired therapeutic effect. Hence, it is likely that these targets cannot achieve the desired therapeutic effect by acting alone.

Phase 2: Simulated Annealing with Heuristics (SAH). The SAH consists of three sub-phases which are repeated until either the temperature t reaches zero or the required number of solutions N is found (Line 5). The subphases consist of *target combination generation* (Line 7), *combination effects calculation* (Line 8) and *candidate acceptance test* (Line 9).

In the GETCOMBI procedure (Algorithm 17), the candidate combination $\mathcal{X}$ consisting of S -target is generated. Lines 3 to 5 implement the target prioritization- and Line 6 the LOEWE-based heuristics. The first target $\mathcal{A}$ is randomly selected using SELECTRANDOMTARGET (Line 4) and accepted in ACCEPTTARGET (Line 5) if the probability of selecting $\mathcal{A}$ (selection probability) is greater than a random number in the range $[0-1]$ (i.e., $sp_{\mathcal{A}} > \text{RAND}(0, 1)$) where $\text{RAND}(\cdot)$ is the random operator). Its activity is then selected within the synergistic range (Definition 7.21) using the SELECTACTIVITY procedure (Line 6). Similar steps are repeated

Algorithm 19: The ACCEPTCOMBI Procedure (Phase 2.3)

Input: Combination therapeutic effect $\zeta_{th}(X)$, combination off-target effects $\zeta_{off}(X)$, therapeutic effect ζ_{th} , adjustment factor for target interaction ϵ_a , adjustment factor for therapeutic effect ϵ_t , temperature t and solution set $\mathcal{R}$.

Output: Solution set $\mathcal{R}$.

```

1  $\mathcal{M} \leftarrow \text{GETCURRENTSOLUTION}(\mathcal{R})$ 
2 if  $\zeta_{off}(X) \leq \zeta_{off}(\mathcal{M})$  and  $\sum_{x \in X} \frac{\Gamma_{\mathcal{R}}}{\Gamma_{\mathcal{R}_0}} < 1 - \epsilon_a$  and  $\frac{|\zeta_{th}(X) - \zeta_{th}|}{\zeta_{th}} \leq \epsilon_t$  then
3    $\mathcal{R} \leftarrow \text{UPDATESOLUTION}(\mathcal{R}, X)$ 
4 else if  $\frac{|\zeta_{th}(X) - \zeta_{th}|}{\zeta_{th}} \leq \epsilon_t$  and  $\text{RAND}(0, 1) \leq e^{-\frac{\zeta_{off}(X) - \zeta_{off}(\mathcal{M})}{t}}$  then
5    $\mathcal{R} \leftarrow \text{UPDATESOLUTION}(\mathcal{R}, X)$ 

```

to find subsequent targets and their activities.

Next, the GETEFFECT procedure (Algorithm 18) obtains the therapeutic and off-target effects by first simulating the candidate solution using *Copasi* [310] (Line 1) and then calculating the therapeutic effect (Line 2) and the off-target effects (Line 3) using Equations 7.10 and 7.10, respectively.

Finally, these effects are used to assess the candidate in ACCEPTCOMBI (Algorithm 19) using the Metropolis criterion. A candidate is accepted under these conditions:

(i) if it is synergistic, achieves the required therapeutic effect and has lower off-target effects lower than the current solution (*curr*) (Line 2), or

(ii) if it achieves the required therapeutic effect and $e^{-\frac{\zeta_{off}(X) - \zeta_{off}(\text{curr})}{t}} \geq \text{RAND}(0, 1)$ (Line 4).

The UPDATESOLUTION procedure updates the solution set and the current solution with X if the candidate is accepted.

For instance, consider a two-target combination. Let backward reaction 29 (denoted as *r29b*) which represents the reaction $\text{AktP} \rightarrow \text{Akt} + \text{P}$, be the first randomly selected target. *r29b* will be accepted if its selection probability $sp_{r29b} > \text{RAND}(0, 1)$. The activity of *r29b* is selected from within the range $[0 - \Gamma_{0(r29b)}]$, where $\Gamma_{0(r29b)}$ is the activity of *r29b* alone that is required to achieve 50% down-regulation of ERKPP with $\epsilon_t=5\%$. The second

target (e.g., $r13$ ($\text{Raf} \rightarrow \text{Raf}^*$)) is randomly selected from the set of candidates excluding $r29b$ (i.e., $\mathcal{T}/r29b$) and will be accepted if $sp_{r13} > \text{RAND}(0, 1)$. The activity of $r13$ is selected from the range $[0 - \Gamma_{r13})$ where $\frac{\Gamma_{(r29b)}}{\Gamma_{0(r29b)}} + \frac{\Gamma_{(r13)}}{\Gamma_{0(r13)}} < 1 - \epsilon_a$ and $\Gamma_{(r29b)} \in [0 - \Gamma_{0(r29b)})$. The therapeutic and off-target effects of the combination $c = \{(r29b, \Gamma_{(r29b)}), (r13, \Gamma_{(r13)})\}$ are computed as $\zeta_{th}(c) = \frac{|\text{AUC}(\Phi_{\text{ERKPP}}^-) - \text{AUC}(\Phi_{\text{ERKPP}}^+)|}{\text{AUC}(\Phi_{\text{ERKPP}}^-)}$ and $\zeta_{off}(c) = \sum_{v \in V_{\text{MAPK}} \setminus \text{ERKPP}} \left(\frac{|\text{AUC}(\Phi_v^-) - \text{AUC}(\Phi_v^+)|}{\text{AUC}(\Phi_v^-)} \right)$, respectively. Finally, the Metropolis criterion is used to assess if the combination will be added into the solution set.

7.5.4 Complexity Analysis

Theorem 7.7 *STERIOD has worst time complexity $O(t_0 \times i_{\max} \times (|V| + |E|) \times |\phi|)$ where t_0 is the initial temperature; $i_{\max}$ is the limit on iterations per cycle; $|V|$ and $|E|$ are the number of nodes and irreversible reactions, respectively, of the given signaling network; and $|\phi|$ is the number of time points in the concentration time-series profiles used to estimate the target effects.*

Proof: In Algorithm 15, the PREPROCESSINPUT procedure performs several tasks: (Task 1) convert reversible reactions to irreversible reactions, (Task 2) identify individual drug targets, (Task 3) modify target reactions and (Task 4) finding ITA. In the worst case, Tasks 1, 2 and 3 require $O(|E|)$, $O(|V||E|)$ and $O(\text{MAX}(|E|, |V|))$ time, respectively, where $\text{MAX}(\cdot)$ is the maximum operator. MCSA, which is used to find the ITA in Task 4, consists of the following steps: generate random candidate solution ($O(1)$); simulate target combination effect using an ODE solver ($O(|E||\phi|)$ [296]); estimate the therapeutic and off-target effects ($O(|V||\phi|)$); and perform candidate acceptance test ($O(1)$), where $|\phi|$ is the number of time points in the concentration-time series profile curve. Hence, the time complexity of each MCSA iteration is $O((|V| + |E|) \times |\phi|)$. In the worst case, no values satisfying the desired therapeutic effect is found when the MCSA terminates on completing all runs ($t_0 \times i_{\max}$ iterations) resulting in a time complexity of $O(\text{MAX}(|E|, |V|) \times t_0 \times i_{\max} \times (|V| + |E|) \times |\phi|)$ to obtain the ITA for all

targets. The time complexity of PREPROCESSINPUT can be further reduced to $O(t_0 \times i_{max} \times (|V| + |E|) \times |\phi|)$.

In the SAH phase, in the worst case, SAH terminates on completing all runs ($t_0 \times i_{max}$ iterations) without finding N solutions. Similar to the MCSA, each SAH iteration consists of the following steps: (Task 5) generate candidate solution, (Task 6) simulate combination effects using an ODE solver, (Task 7) estimate combination effects and (Task 8) perform candidate acceptance test. Task 5 involves target prioritization-based heuristic for selecting the target. This heuristics accepts higher prioritized targets with higher probability (Section 7.5.2). In the worst case, the lowest prioritized target (y) from the set of individual targets ($\mathcal{T}$) is always considered. y will be accepted if a randomly generated number is lower than the selection probability of y (sp_y). It takes about $O(|\mathcal{T}|)$ tries in order to accept the worst candidate. Hence, the complexity of generating the candidate solution is $O(|X||\mathcal{T}|)$ where $|X|$ is the size of the candidate combination and $|\mathcal{T}|$ is the size of the candidate target. Tasks 6, 7 and 8 take $O(|E||\phi|)$ [296], $O(|V||\phi|)$ and $O(1)$ time, respectively. Since $|\phi| \gg |X|$ and $O(|E|) = O(|\mathcal{T}|)$, the time complexity of SAH is $O(t_0 \times i_{max} \times (|V| + |E|) \times |\phi|)$ time in the worst case. Thus, the overall worst time complexity of STERIOD is $O(t_0 \times i_{max} \times (|V| + |E|) \times |\phi|)$.

Theorem 7.8 STERIOD has worst space complexity $O(|V|(|E| + |\phi|))$ where $|V|$ and $|E|$ are the number of nodes and irreversible reactions, respectively, of the given signaling network, and $|\phi|$ is the number of time points in the concentration time-series profiles used to estimate the target effects.

Proof: In Algorithm 15, $O(|V| + |E|)$ is needed for storing the input signaling network G and set of prioritized node rank Ψ . Recall that the PREPROCESSINPUT procedure performs several tasks: (Task 1) convert reversible reactions to irreversible reactions, (Task 2) identify individual drug targets, (Task 3) modify target reactions and (Task 4) finding ITA. In the worst case, Tasks 1, 2 and 3 require $O(2|E|)$, $O(2|V||E|)$ and $O(\text{MAX}(|E|, |V|))$ space, respectively,

where $\text{MAX}(\cdot)$ is the maximum operator. Recall that MCSA, which is used to find the ITA in Task 4, consists of the following steps: generate random candidate solution ($O(1)$); simulate target combination effect using an ODE solver ($O(|V||\phi|)$); estimate the therapeutic and off-target effects ($O(|V||\phi|)$); and perform candidate acceptance test ($O(1)$), where $|\phi|$ is the number of time points in the concentration-time series profile curve. Hence, the space complexity of MCSA is $O(|V||\phi|)$. Hence, PREPROCESSINPUT has space complexity $O(|V|(|E| + |\phi|))$.

In the SAH phase, in the worst case, SAH returns N solutions which requires $O(\mathcal{S} \times N)$ storage space. Similar to the MCSA, SAH consists of the following steps: (Task 5) generate candidate solution ($O(\mathcal{S})$), (Task 6) simulate combination effects using an ODE solver ($O(|V||\phi|)$), (Task 7) estimate combination effects ($O(|V||\phi|)$) and (Task 8) perform candidate acceptance test ($O(1)$). Hence, the SAH phase requires $O(\mathcal{S} \times N + |V||\phi|)$ space. Since in most applications, we expect $|V||\phi| > \mathcal{S} \times N$, the space complexity of the SAH phase can be further reduced to $O(|V||\phi|)$. Thus, the overall worst time complexity of STERIOD is $O(|V|(|E| + |\phi|))$.

7.6 Experiments

In this section, we describe the experimental settings and discuss the experimental results obtained. In particular, the experiments are designed to answer the following questions.

- (i) Q1: How do the heuristics affect the combination target solutions found?
- (ii) Q2: Which prioritization approach results in better combination targets?
- (iii) Q3: Are the combination targets found biologically relevant?
- (iv) Q4: How does STERIOD compare with the MCSA-based approach?
- (v) Q5: How does STERIOD perform across different signaling networks?

7.6.1 Experimental Setting

STEROID is implemented using Java. In addition, we use several publicly available tools and libraries that are listed below:

- (i) *libSBML* library [37]: for reading and processing the SBML files of the networks.
- (ii) *Copasi* [152]: *Copasi* is a simulation tool for biological networks and offers several functionalities including parameter estimation, sensitivity analysis and steady-state analysis. We use *Copasi* for performing LSA-based target prioritization, and its java API to simulate the target perturbation effects in STEROID.
- (iii) *NetworkPrioritizer* [171] plugin for *Cytoscape* [320]: for performing *Weighted Borda Fuse* (WBF)-based and *Weighted AddScore Fuse* (WASF)-based target prioritization.

The experiments are performed on a computer system using a 64-bit operating system with 8GB RAM and a dual core processor running at 3.60GHz.

Summary of Approaches Studied. Recall that in Section 7.5.2 we study two heuristics. In particular, the target prioritization-based heuristic is orthogonal to any specific target prioritization technique and we investigate different target prioritization approaches, namely, TAPESTRY (detailed in Chapter 6), LSA [139] and *NetworkPrioritizer* [171] in the comparative study. Recall that *NetworkPrioritizer* provides two prioritization approaches, namely WBF and WASF. We study both approaches and compare them against TAPESTRY and LSA. Table 7.3 summarizes the different approaches studied and the heuristics used in each approach.

Note that modification of the candidate differs depending on the heuristics used. For MCSA, the targets and activities are selected randomly. For STEROID-TAPESTRY, STEROID-LSA, STEROID-WBF and STEROID-WASF, the targets are selected using Algorithm 17 (Lines 3 to 5) while the activities are selected randomly. For STEROID-LOEWE, the targets are selected

Approach	ID	LOEWE	TAPESTRY	LSA	WBF	WASF
MCSA	1					
STEROID-LOEWE	2	✓				
STEROID-TAPESTRY	3		✓			
STEROID-LSA	4			✓		
STEROID-WBF	5				✓	
STEROID-WASF	6					✓
STEROID-LOEWE TAPESTRY	7	✓	✓			
STEROID-LOEWE LSA	8	✓		✓		
STEROID-LOEWE WBF	9	✓			✓	
STEROID-LOEWE WASF	10	✓				✓

Table 7.3: The different approaches studied. ✓ indicates the heuristic used by each approach. For instance, STEROID-LOEWE TAPESTRY uses LOEWE and TAPESTRY prioritization-based heuristics.

randomly while the activities are selected from within the synergistic range (Algorithm 17, Line 6). For STEROID-LOEWE TAPESTRY, STEROID-LOEWE LSA, STEROID-LOEWE WBF and STEROID-LOEWE WASF, the targets and activities are selected using Algorithm 17 (Lines 3 to 6).

Note that we did not perform comparison against TIMMA and OPMET for the following reasons:

- (i) TIMMA requires knowledge of target binding assays and high-throughput drug sensitivity screens specific for the networks and these are not available for the networks that we studied.
- (ii) OPMET requires enumeration of the entire search space. For the networks that we are investigating, the search space is infinitely large due to the wide range of possible target activities and full enumeration of the entire space is impractical.

Performance Metrics. Recall that the goal of STEROID is to identify target combinations which satisfy the therapeutic effect and have minimized off-target effects. Hence, we assess the different approaches based on several criteria:

- (i) off-target effects of target combinations,
- (ii) runtime performance, and
- (iii) actual number of solutions found.

STEROID Configuration. Unless otherwise stated, the combination size shall be set to 2 and the following default values shall be used for the rest of the parameters: $\{t_0 = 100, i_{max} = 500, N = 50, \epsilon_t = 5\%, \epsilon_a = 5\%, \lambda^+ = 1.8\}$. We used *Copasi* java API for estimating the combination effects and its parameters were set as follows: $\{trajectory\ task = \text{“Time-Course”}, method\ type = \text{deterministic}, absolute\ tolerance = 1 \times 10^{-12}\}$.

For all tables and discussion that follow in this chapter, the terms ACT and IN shall denote activators and inhibitors, respectively, whereas forward and backward reactions are marked with superscripts f and b , respectively.

7.6.2 Experimental Results

We performed several sets of experiments to compare the different variants of STERIOD against the state-of-the-art approach (MCSA). First, we examine the effects of using different heuristics on the MAPK-PI3K network to determine the best STERIOD variant. Then, we assess the biological relevance of the target combinations identified. Next, we examine effects of parameter variations. Note that figures in this subsection uses the approach IDs given in Table 7.3.

Effects of Heuristics. In this set of experiments, we investigate the effects of using different heuristics (*i.e.*, different variants of STERIOD) on the target combinations found in the MAPK-PI3K network. First, we assess different STERIOD variants in terms of off-target effects, actual size² of solution sets found and execution time. Figure 7.2 summarizes the effects

²Note that in the experiments, although we set the required number of solution to be 50, the actual number of solution found may be lesser.

of heuristics. In all our experiments, we use MCSA (Approach 1) as our baseline comparison. We make several observations. First, heuristics can be used to reduce off-target effects. This is illustrated by the boxplot in Figure 7.2. The median off-target effects of all STERIOD variants (Approaches 2 to 10) are either the same or lower than that of MCSA. Second, the use of LOEWE-based heuristic produces fewer outliers, reduces off-target effects and cuts down execution time more effectively than target prioritization-based heuristic. In addition, compared to approaches that use only target prioritization-based heuristic, those approaches that use LOEWE-based heuristic have larger actual solution set sizes. These results suggest that LOEWE-based heuristic may have a greater role to play in influencing off-target effects, actual solution set size and the execution time required to find target combinations. Third, the use of LOEWE-based heuristic together with an appropriate target prioritization-based heuristic may help to reduce the minimum off-target effects. For instance, in Approaches 7 (STERIOD-LOEWETAPESTRY) and 9 (STERIOD-LOEWEWBF), the minimum off-target effects are 1.037 and 1.069, respectively. These minimum values are lower³ than the corresponding approaches that use only a single heuristic (*i.e.*, STERIOD-LOEWE, STERIOD-TAPESTRY and STERIOD-WBF).

Next, we examine the characteristics of solutions found by using different STERIOD variants. The solutions are characterized based on target interaction (synergistic, additive or antagonistic) and combination type (activators, inhibitors, or mixed activator and inhibitor). We make the following observations from Figure 7.3. First, target prioritization-based heuristic appears to favor identification of antagonistic target combinations. Antagonistic drug combinations are found to be useful in combating drug resistance in various diseases including cancer [29]. Hence, this heuristic may be particularly useful in identifying combinations that can overcome drug resistance. Second, approaches incorporating the LOEWE-based heuristic tend to yield more solutions that are made up of only inhibitors. Note that development of drugs that

³Approaches 2 (STERIOD-LOEWE), 3 (STERIOD-TAPESTRY) and 5 (STERIOD-WBF) have minimum off-target effects of 1.073, 1.053 and 9.971, respectively.

Figure 7.2: Effect of heuristics.

Figure 7.3: Characterization of solution sets.

PMID	Target 1	Target 2
22180401, 21062259	Akt inhibitor	MEK inhibitor
21632463	PI3K inhibitor	MEK inhibitor
14675307	PI3K inhibitor	Akt inhibitor

Table 7.4: *PubMed* results relevant to ovarian cancer drug combinations targeting the MAPK-PI3K network.

Target in Table 7.4	Inhibition Mechanism	Corresponding target(s) in MAPK-PI3K network
Akt inhibitor	Disruption of Akt binding to its membrane localizing factor (PIP3) [264] or dephosphorylation of PIP3 [228]	Activators of Reaction 29 ^b , Reaction 30, Reaction 33; Inhibitors of Reaction 29 ^f , Reaction 31, Reaction 32
MEK inhibitor	MEK dephosphorylation [240] or blockade of MEK phosphorylation [119]. Is also used to achieve ERK inhibition as there is no known ERK inhibitors.	Activators of Reaction 16, Reaction 18, Reaction 20 or Reaction 22; Inhibitors of Reaction 15, Reaction 17, Reaction 19 or Reaction 21
PI3K inhibitor	Inhibits PI3K in ATP-competitive manner [228]	Activators of Reaction 24 ^b , Reaction 26; Inhibitor of Reaction 24 ^f

Table 7.5: Mapping between targets in Table 7.4 and MAPK-PI3K network.

inhibit PPI is perceived to be easier than drugs that activate PPI. This is because in the design of drugs that are activators, there is a need to achieve good binding in order to replicate the protein interaction and stimulate increase in target activity [133]. Hence, LOEWE-based heuristic may also be useful in enriching the solution set with inhibitor-only target combinations.

In particular, Approach 7 (STERIOD-LOEWETAPESTRY) has the lowest median (8.006) and minimum (1.037) off-target effects. Hence, Approach 7 is the best STERIOD variant.

Biological Relevance of Target Combinations. We shall now examine the biological relevance of the target combinations found using the best STERIOD variant, STERIOD-LOEWETAPESTRY and the baseline approach, MCSA.

First, we examine the ability of both approaches in finding a set of benchmark target combination relevant to ovarian cancer and targeting the MAPK-PI3K network. The benchmark

Approach	Target 1 [Activity]	Target 2 [Activity]	ζ_{off}	Combination
STEROID- LOEWETAPESTRY	Reaction 33 ACT [1633.218]	Reaction 16 ACT [185.982]	11.889	Akt inhibitor + MEK inhibitor
MCSA	Reaction 16 ACT [323.857]	Reaction 31 IN [1640.247]	15.534	MEK inhibitor + Akt inhibitor

Table 7.6: Known combinations in solution sets.

combination set is curated from literature in the *PubMed* repository using “ovarian”, “cancer”, “combination” as keywords. Among 5863 *PubMed* records⁴ that were returned, we find 3 combinations (Table 7.4) fitting the criteria. Table 7.5 shows the targets in the MAPK-PI3K network that match those in Table 7.4. We examine our solution sets to identify those combinations involving the targets in Table 7.5. Both solution sets contain one known combination (reported in Table 7.6). In particular, the target activities and off-target effects of the combination predicted by STEROID-LOEWETAPESTRY are lower than that predicted by MCSA.

In addition, we observe that a large percentage of solution set found using STEROID-LOEWETAPESTRY contain at least one target in known drug combinations compared to MCSA (Figure 7.4). In particular, the top-10 solutions (Table 7.7) in STEROID-LOEWETAPESTRY (top-10 solutions with least off-target effects) are highly enriched with these known targets, particularly the MEK (or ERK) inhibitor. This finding suggests that the MEK inhibitor may be a good target to be used in combinations.

Then, we examine the biological relevance of the top-10 solutions found using both approaches. In the STEROID-LOEWETAPESTRY solution set, 2 target combinations (combinations ST5 and ST10) involve large⁵ target activities whereas in MCSA, all top-10 solutions implicate large target activities. Note that large target activity implies either high drug concentration or very small dissociation constant, and is an indication of a high potential in having side effects, especially if the treatment regime requires repeated drug dosing [74]. Hence, MCSA tends to yield target combinations that require large target activities, increasing the potential

⁴as of 17 March 2012.⁵Target activities greater than 1000.

Figure 7.4: Percentage of STERIOD-LOEWETAPESTRY and MCSA solution sets with targets in known drug combinations.

for side effects. In addition, we perform literature search to look for references that support the usefulness of these combinations. We search *PubMed* specifically for publications related to the predicted combinations in Tables 7.7 and 7.8. We summarize these findings below:

- (i) ST1: Studies reveal that MEK inhibition exhibits the best clinical response in the basal subtype of ovarian cancer where there is no mutation of the oncogenes, RAS and RAF [241]. RAF mutation is common in ovarian cancer and high level of Raf-1 activity was found to correlate with the advanced stage of ovarian cancer [131]. In [205], Li et al. establish the role of Raf kinase inhibitor protein (RKIP) as a metastasis suppressor gene. Based on the above, we reason that combination ST1 consisting of an ERK phosphatase and a Raf inhibitor may be useful in treating the advance stage of ovarian cancer.
- (ii) ST2 and ST3: Profound growth inhibition and apoptosis were observed in ovarian cancer cells treated with CI-1040 , a MEK1/2 inhibitor [282]. These ovarian cells carry mutations in KRAS or BRAF and they typically overexpress DUSP4 , an ERK phosphatase [325]. Note that there is currently no known inhibitor that acts directly on ERK and ERK inhibition is typically achieved through MEK inhibitors [301]. Hence, this correlates with our computational prediction of combinations ST2 and ST3 involving ERK phosphatase activator and ERK (or MEK) kinase inhibitor.

No.	Target 1 [Activity]	Target 2 [Activity]	ζ_{off}	Combination	References/ Comments
ST1	Reaction 22 ACT* [2.142]	Reaction 14 ACT [0.009]	1.037	ERK phosphatase activator + Raf inhibitor	[131, 205, 241]
ST2	Reaction 22 ACT* [2.209]	Reaction 15 IN* [0.097]	1.091	ERK phosphatase activator + MEK kinase inhibitor	[282, 325]
ST3	Reaction 21 IN* [2.080]	Reaction 22 ACT* [0.203]	1.092	ERK kinase inhibitor + ERK phosphatase activator	[282, 325]
ST4	Reaction 21 IN* [2.174]	Reaction 17 IN* [0.068]	1.097	ERK kinase inhibitor + MEK kinase inhibitor	Mono-therapy
ST5	Reaction 22 ACT* [2.150]	Reaction 1 ^b IN [7575.148]	1.138	ERK phosphatase activator + Inhibitor of dissociation of heregulin from ErbB4 receptor	-
ST6	Reaction 19 IN* [1.623]	Reaction 21 IN* [0.089]	1.598	ERK kinase inhibitor + ERK kinase inhibitor	Mono-therapy
ST7	Reaction 19 IN* [1.636]	Reaction 12 ACT [0.012]	1.796	ERK kinase inhibitor + Ras inhibitor	[254, 375]
ST8	Reaction 18 ACT* [1.299]	Reaction 19 IN* [0.011]	2.411	MEK phosphatase activator + ERK kinase inhibitor	-
ST9	Reaction 17 IN* [1.078]	Reaction 18 ACT* [0.399]	2.458	MEK kinase inhibitor + MEK phosphatase activator	-
ST10	Reaction 18 ACT* [1.288]	Reaction 1 ^b IN [7803.570]	2.519	MEK phosphatase activator + Inhibitor of dissociation of heregulin from ErbB4 receptor	-

Table 7.7: Top-10 target combinations for MAPK-PI3K network found using STERIOD-LOEWETAPESTRY. Note that targets present in Table 7.5 are marked with *.

- (iii) ST7 and MT3: We did not find any supporting evidence that combinations ST7 and MT3 have been performed, successfully or otherwise, in experiments. However, individual components of these combinations have shown efficacy in ovarian cancer [254,375,402]. Hence, they warrant further investigation as potential target combinations.
- (iv) MT4: Corresponds to the known target combination consisting of a MEK inhibitor and a Akt inhibitor [23] in Table 7.4.

Note that combinations ST4, ST6, MT8 and MT10 are akin to mono-therapies as they involve the same type of drugs (*i.e.*, ERK or MEK kinase inhibitor, and tyrosine kinase inhibitor). In addition, we observe that MCSA produces many (60%) combinations (combinations MT1, MT2, MT5, MT6, MT7 and MT9) that involve promoters of known mediators of cancer (*e.g.*,

No.	Target 1 [Activity]	Target 2 [Activity]	ζ_{off}	Combination	References/ Comments
MT1	Reaction 13 ACT [1294.274]	Reaction 14 ACT [7706.958]	2.945	Raf activator + Raf inhibitor	-
MT2	PP2A IN [26.286]	Reaction 3 ^b IN [6175.158]	6.827	PP2A inhibitor + tyrosine kinase activator	-
MT3	Reaction 16 ACT* [220.584]	Reaction 4 IN [7444.068]	15.409	MEK phosphatase activator + inhibitor of receptor dimerization	[375,402]
MT4	Reaction 16 ACT* [656.857]	Reaction 31 IN* [1640.247]	15.534	MEK phosphatase activator + PIP kinase inhibitor	[23]
MT5	Reaction 26 IN [6535.419]	Reaction 3 ^b ACT [6033.846]	17.329	PI3K activator + tyrosine kinase inhibitor	-
MT6	Reaction 3 ^b ACT [5730.172]	Reaction 16 IN [6892.326]	18.375	tyrosine kinase inhibitor + MEK phosphatase inhibitor	-
MT7	Reaction 16 IN [6163.809]	Reaction 3 ^b ACT [5845.658]	18.455	MEK phosphatase inhibitor + tyrosine kinase inhibitor	-
MT8	Reaction 3 ^b ACT [5551.488]	Reaction 34 ^b IN [7065.998]	18.486	tyrosine kinase inhibitor + tyrosine kinase inhibitor	Mono-therapy
MT9	Reaction 3 ^b ACT [5541.069]	Reaction 1 ^b IN [8923.956]	18.522	tyrosine kinase inhibitor + inhibitor of dissociation of heregulin from ErbB4 receptor	-
MT10	Reaction 3 ^b ACT [5758.748]	Reaction 34 ^b IN [942.042]	18.679	tyrosine kinase inhibitor + tyrosine kinase inhibitor	Mono-therapy

Table 7.8: Top-10 target combinations for MAPK-PI3K network found using MCSA. Note that targets present in Table 7.5 are marked with *.

PI3K [76]). Such target combinations, though counter-intuitive, can still achieve the therapeutic goal if the effect of other targets can offset the pro-cancer signals, resulting in an overall anti-cancer signal. However, extra caution should be exercised for these combinations since improper management of the balance between the pro- and anti-cancer signal can easily aggravate the cancer. We did not find any supporting evidence for the remaining combinations (combinations ST8 and ST9). Hence, compared to MCSA, STERIOD-LOEWETAPESTRY is able to identify target combinations that are biologically more relevant.

Next, we examine the bottom-10 solutions (Tables 7.9 and 7.10). We note that 60% of STERIOD-LOEWETAPESTRY solutions and 100% of MCSA solutions have at least one target with activity greater than 1000. Hence, compared to the top-10 solutions, the bottom-10 solutions have more combinations with large target activities. This implies that combinations with

No.	Target 1 [Activity]	Target 2 [Activity]	ζ_{off}	Combination	Reference/ Comments
SB1	Reaction 3 ^f IN [29.517]	Reaction 18 ACT* [0.043]	17.866	tyrosine kinase inhibitor + MEK phosphatase	-
SB2	Reaction 3 ^b ACT [5556.930]	Reaction 5 ^f IN [0.042]	18.689	tyrosine kinase inhibitor + inhibitor of RFP and Shc binding	-
SB3	Reaction 4 ACT [44.387]	Reaction 26 ACT* [102.262]	19.228	promoter of receptor dimerization + PI3K inhibitor	-
SB4	Reaction 3 ^f IN [20.701]	Reaction 29 ^f ACT* [5784.119]	19.671	tyrosine kinase inhibitor + Akt activator	-
SB5	Reaction 29 ^f ACT [8541.949]	Reaction 3 ^b ACT [4481.641]	20.635	PI3K inhibitor + tyrosine kinase inhibitor	[129]
SB6	Reaction 1 ^f IN [13.847]	Reaction 30 ACT* [9707.636]	21.885	tyrosine kinase inhibitor + PIP3 phosphatase activator	-
SB7	Reaction 21 IN* [2.153]	Reaction 1 ^b ACT [158.909]	22.021	ERK kinase inhibitor + promoter of dissociation of heregulin from ErbB4 receptor	[231,375]
SB8	Reaction 1 ^f IN [31.578]	Reaction 16 ACT* [9.148]	35.070	tyrosine kinase inhibitor + MEK phosphatase	-
SB9	Reaction 1 ^f IN [37.157]	Reaction 9 ^b ACT [7084.384]	35.086	tyrosine kinase inhibitor + promoter of GS and ShP binding	-
SB10	Reaction 25 ^b ACT [2859.395]	Reaction 11 IN [164.144]	71.650	promoter of PI3K binding with activated ErbB4 receptor + MEK kinase inhibitor	-

Table 7.9: Bottom-10 target combinations for MAPK-PI3K network found using STERIOD-LOEWETAPESTRY. Note that targets present in Table 7.5 are marked with *.

smaller target activities are likely to produce combinations with smaller off-target effects.

In addition, we note that the targets in the bottom-10 STERIOD-LOEWETAPESTRY solutions are mostly located upstream in the MAPK-PI3K network. This is in contrast with the top-10 STERIOD-LOEWETAPESTRY solutions where targets are positioned further downstream and nearer to the output node (ERKPP). No particular trends exist for the MCSA solutions.

The literature curation findings for the bottom-10 solutions are summarized as follows:

- (i) SB5: Glaysher et al. investigated combinations of EGFR inhibitors and PI3K inhibitors and found that certain combinations (e.g., ZSTK474 with erlotinib and gefitinib) exhibit enhanced synergistic activity [129]. This correlates well with combination SB5.
- (ii) SB7, MB6, MB7 and MB9: We did not find any supporting evidence that combinations

No.	Target 1 [Activity]	Target 2 [Activity]	ζ_{off}	Combination	Reference/Comments
MB1	Reaction 34 ^f ACT [3797.693]	Reaction 24 ^b IN [5956.664]	66.220	promoter of receptor internalization + PIP3K activator	-
MB2	Reaction 23 ^b ACT [162.172]	Reaction 34 ^f ACT [3800.193]	66.247	promoter of dissociation of PIP3K from RP + promoter of receptor internalization	-
MB3	Reaction 15 ACT [5993.379]	Reaction 6 ^f IN [148.058]	97.426	MEK kinase activator + Shc kinase inhibitor	-
MB4	Reaction 17 ACT [3168.839]	Reaction 5 ^b ACT [7628.053]	107.372	MEK kinase activator + promoter of dissociation of Shc from RP	-
MB5	Reaction 11 ACT [20.804]	Reaction 5 ^f IN [9864.965]	116.578	MEK kinase activator + inhibitor of RP and Shc binding	-
MB6	Reaction 4 IN [8413.204]	Reaction 1 ^b ACT [6145.133]	122.430	inhibitor of receptor dimerization + promoter of dissociation of heregulin from ErbB4 receptor	[3,262]
MB7	Reaction 30 ACT* [4506.434]	Reaction 1 ^b ACT [3407.957]	256.984	PIP3 phosphatase activator + promoter of dissociation of heregulin from ErbB4 receptor	[246,262]
MB8	Reaction 1 ^b ACT [5221.591]	Reaction 10 IN [5680.547]	303.923	promoter of dissociation of heregulin from ErbB4 receptor + Shc phosphatase inhibitor	-
MB9	Reaction 32 IN* [5538.346]	Reaction 1 ^b ACT [5418.050]	308.214	PIP3 kinase inhibitor + promoter of dissociation of heregulin from ErbB4 receptor	[154,262]
MB10	Reaction 1 ^b ACT [3556.864]	Reaction 28 IN [6870.239]	3998.338	promoter of dissociation of heregulin from ErbB4 receptor + PIP3 phosphatase inhibitor	-

Table 7.10: Bottom-10 target combinations for MAPK-PIP3K network found using MCSA. Note that targets present in Table 7.5 are marked with *.

SB7, MB6, MB7 and MB9 have been performed, successfully or otherwise, in experiments. However, individual components of these combination have shown efficacy in ovarian cancer [3, 154, 231, 246, 262, 375]. Hence, they warrant further investigation as potential target combinations.

We note that several of the targets identified in the combinations (combinations SB3, SB4, SB9, SB10, MB1, MB3, MB4, MB5 and MB10) involve promoters of known mediators of cancer.

We did not find any supporting evidence for the remaining combinations (combinations SB1, SB2, SB6, SB8, MB2 and MB8). Although the bottom-10 combinations contain several biologically relevant combinations, many of them have relatively large off-target effects (MB6, MB7 and MB9) and involve targets with large activities (SB5, MB6, MB7 and MB9). In addition, 70% of combinations found using STERIOD-LOEWETAPESTRY contain tyrosine kinase inhibitors (TKI). Clinical studies have found that although treatments with tyrosine kinase inhibitors (TKI) sometimes produce promising results, most treatment lose their effectiveness soon due to resistance often caused by activating mutations in downstream effectors of the tyrosine kinases [326]. Apart from drug resistance, another potential issue of TKI is toxicity (*e.g.*, nephrotic syndrome) due to the disruption of multiple downstream signaling pathways of the tyrosine kinases which are involved in normal organ functioning [134]. Hence, designing combinations involving TKI requires understanding the idiosyncrasy of a patient's genome in order to select other suitable targets for the combinations which can minimize TKI-resistance. The activity of TKI in the combinations should also be kept low to reduce potential toxicity. However, the predicted combinations that involve TKI generally have high activity level, making them less than ideal as safe target combinations. Hence, target combinations with larger predicted off-target effects may be indicative of less effective and/or more toxic combinations in the real world.

Finally, we apply STERIOD-LOEWETAPESTRY on another⁶ signaling network, glucose-stimulated insulin secretion network [168], for further validation on its ability in identifying biological relevant combinations. STERIOD was configured with the following settings:

(i) Network: glucose-stimulated insulin secretion network in [168].

(ii) Therapeutic goal: 25% increase in mitochondrial ATP⁷.

⁶Note that we did not present the results for the Ras activation network because STERIOD encountered error due to numerical instability in the *Copasi* ODE solver.

⁷Recall that in Figure 4.5A, accumulation of mitochondrial ATP results in activation of insulin granule exocytosis. Hence we set an arbitrary increase of 25% increase in mitochondrial ATP as our therapeutic goal.

No.	Target 1 [Activity]	Target 2 [Activity]	ζ_{off}	Combination	Reference
1	Reaction 7 IN [12.479]	Reaction 38 IN [8483.898]	97.318	glyceraldehyde 3-phosphate dehydrogenase inhibitor + FAD dependent glycerol-3-phosphate dehydrogenase inhibitor	-
2	Reaction 7 IN [12.998]	Reaction 17 IN [5079.313]	115.524	glyceraldehyde 3-phosphate dehydrogenase inhibitor + oxoglutarate dehydrogenase com- plex inhibitor	-
3	Reaction 10 IN [31.713]	Reaction 45 ACT [3258.442]	126.778	lactate dehydrogenase inhibitor + malate dehydrogenase activator	-
4	Reaction 36 ACT [4371.772]	Reaction 10 IN [45.927]	2470.605	glutathione reductase activator + lactate dehydrogenase inhibitor	-
5	Reaction 38 IN [8974.797]	Reaction 10 IN [50.682]	2531.171	FAD dependent glycerol-3-phosphate dehydrogenase inhibitor + lactate dehydrogenase inhibitor	[222, 243, 293]
6	Reaction 38 ACT [4537.414]	Reaction 10 IN [50.993]	2576.963	FAD dependent glycerol-3-phosphate dehydrogenase activator + lactate dehydrogenase inhibitor	[222, 243, 293]
7	Reaction 36 ACT [2953.557]	Reaction 10 IN [52.789]	2583.011	glutathione reductase activator + lactate dehydrogenase inhibitor	-
8	Reaction 21 IN [2739.023]	Reaction 10 IN [52.789]	2613.709	mitochondrion malate dehydrogenase inhibitor + lactate dehydrogenase in- hibitor	-
9	Reaction 38 IN [1189.972]	Reaction 7 IN [21.114]	2615.501	FAD dependent glycerol-3-phosphate dehydrogenase inhibitor + oxoglutarate dehydrogenase com- plex inhibitor	-
10	Reaction 36 ACT [784.932]	Reaction 10 IN [53.747]	2624.339	glutathione reductase activator + lactate dehydrogenase inhibitor	-

Table 7.11: Target combinations for glucose-stimulated insulin secretion network found using STEROID-LOEWETAPESTRY.

(iii) Parameters: $\{t_0 = 100, i_{max} = 500, N = 50, \epsilon_t = 5\%, \epsilon_a = 5\%, \lambda^+ = 1.8\}$

The solution set⁸ is listed in Table 7.11 and its characteristics is quite different from that of the MAPK-PI3K network. First, the solution set of the glucose-stimulated insulin secretion network contains fewer solutions. Second, the off-target effects of these solutions are generally relatively high (*e.g.*, greater than 1000). Third, for majority (90%) of the solutions, at least one target has activity that exceed 1000. This difference in result could be due to the

⁸For the glucose-stimulated insulin secretion network, the MCSA did not complete running within 8 days. Hence, we did not present the results.

presence of fewer targets that can effectively control the activity of mitochondrial ATP in the glucose-stimulated insulin secretion network. In particular, there were only two targets (4% of total targets) ⁹ that can effectively regulate the activity of mitochondrial ATP when perturbed individually. This is in contrast to the MAPK-PI3K network in which 64% of the targets can achieve the therapeutic goal when they are perturbed alone. Hence, in comparison to the MAPK-PI3K network, the LOEWE-based heuristic plays a limited role in the glucose-stimulated insulin secretion network, resulting in fewer solutions found, higher off-target effects and higher target activities in the combinations.

In Table 7.11, we note an over-representation of two targets that have activities lesser than 100. They are inhibitors of reactions 7 and 10 which correspond to inhibitors of glyceraldehyde 3-phosphate dehydrogenase and lactate dehydrogenase, respectively. We explore the possibility of combining these two targets to obtain combinations that have better target activity profile. Indeed, when reactions 7 and 10 are inhibited at target activity of 10 and 39.25, respectively, the desired therapeutic effect was achieved with off-target effects of 2628.29. Hence, the solution sets of STEROID are useful as a guide for potential targets and target activities to explore in creating new target combinations.

In addition, we perform literature curation to assess the biological relevance of the solutions in Table 7.11. A search in *PubMed* using the keywords “type 2 diabetes mellitus drug combinations” did not yield any relevant target combinations for the glucose-stimulated insulin secretion network. Hence, we perform further curation by specifically looking for relevant literature pertaining to the predicted combinations in Table 7.11. Although we did not find any supporting evidence that combinations 5 and 6 have been performed, successfully or otherwise, in experiments, their individual components have shown efficacy in diabetes [222, 243, 293]. Hence, these combinations warrant further investigation as potential target combinations. We

⁹The two individual targets are activator of reaction 40 and inhibitor of reaction 43 and they correspond to the activator of ATP/ADP carrier from the cytoplasm to the mitochondrial, and from the mitochondrial to the cytoplasm, respectively.

Figure 7.5: Effects of solution set size.

did not find any supporting evidence for the remaining combinations. However, this does not necessarily imply that these combinations are not useful for diabetes. Current research may be focussed on examining other diabetes-related pathway (*e.g.*, protein kinase C-activation pathway [260]). Hence, these predicted combinations may not have been explored yet. Additional research is necessary to confirm the relevance of these combinations.

Effect of Parameters on Solution Set. In this final set of experiments, we shall investigate the effects of various parameters on the solution set of STEROID-LOEWETAPESTRY. In the following, we list the parameters and their ranges that are studied in the MAPK-PI3K network:

- (i) solution set size N : $\{5, 10, 25, 40, 50\}$,
- (ii) initial temperature of simulated annealing t_0 : $\{5, 25, 50, 75, 100\}$,
- (iii) limits of iterations per cycle of simulated annealing i_{max} : $\{5, 100, 250, 350, 500\}$,
- (iv) adjustment factor for therapeutic effect ϵ_t : $\{5 \times 10^{-4}, 5 \times 10^{-3}, 0.05, 0.1, 0.5\}$,
- (v) adjustment factor for target interaction ϵ_a : $\{5 \times 10^{-4}, 5 \times 10^{-3}, 0.05, 0.1, 0.5\}$,
- (vi) selective pressure λ^+ : $\{1, 1.3, 1.5, 1.8, 2\}$, and
- (vii) combination size $\mathcal{S}$: $\{2, 3, 4, 5, 6\}$.

Figure 7.6: Effects of initial temperature of simulated annealing.

Figure 7.7: Effects of limits of iterations per cycle of simulated annealing.

First, we vary the solution set size N . We make the following observations from Figure 7.5. The whiskers and inter-quartile ranges of the off-target effects boxplot are observed to increase as N increases. This implies that for smaller N , majority of the solutions within the middle 50% of off-target effects have off-target effects values that are closer with each other as compared to those for larger N . This is expected as increased number of solutions invariably results in a higher likelihood of encountering combinations with off-target effects that vary widely. Despite the potential of variability in the off-target effects, the solution set is relatively robust as the whiskers of the boxplot (which contains the majority of the solutions) are relatively constant in the range [0–20] when N is greater than 25. In addition, the increase in N demanded longer execution time to complete analysis as more iterations are required to identify the larger number of solutions.

Then, we examine the effect of varying t_0 (Figure 7.6) and find that there is no significant changes in terms of execution time, actual solution set size and off-target effects.

Next, we investigate the effects of varying i_{max} . The effects on execution time and actual solution set size that are observed in Figure 7.7 are similar to that in Figure 7.5. This is be-

Figure 7.8: Effects of adjustment factor for therapeutic effect.

Figure 7.9: Solution space when adjustment factor for therapeutic effect is varied.

cause increasing i_{max} increases the maximum number of iterations for the simulated annealing, allowing a larger solution space to be explored. Hence, more execution time is needed and the actual solution set size is larger. The whiskers of the off-target effects boxplot remain relatively constant in the range [0–20]. This corresponds well with our previous observations in Figure 7.5. Compared to t_0 , the effect of varying i_{max} is more pronounced. This is because, in our design of the simulated annealing algorithm, we have set t_0 to be smaller than i_{max} . Note that t_0 and i_{max} control the outer- and inner-loop of the algorithm, respectively. For instance, when t_0 is set to 5, the maximum number of iterations is $5 \times 500 = 2500$, whereas when i_{max} is set to 5, the maximum number of iterations is reduced further to $100 \times 5 = 500$.

Then, we vary ϵ_t . We note an increase in the actual solution set size and corresponding decrease in execution time as ϵ_t is increased. As ϵ_t is increased, there is further relaxation of the condition for therapeutic effect and this allows more candidates to be accepted. This could also be the reason for an increase in the range of the whiskers of the off-target effects boxplot when ϵ_t increased from 5×10^{-4} to 5×10^{-3} as there is a large increase in solution set size from 7 to 50. Note that although increasing ϵ_t results in a dramatic decrease in execution

Figure 7.10: Effects of adjustment factor for target interaction.

Figure 7.11: Effects of selective pressure.

Figure 7.12: Effects of combination size.

time, this is at the expense of solutions moving further away from the desired therapeutic effect (as observed in Figure 7.9).

Next, we vary ϵ_a (Figure 7.10) and λ^+ (Figure 7.11) and find that there are no significant changes in terms of execution time, actual solution set size and off-target effects.

Finally, we investigate the effects of combination size S on the result. We vary S in the range of $\{2, 3, 4, 5, 6\}$. Tables 7.12 to 7.14 list the top-5 results for the entire range of S we tested whereas Figure 7.12 summarizes the effect of varying combination size on off-target effects, execution time and the actual solution set size. We make the following observations.

$S=2$

No.	Target [Activity]	Reaction in [143]	Description	ζ_{off}
1	Reaction 21 IN [2.248] Reaction 1 ^f IN [0.004]	ERKP→ERKPP R+HRG→RHRG	ERK kinase inhibitor tyrosine kinase inhibitor	1.02187
2	Reaction 22 ACT [2.216] Reaction 17 IN [0.009]	ERKPP→ERKP MEKP→MEKPP	ERK phosphatase activator MEK kinase inhibitor	1.03207
3	Reaction 22 ACT [2.121] Reaction 14 ACT [0.023]	ERKPP→ERKP Raf*→Raf	ERK phosphatase activator Raf inhibitor	1.08848
4	Reaction 20 ACT [1.791] Reaction 21 IN [1.746]	ERKP→ERK ERKP→ERKPP	ERK phosphatase activator ERK kinase inhibitor	1.25864
5	Reaction 19 IN [1.423] Reaction 17 IN [0.265]	ERK→ERKP MEKP→MEKPP	ERK kinase inhibitor MEK kinase inhibitor	1.84469

$S=3$

No.	Target [Activity]	Reaction in [143]	Description	ζ_{off}
1	Reaction 34 ^b IN [1444.745] Reaction 21 IN [0.610] Reaction 22 ACT [1.186]	internalization→RP ERKP→ERKPP ERKPP→ERKP	tyrosine kinase inhibitor ERK kinase inhibitor ERK phosphatase activator	1.03054
2	Reaction 21 IN [2.099] Reaction 19 IN [0.13054] Reaction 17 IN [0.035]	ERKP→ERKPP ERK→ERKP MEKP→MEKPP	ERK kinase inhibitor ERK kinase inhibitor MEK kinase inhibitor	1.13798
3	Reaction 21 IN [2.025] Reaction 20 ACT [0.820] Reaction 15 IN [0.150]	ERKP→ERKPP ERKP→ERK MEK→MEKP	ERK kinase inhibitor ERK phosphatase activator MEK kinase inhibitor	1.27025
4	Reaction 14 ACT [0.053] Reaction 19 IN [1.551] Reaction 3 ^b IN [8624.076]	Raf*→Raf ERK→ERKP RP→RHRG2	Raf inhibitor ERK kinase inhibitor tyrosine kinase activator	1.79488
5	Reaction 12 ACT [0.029] Reaction 22 ACT [1.622] Reaction 14 ACT [0.154]	RasGTP→RasGDP ERKPP→ERKP Raf*→Raf	RasGAP activator ERK phosphatase activator Raf inhibitor	1.90132

$S=4$

No.	Target [Activity]	Reaction in [143]	Description	ζ_{off}
1	Reaction 21 IN [1.711] Reaction 6 ^f IN [0.741] Reaction 8 ^b ACT [0.034] Reaction 10 IN [5927.404]	ERKP→ERKPP RShc→RShP ShGS+RP→RShGS ShP→Shc	ERK kinase inhibitor Shc kinase inhibitor promoter of ShGS and RP binding Shc phosphatase inhibitor	2.57488
2	Reaction 24 ^f ACT [9096.206] Reaction 7 ^b ACT [1.140] Reaction 22 ACT [1.148] Reaction 17 IN [0.155]	RPI3K→RPI3K* RShGS→RShP+GS ERKPP→ERKP MEKP→MEKPP	PI3K activator promoter of RShGS dissociation ERK phosphatase activator MEK kinase inhibitor	4.60078
3	Reaction 31 ACT [7516.839] Reaction 6 ^b ACT [1.291] Reaction 21 IN [1.323] Reaction 13 IN [0.954]	AktPIP→AktPIP3 RShP→RShc ERKP→ERKPP Raf→Raf*	PIP kinase activator Shc phosphatase activator ERK kinase inhibitor Raf inhibitor	5.85213
4	Reaction 31 IN [2943.773] Reaction 18 ACT [1.249] Reaction 29 ^f IN [3705.561] Reaction 21 IN [0.059]	AktPIP→AktPIP3 MEKPP→MEKP PIP3+Akt→AktPIP3 ERKP→ERKPP	PIP kinase inhibitor MEK phosphatase activator Akt inhibitor ERK kinase inhibitor	6.03896
5	Reaction 16 IN [3446.009] Reaction 14 ACT [0.671] Reaction 5 ^b ACT [1.047] Reaction 31 ACT [2110.600]	MEKP→MEK Raf*→Raf RShc→RP+Shc AktPIP→AktPIP3	MEK phosphatase inhibitor Raf inhibitor promoter of RShc dissociation PIP kinase activator	6.15524

Table 7.12: Top-5 results of STEROID-LOEWETAPESTRY for $S=\{2,3,4\}$.

$S=5$

No.	Target [Activity]	Reaction in [143]	Description	S_{off}
1	Reaction 21 IN [1.299] Reaction 20 IN [9766.701] Reaction 30 IN [5523.830] Reaction 33 IN [6767.722] Reaction 18 ACT [0.520]	ERKP→ERKPP ERKP→ERK AktPIP3→AktPIP AktPIPP→AktPIP MEKPP→MEKP	ERK kinase inhibitor ERK phosphatase inhibitor PIP3 phosphatase inhibitor PIP3 phosphatase inhibitor MEK phosphatase activator	4.88571
2	Reaction 28 ACT [7302.668] Reaction 21 IN [0.705] Reaction 19 IN [0.964] Reaction 33 ACT [2569.067] Reaction 3 ^b IN [4039.982]	PIP3→PI ERKP→ERKPP ERK→ERKP AktPIPP→AktPIP RP→RHRG2	PIP3 phosphatase activator ERK kinase inhibitor ERK kinase inhibitor PIP3 phosphatase activator tyrosine kinase activator	6.39667
3	Reaction 32 IN [7925.339] Reaction 12 ACT [0.040] Reaction 18 ACT [0.246] Reaction 9 ^b ACT [1393.884] Reaction 8 ^b ACT [0.765]	AktPIP→AktPIPP RasGTP→RasGDP MEKPP→MEKP GS+ShP→ShGS ShGS+RP→RShGS	PIP kinase inhibitor RasGAP activator MEK phosphatase activator promoter of GS and ShP binding promoter of ShGS and RP binding	6.48116
4	Reaction 12 ACT [0.237] Reaction 22 ACT [0.244] Reaction 24 ^b IN [1816.184] Reaction 29 ^f IN [9089.066] Reaction 31 ACT [7188.479]	RasGTP→RasGDP ERKPP→ERKP RPI3K*→RPI3K PIP3+Akt→AktPIP3 AktPIP→AktPIPP3	RasGAP activator ERK phosphatase activator PI3K activator Akt inhibitor PIP kinase activator	6.64748
5	Reaction 27 IN [7608.703] Reaction 9 ^b ACT [2710.060] Reaction 10 IN [3073.074] Reaction 13 IN [3.837] Reaction 21 IN [0.901]	PI→PIP3 GS+ShP→ShGS ShP→Shc Raf→Raf* ERKP→ERKPP	PI kinase inhibitor promoter of GS and ShP binding Shc phosphatase inhibitor Raf inhibitor ERK kinase inhibitor	6.74592

Table 7.13: Top-5 results of STEROID-LOEWETAPESTRY for $S=5$.

First, off-target effects increased as the combination size increased. Incorporating additional targets into the combination is likely to cause perturbation of a larger number of downstream nodes of these target. This inevitably increases off-target effects. Second, the increase in combination size also resulted in a slight increase in execution time. This is probably due to additional computation required for the selection of additional targets and their activities. Third, targets closer to the output node are frequently selected for the combinations. This is evident from the fact that all the listed combinations include either a ERK (or MEK) kinase inhibitor or a ERK (or MEK) phosphatase activator. Fourth, there is a tendency for less desirable targets (*i.e.*, promoter of pro-cancer signals) such as tyrosine kinase activator to be included in the combination as S increased. Hence, it may be necessary to consider additional rules such as exclusion of pro-cancer signal promoters when generating combinations of larger sizes.

$S=6$

No.	Target [Activity]	Reaction in [143]	Description	ζ_{off}
1	Reaction 25 ^f ACT [2019.677] Reaction 28 ACT [7230.289] Reaction 18 ACT [1.093] Reaction 27 ACT [8967.207] Reaction 24 ^b IN [4964.483] Reaction 1 ^f IN [1.128]	$RPI3K^* \rightarrow RP + PIP3K^*$ $PIP3 \rightarrow PI$ $MEKPP \rightarrow MEKP$ $PI \rightarrow PIP3$ $RPI3K^* \rightarrow RPI3K$ $R + HRG \rightarrow RHRG$	promoter of $RPI3K^*$ dissociation $PIP3$ phosphatase activator MEK phosphatase activator PI kinase activator $PI3K$ activator tyrosine kinase inhibitor	4.97259
2	Reaction 30 ACT [2380.633] $MKP3$ IN [5.477] Reaction 11 IN [13.181] Reaction 29 ^f IN [8314.225] Reaction 22 ACT [0.217] Reaction 16 ACT [0.013]	$AktPIP3 \rightarrow AktPIP$ $[MKP3]_0$ $RasGDP \rightarrow RasGTP$ $PIP3 + Akt \rightarrow AktPIP3$ $ERKPP \rightarrow ERKP$ $MEKP \rightarrow MEK$	$PIP3$ phosphatase activator $MKP3$ inhibitor $RasGEF$ inhibitor Akt inhibitor ERK phosphatase activator MEK phosphatase activator	5.60381
3	Reaction 25 ^b IN [2296.798] Reaction 12 ACT [0.222] Reaction 16 IN [1199.750] Reaction 3 ^f IN [0.114] Reaction 5 ^f IN [0.188] Reaction 2 ^b IN [2643.021]	$RP + PIP3K^* \rightarrow RPI3K^*$ $RasGTP \rightarrow RasGDP$ $MEKP \rightarrow MEK$ $RHRG2 \rightarrow RP$ $RP + Shc \rightarrow RShc$ $RHRG2 \rightarrow 2^*RHRG$	inhibitor of RP and $PI3K^*$ binding $RasGAP$ activator MEK phosphatase inhibitor tyrosine kinase inhibitor inhibitor of RP and Shc binding tyrosine kinase activator	5.62525
4	Reaction 34 ^b ACT [4766.160] Reaction 18 ACT [1.268] Reaction 33 ACT [5512.159] Reaction 11 IN [9.039] Reaction 27 IN [4232.607] Reaction 19 IN [0.011]	internalization $\rightarrow RP$ $MEKPP \rightarrow MEKP$ $AktPIP3 \rightarrow AktPIP$ $RasGDP \rightarrow RasGTP$ $PI \rightarrow PIP3$ $ERK \rightarrow ERKP$	tyrosine kinase activator MEK phosphatase activator $PIPP$ phosphatase activator $RasGEF$ inhibitor PI kinase inhibitor ERK kinase inhibitor	7.19757
5	Reaction 27 IN [3883.773] Reaction 11 IN [64.156] Reaction 22 ACT [1.177] Reaction 34 ^b ACT [7191.794] Reaction 29 ^b IN [7826.257] Reaction 8 ^f IN [0.152]	$PI \rightarrow PIP3$ $RasGDP \rightarrow RasGTP$ $ERKPP \rightarrow ERKP$ internalization $\rightarrow RP$ $AktPIP3 \rightarrow PIP3 + Akt$ $RShGS \rightarrow ShGS + RP$	PI kinase inhibitor $RasGEF$ inhibitor ERK phosphatase activator tyrosine kinase activator Akt activator inhibitor of $RShGS$ dissociation	8.52135

Table 7.14: Top-5 results of STERIOD-LOEWETAPESTRY for $S=6$.

7.7 Summary

In this chapter, we describe STERIOD, a heuristic-based simulated annealing approach for identifying target combinations. Specifically, we leverage on two heuristics, namely, target prioritization-based heuristic and LOEWE-based heuristic. The former heuristic is used for selecting appropriate targets when generating candidate target combinations in the simulated annealing algorithm whereas the latter heuristic is used to select target activity in order to ensure synergistic target interaction within the combinations. Our results reveal that the heuristics indeed improve execution time and off-target effects of the target combinations when compared

to the state-of-the-art approach that is MCSA-based. The solutions found using STERIOD are also enriched with targets in known drug combinations and found to be biologically relevant. In summary, our experiments demonstrate the potential value of knowledge-based heuristics for sampling and evaluating targets for the purpose of identifying good target combinations and STERIOD provides a first step in this regard.

Chapter 8

Conclusions and Future Work

In this chapter, we summarize the contributions of this thesis and discuss several open research topics related to our work.

8.1 Summary

In this thesis, we proposed a framework called TRECENTO (**T**arget **comb**Ina**T**ion **I**dentific**A**tion using a **N**etwork-based approach). TRECENTO uses a variety of data mining techniques to find target combinations for signaling networks. Specifically, the contributions of our work are as follows.

- (i) The lack of understanding of targets in signaling networks has hindered the development of tools for identifying putative targets in these networks. In this thesis, we propose TENET for characterizing targets of signaling networks based on topological features (Chapter 5). In particular, the TENET algorithm uses a SVM-based approach that affords flexibility to use signaling networks of different sizes and with different number of known targets as inputs. First, TENET identifies structural features that best predict known targets using feature selection approaches. Then, it generates prioritization models based on these features. We evaluated TENET on four signaling networks and found that it characterizes known targets better than state-of-the-art approaches. Importantly,

our experiments revealed that the predictive structural features are unique for different networks and single structural features are unlikely to yield good target prediction.

- (ii) Target prioritization is an important step in designing experiments for both basic research (*e.g.*, identification of gene regulators) and industrial application (*e.g.*, identification of drug targets). In this thesis, we propose TAPESTRY (Chapter 6), a network-based target prioritization approach. In particular, TAPESTRY uses the structural feature-based ranking obtained from prioritization models (generated by TENET) and integrates it with dynamic feature (PSSD)-based ranking using a weighted sum approach. A unique aspect of TAPESTRY is the learning phase that helps to identify appropriate structural features and weights to use for target prioritization. Note that TAPESTRY is different from TENET in that the latter performs characterization of targets based on structural features whereas the former performs target prioritization using insights gained from TENET. Unlike state-of-the-art approaches that ignore network dynamics, an important part of signaling networks, TAPESTRY uses both structural and dynamic features for target prioritization. Our experiments revealed that compared to state-of-the-art approaches, TAPESTRY is superior in terms of target prioritization results and has moderate runtime performance. Another important finding in our study is that the dynamic feature does have a role to play in target prioritization and can indeed improve the overall results.
- (iii) Combination therapy has often been touted as the silver bullet for diseases (*e.g.*, cancer) which implicate multiple pathways. However, the design of combination therapy is fraught with difficulties due in part to the diversity and complexity of signaling pathways which render exhaustive search of combination therapy impossible. In this thesis, we take a first step towards addressing the challenges of combination therapy design by proposing STERIOD, a rational *in silico* data mining driven approach for identifying target combinations (Chapter 7). In particular, STERIOD is designed to identify synergistic

target combinations which achieve the desired therapeutic effect and have minimized off-target effects by leveraging heuristics to reduce the vast search space prudently during simulated annealing. Specifically, the target prioritization-based heuristic uses target ranks generated by approaches such as TAPESTRY to select for more effective targets whereas the LOEWE-based heuristic is for selecting synergistic target activities. Compared to state-of-the-art approaches which are MCSA-based, STERIOD is more efficient in finding effective target combinations with smaller off-target effects which are biologically relevant.

8.2 Future Work

Our work in this thesis raises some open questions which have yet been addressed. These questions may point us to new research directions in combination therapy. In this section, we briefly discuss some key future research topics.

8.2.1 Network Uncertainty

In this thesis, TRECENTO uses signaling networks as inputs. An implicit assumption of TRECENTO is that signaling networks provided are correct. That is, there are no missing or incorrect nodes and edges in the networks. In actual fact, biological networks are often noisy [21]. Noise in the network affects quality of the results since it directly affects the structural and dynamic features of nodes. For instance, a potential target node which previously was found to have low degree centrality may in actual fact have high degree centrality due to missing edges. The inaccurate reflection of the node as low degree can affect its prioritization rank which may subsequently result in low probability of being selected as a target in the combination. A potentially viable approach is to consider signaling networks as probabilistic instead of deterministic [10, 199]. We intent to extend TRECENTO to support noisy signaling networks by exploring new data mining approaches for probabilistic networks in our future research.

8.2.2 Network Annotations

The vast amount of biological knowledge generated in recent years has created demands for organizing biological information. Towards this end, several semantic-based annotation tools [127, 179, 381] have been proposed. Signaling networks are also increasingly annotated [56, 173] with various information including gene function (*e.g.*, using Gene Ontology (GO) [13]), 3D structure (*e.g.*, using Entrez’s Molecular Modeling Database (MMDB) [57]) and even experimental data using heat maps [77]. Hence, network annotations can facilitate tasks such as validation of network accuracy, gaining new insights of targets and integration of experimental data from various sources. In particular, we propose to explore the use of network annotations for improving the quality of signaling networks that TRECENTO uses as inputs, for identifying novel target combinations and for investigating drug repositioning.

8.2.3 Network Dynamics

In Chapter 6, we investigated the use of a network dynamic feature for target prioritization and found that it can indeed improve overall target prioritization. In addition, our experiments reveal that this dynamic feature plays a minor role compared to the structural features we studied and the beneficial effect is also dependent on the signaling network studied. In our experiments, we explored a total of 16 structural and one dynamic features. Note that the lack of available dynamic features prevented us from performing an extensive study. The minor role of this dynamic feature could be due to it being an inappropriate choice (*e.g.*, lack of discriminative power to distinguish targets from non-targets) for the networks studied. Hence, a future research goal is to explore and identify new dynamic features for prioritizing targets. In addition, we plan to conduct further study on the complex interplay between network structure and dynamics, an important research area which holds the key for better understanding of signaling networks.

8.2.4 Integration with Experimental Data

Our work in this thesis focusses on *in silico* identification of good target combinations. Validation of TRECENTO's prediction *in vitro* or *in vivo* is the natural next step. The signaling networks that TRECENTO use as inputs although reflective of the biological system function, are likely to be different from the *in vitro* cell-based systems and *in vivo* small whole organism systems (*e.g.*, zebrafish) for several reasons. First, the scope of the *in silico* system is different from that of *in vitro* and *in vivo* systems. *In vitro* and *in vivo* systems contain all components of a cell and the whole organism, respectively. Hence, their scopes are generally wider than that of the *in silico* model, which typically focusses on a particular biological function. That is, the *in vitro* and *in vivo* systems usually contain more proteins and genes and interactions associated with them. Second, experimental results tend to suffer from lack of reproducibility [50] due in part to biological systems being complex and chaotic by nature. Although similar result trends may be observed from multiple replicate experiments, the results obtained are unlikely to be exactly the same. Third, the *in silico* model could be based on human cell line while that of the *in vivo* system is based on a non-human whole organism such as mouse or zebrafish. Organism-to-organism differences can result in differential response of the *in silico* and *in vivo* systems. Hence, it is important to extend TRECENTO to allow feedback of experimental results in order for TRECENTO to fine-tune its target combinations prediction. One key future research topic we shall pursue is the exploration of methods for integrating experimental results with network-based analysis.

8.2.5 Drug Combination

The next important step following TRECENTO's target combination prediction is drug combination prediction. Finding an appropriate drug combination requires us to answer several questions. How do we select an appropriate drug for a particular target? What is the appropri-

ate dosage for each drug? What is the appropriate dosing schedule for the drugs? How do we measure the drug effects?

Various key challenges exist and they have to be addressed in order to answer these questions. First, multiple drugs can act on the same target and the same drug can in turn act on multiple targets. For instance, both `BYL719` and `SAR245409` act on activated `PI3K`. In addition, `SAR245409` also targets activated `MTOR` [257]. The many-to-many relationship between drugs and targets, coupled with the complexity of signaling networks imply that drug effects are likely to be highly non-linear with dosages. In addition, the appropriate dosage of one drug may be influenced by interaction with another drug in the drug combination. For example, given a drug combination $\{\text{BYL719}, \text{SAR245409}\}$ which targets activated `PI3K` and `MTOR`, the appropriate dosage for `BYL719` would have to take into consideration `SAR245409`'s effect on activated `PI3K`. Second, interactions between drugs can affect the drug efficacy and toxicity profile. Some drugs interact with members of cytochrome P450 family, an important class of molecules that metabolizes many drugs. These drug interactions can affect the metabolism and hence the toxicity profiles of other drugs. For instance, Mibefradil, a potent inhibitor of `CYP3A4` was withdrawn from the market due to such adverse drug-drug interactions [263]. Third, the different clearance rate of drugs affect the dosing schedule. Choosing drugs with similar dosing schedule helps to keep the overall dosing schedule of the drug combination simple and improves patients' medication adherence. Poor medication adherence has been found to be linked to poor treatment outcome even though the medication may demonstrate efficacy in clinical trials [150].

Hence, identifying good drug combinations is a highly complex problem which demands data-mining tools that help scientists make decisions on drug combinations. In particular, for future research, we plan to develop data-mining approaches that integrate knowledge from various sources including drug databases (*e.g.*, *DrugBank* [385]) and publication databases (*e.g.*, *PubMed* [319]) in order to identify appropriate drugs for target combinations identified by `STERIOD`.

8.3 Conclusion

In this thesis, we presented TRECENTO, a network-based framework to identify target combinations of signaling networks *in silico*. First, TRECENTO characterizes known targets based on structural features and generates prioritization models from these structural features using TENET (Chapter 5), a SVM-based approach. Then, it performs target prioritization using TAPESTRY (Chapter 6). In particular, given a signaling network, TAPESTRY uses a network similarity approach to identify appropriate prioritization model for that network to generate structural feature-based ranks. These ranks are then combined with dynamic feature-based ranks using a weighted sum approach for prioritizing putative targets. Finally, synergistic target combinations are found using STERIOD (Chapter 7), a heuristics-based simulated annealing approach. Our research is carried out on several signaling networks, some of which are associated with diseases. In particular, it spans multiple research areas including machine learning, optimization and target prioritization and raises several open questions that point to potential research directions. Compared to state-of-the-art MCSA-based approaches, TRECENTO is more efficient in identifying biologically relevant target combinations that have smaller off-target effects. In summary, the work in this thesis is an important first step towards rational drug combination design.

References

- [1] J. Aach *et al.*, “Aligning gene expression time series with time warping algorithms.” *Bioinformatics*, vol. 17, no. 6, pp. 495–508, Jun 2001.
- [2] A. Adjei, “Blocking oncogenic ras signaling for cancer therapy,” *Journal of the National Cancer Institute*, vol. 93, no. 14, pp. 1062–1074, 2001.
- [3] D. Agus *et al.*, “Efficacy and safety of single-agent pertuzumab (rhumab 2c4), a human epidermal growth factor receptor dimerization inhibitor, in castration-resistant prostate cancer after progression from taxane-based therapy,” *Journal of clinical oncology*, vol. 25, no. 6, pp. 675–681, 2007.
- [4] R. Akbani *et al.*, “Applying support vector machines to imbalanced datasets,” in *Machine Learning: ECML 2004*. Springer, 2004, pp. 39–50.
- [5] R. Albert, “Scale-free networks in cell biology,” *Journal of Cell Science*, vol. 118, no. 21, pp. 4947–4957, 2005. [Online]. Available: <http://jcs.biologists.org/content/118/21/4947.abstract>
- [6] B. Aldridge *et al.*, “Physicochemical modelling of cell signalling pathways,” *Nature cell biology*, vol. 8, no. 11, pp. 1195–1203, 2006.
- [7] R. Alexander *et al.*, “Understanding modularity in molecular networks requires dynamics,” *Science signaling*, vol. 2, no. 81, p. pe44, 2009.

- [8] U. Alon, "Network motifs: theory and experimental approaches," *Nature Reviews Genetics*, vol. 8, no. 6, pp. 450–461, 2007.
- [9] S.-I. Amari *et al.*, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.
- [10] G. Antunes *et al.*, "A stochastic signaling network mediates the probabilistic induction of cerebellar long-term depression," *The Journal of Neuroscience*, vol. 32, no. 27, pp. 9288–9300, 2012.
- [11] R. Araujo *et al.*, "A mathematical model of combination therapy using the egfr signaling network," *Biosystems*, vol. 80, no. 1, pp. 57–69, 2005.
- [12] C. Arteaga, "The phosphatidylinositol-3 kinase/mtor pathway: new agents," *Breast Cancer Research*, vol. 13, no. Suppl 2, p. O8, 2011.
- [13] M. Ashburner *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.
- [14] A. Asthagiri *et al.*, "A computational study of feedback effects on signal dynamics in a mitogen-activated protein kinase (MAPK) pathway model," *Biotechnol Prog*, vol. 17, no. 2, pp. 227–239, 2001.
- [15] J. Avruch *et al.*, "Ras activation of the Raf kinase: Tyrosine kinase recruitment of the MAP kinase cascade," *Recent Prog Horm Res*, vol. 56, no. 1, pp. 127–156, 2001.
- [16] F. Ay *et al.*, "A fast and accurate algorithm for comparative analysis of metabolic pathways," *Journal of bioinformatics and computational biology*, vol. 7, no. 03, pp. 389–428, 2009.
- [17] J. Azuma *et al.*, "Mechanism of direct cardiostimulating actions of hydralazine." *Eur J Pharmacol*, vol. 135, no. 2, pp. 137–144, Mar 1987.

- [18] D. Bachovchin *et al.*, “Academic cross-fertilization by public screening yields a remarkable class of protein phosphatase methylesterase-1 inhibitors,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6811–6816, 2011.
- [19] C. Bambach, “Drawing and painting in the italian renaissance workshop,” *Theory and Practice*, vol. 1, pp. 300–1600, 1999.
- [20] J. Bangham, “Therapeutics: Cetuximab constricts conformational contortionist,” *Nature Reviews Cancer*, vol. 5, p. 421, June 2005.
- [21] N. Barkai *et al.*, “Variability and robustness in biomolecular systems,” *Molecular cell*, vol. 28, no. 5, pp. 755–760, 2007.
- [22] B. Barzel *et al.*, “Universality in network dynamics,” *Nature physics*, vol. 9, no. 10, pp. 673–681, 2013.
- [23] R. Bast, “Molecular approaches to personalizing management of ovarian cancer,” *Ann. Oncol.*, vol. 22, no. Suppl 8, pp. viii5–viii15, 2011. [Online]. Available: http://annonc.oxfordjournals.org/content/22/Suppl_8/viii5.abstract
- [24] M. Berlingerio *et al.*, “Netsimile: a scalable approach to size-independent network similarity,” *arXiv preprint arXiv:1209.2684*, 2012.
- [25] P. Bermolen *et al.*, “Support vector regression for link load prediction,” *Computer Networks*, vol. 53, no. 2, pp. 191–201, 2009.
- [26] U. Bhalla *et al.*, “Emergent properties of networks of biological signaling pathways,” *Science*, vol. 283, no. 5400, pp. 381–387, 01 1999.
- [27] A. Bhuyan *et al.*, “Resting membrane potential as a marker of apoptosis: studies on xenopus oocytes microinjected with cytochrome c,” *Cell death and differentiation*, vol. 8, no. 1, p. 63, 2001.

- [28] L. Biegler *et al.*, “Nonlinear parameter estimation: a case study comparison,” *AIChE Journal*, vol. 32, no. 1, pp. 29–45, 1986.
- [29] M. Blagosklonny, “Antagonistic drug combinations that select against drug resistance: from bacteria to cancer,” *Cancer biology & therapy*, vol. 6, no. 7, pp. 1013–1014, 2007.
- [30] J. Blake *et al.*, “The mouse genome database: integration of and access to knowledge about the laboratory mouse,” *Nucleic acids research*, p. gkt1225, 2013.
- [31] K. Bleicher *et al.*, “Hit and lead generation: beyond high-throughput screening,” *Nature Reviews Drug Discovery*, vol. 2, no. 5, pp. 369–378, 2003.
- [32] T. Blickle *et al.*, “A comparison of selection schemes used in evolutionary algorithms,” *Evol. Comput.*, vol. 4, no. 4, pp. 361–394, 1996.
- [33] H. Bock, *Recent advances in parameter identification techniques for ode*. Springer, 1983.
- [34] P. Bonacich *et al.*, “Eigenvector-like measures of centrality for asymmetric relations,” *Social Networks*, vol. 23, no. 3, pp. 191–201, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378873301000387>
- [35] E. Borenstein, “Computational systems biology and in silico modeling of the human microbiome,” *Briefings in bioinformatics*, vol. 13, no. 6, pp. 769–780, 2012.
- [36] A. Borisy *et al.*, “Systematic discovery of multicomponent therapeutics,” *Proc Natl Acad Sci U S A*, vol. 100, no. 13, pp. 7977–7982, Jun 2003.
- [37] B. Bornstein *et al.*, “Libsbml: an api library for sbml,” *Bioinformatics*, vol. 24, no. 6, pp. 880–881, 2008.

- [38] S. Brailsford *et al.*, “Constraint satisfaction problems: Algorithms and applications,” *Eur. J. Oper. Res.*, vol. 119, no. 3, pp. 557 – 581, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377221798003646>
- [39] U. Brandes, “A faster algorithm for betweenness centrality,” *Journal of Mathematical Sociology*, vol. 25, pp. 163–177, 2001.
- [40] M. Brehm *et al.*, “Humanized mouse models to study human diseases,” *Current opinion in endocrinology, diabetes, and obesity*, vol. 17, no. 2, p. 120, 2010.
- [41] ———, “Overcoming current limitations in humanized mouse research,” *Journal of Infectious Diseases*, vol. 208, no. suppl 2, pp. S125–S130, 2013.
- [42] A. Brodie, “Aromatase inhibitor development and hormone therapy: a perspective,” in *Seminars in oncology*, vol. 30. Elsevier, 2003, pp. 12–22.
- [43] R. Brough *et al.*, “Searching for synthetic lethality in cancer,” *Current opinion in genetics & development*, vol. 21, no. 1, pp. 34–41, 2011.
- [44] C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [45] H. Byun *et al.*, “Applications of support vector machines for pattern recognition: A survey,” in *Pattern recognition with support vector machines*. Springer, 2002, pp. 213–236.
- [46] D. Calzolari *et al.*, “Selective control of the apoptosis signaling network in heterogeneous cell populations,” *PLoS ONE*, vol. 2, no. 6, p. e547, 2007. [Online]. Available: <http://dx.plos.org/10.1371/journal.pone.0000547>
- [47] ———, “Search algorithms as a framework for the optimization of drug combinations,” *PLoS computational biology*, vol. 4, no. 12, p. e1000249, 2008.

- [48] P. Campbell *et al.*, “Tln-4601 suppresses growth and induces apoptosis of pancreatic carcinoma cells through inhibition of ras-erk mapk signaling,” *Journal of Molecular Signaling*, vol. 5, no. 1, p. 18, 2010. [Online]. Available: <http://www.jmolecularsignaling.com/content/5/1/18>
- [49] G. Carpenter, “ErbB-4: mechanism of action and biology,” *Experimental cell research*, vol. 284, no. 1, pp. 66–77, 2003.
- [50] A. Casadevall *et al.*, “Reproducible science,” *Infection and immunity*, vol. 78, no. 12, pp. 4972–4975, 2010.
- [51] C.-C. Chang *et al.*, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [52] D.-E. Chang *et al.*, “Acetate metabolism in a pta mutant of *Escherichia coli* w3110: Importance of maintaining acetyl coenzyme A flux for growth and survival,” *Journal of Bacteriology*, vol. 181, no. 21, pp. 6656–6663, 1999.
- [53] Y.-P. Chao *et al.*, “Alteration of growth yield by overexpression of phosphoenolpyruvate carboxylase and phosphoenolpyruvate carboxykinase in *Escherichia coli*,” *Applied and environmental microbiology*, vol. 59, no. 12, pp. 4261–4265, 1993.
- [54] O. Chapelle *et al.*, “Choosing multiple parameters for support vector machines,” *Machine learning*, vol. 46, no. 1-3, pp. 131–159, 2002.
- [55] G. Chaurasia *et al.*, “UniHi: an entry gate to the human protein interactome,” *Nucleic acids research*, vol. 35, no. suppl 1, pp. D590–D594, 2007.
- [56] H. Chen *et al.*, “Content-rich biological network constructed by mining pubmed abstracts,” *BMC bioinformatics*, vol. 5, no. 1, p. 147, 2004.

- [57] J. Chen *et al.*, “Mmdb: Entrez’s 3d-structure database,” *Nucleic Acids Research*, vol. 31, no. 1, pp. 474–477, 2003.
- [58] —, “Disease candidate gene identification and prioritization using protein interaction networks,” *BMC bioinformatics*, vol. 10, no. 1, p. 73, 2009.
- [59] L. Chen *et al.*, “Stack-based algorithms for pattern matching on DAGs,” in *VLDB ’05: Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 493–504.
- [60] Y.-A. Chen *et al.*, “Targetmine, an integrated data warehouse for candidate gene prioritisation and target discovery,” *PLoS One*, vol. 6, no. 3, p. e17844, 2011.
- [61] W. Chiou, “Critical evaluation of the potential error in pharmacokinetic studies of using the linear trapezoidal rule method for the calculation of the area under the plasma level-time curve,” *J. Pharmacokinet. Pharmacodynamics*, vol. 6, pp. 539–546, 1978.
[Online]. Available: <http://dx.doi.org/10.1007/BF01062108>
- [62] T.-C. Chou, “Theoretical basis, experimental design, and computerized simulation of synergism and antagonism in drug combination studies,” *Pharmacological reviews*, vol. 58, no. 3, pp. 621–681, 2006.
- [63] T. Chou *et al.*, “A simple generalized equation for the analysis of multiple inhibitions of michaelis-menten kinetic systems,” *J. Biol. Chem.*, vol. 252, no. 18, pp. 6438–42, 1977.
[Online]. Available: <http://www.jbc.org/content/252/18/6438.abstract>
- [64] —, “Compusyn for drug combinations: Pc software and users guide: a computer program for quantitation of synergism and antagonism in drug combinations, and the determination of ic50 and ed50 and ld50 values,” *Paramus, NJ: ComboSyn*, 2005.

- [65] G. Chu, “Cellular responses to cisplatin. The roles of DNA-binding proteins and DNA repair.” *Journal of Biological Chemistry*, vol. 269, no. 2, pp. 787–790, 1994. [Online]. Available: <http://www.jbc.org/content/269/2/787.abstract>
- [66] H. Chua *et al.*, “One feature doesn’t fit all: Characterizing topological features of targets in signaling networks. [www.cais.ntu.edu.sg/~assourav/TechReports/ DIFFER-TR.pdf](http://www.cais.ntu.edu.sg/~assourav/TechReports/DIFFER-TR.pdf), 2014.”
- [67] —, “STEROID:in silico heuristic target combination identification for disease-related signaling networks. www.cais.ntu.edu.sg/~assourav/TechReports/STEROID-TR.pdf,2012.”
- [68] —, “PANI:a novel algorithm for fast discovery of putative target nodes in signaling networks,” *ACM BCB*, 2011.
- [69] —, “In silico identification of endo16 regulators in the sea urchin endomesoderm gene regulatory network.” in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, ser. IHI ’12, 2012.
- [70] M. Cloutier *et al.*, “Dynamic modeling and analysis of cancer cellular network motifs,” *Integrative Biology*, vol. 3, no. 7, pp. 724–732, 2011.
- [71] R. Collobert *et al.*, “Svmtorch: Support vector machines for large-scale regression problems,” *The Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.
- [72] V. Conitzer *et al.*, “Improved bounds for computing kemeny rankings,” in *AAAI*, vol. 6, 2006, pp. 620–626.
- [73] A. Copeland, “A reasonable social welfare function. university of michigan seminar on applications of mathematics to the social sciences. university of michigan,” *Ann Arbor*, 1951.

- [74] R. Copeland *et al.*, “Drug-target residence time and its implications for lead optimization.” *Nat Rev Drug Discov*, vol. 5, no. 9, pp. 730–739, Sep 2006. [Online]. Available: <http://dx.doi.org/10.1038/nrd2082>
- [75] I. Cortes-Ciriano *et al.*, “Experimental validation of in silico target predictions on synergistic protein targets.” *J. Cheminformatics*, vol. 5, no. S-1, p. P31, 2013.
- [76] K. Courtney *et al.*, “The pi3k pathway as drug target in human cancer,” *Journal of Clinical Oncology*, vol. 28, no. 6, pp. 1075–1083, 2010.
- [77] M. Creamer and otherse, “Specification, annotation, visualization and simulation of a large rule-based model for erbb receptor signaling,” *BMC systems biology*, vol. 6, no. 1, p. 107, 2012.
- [78] G. Crowther *et al.*, “Identification of attractive drug targets in neglected-disease pathogens using an in silico approach,” *PLoS neglected tropical diseases*, vol. 4, no. 8, p. e804, 2010.
- [79] P. Csermely *et al.*, “The efficiency of multi-target drugs: the network approach might help drug design.” *Trends Pharmacol Sci*, vol. 26, no. 4, pp. 178–182, Apr 2005.
- [80] ———, “Structure and dynamics of molecular networks: A novel paradigm of drug discovery: A comprehensive review,” *Pharmacology & therapeutics*, 2013.
- [81] C. Cunningham *et al.*, “A phase i trial of c-raf kinase antisense oligonucleotide isis 5132 administered as a continuous intravenous infusion in patients with advanced cancer.” *Clin Cancer Res*, vol. 6, no. 5, pp. 1626–1631, May 2000.
- [82] B. Davies *et al.*, “AZD6244 (ARRY-142886), a potent inhibitor of mitogen-activated protein kinase/extracellular signal-regulated kinase kinase 1/2 kinases: mechanism of action in vivo, pharmacokinetic/pharmacodynamic relationship, and potential for

- p>combination in preclinical models,”
- Mol. Cancer Ther.*
- , vol. 6, no. 8, pp. 2209–2219, 2007. [Online]. Available:
- <http://mct.aacrjournals.org/content/6/8/2209.abstract>
- [83] J. Davis *et al.*, “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.
- [84] J. Dawson *et al.*, “Quantitative phenotypic and pathway profiling guides rational drug combination strategies,” *Frontiers in Pharmacology*, vol. 5, 2014.
- [85] J. de Borda, “Mémoire sur les élections au scrutin,” 1781.
- [86] R. de Groot *et al.*, “Middle east respiratory syndrome coronavirus (mers-cov): announcement of the coronavirus study group,” *Journal of virology*, vol. 87, no. 14, pp. 7790–7792, 2013.
- [87] M. Debono *et al.*, “Antibiotics that inhibit fungal cell wall development,” *Annual Reviews in Microbiology*, vol. 48, no. 1, pp. 471–497, 1994.
- [88] D. Deeb *et al.*, “Growth inhibitory and apoptosis-inducing effects of xanthohumol, a prenylated chalone present in hops, in human prostate cancer cells,” *Anticancer research*, vol. 30, no. 9, pp. 3333–3339, 2010.
- [89] D. Desai *et al.*, “Formulation design, challenges, and development considerations for fixed dose combination (fdc) of oral solid dosage forms,” *Pharmaceutical development and technology*, vol. 18, no. 6, pp. 1265–1276, 2013.
- [90] E. Deu *et al.*, “Functional studies of plasmodium falciparum dipeptidyl aminopeptidase i using small molecule inhibitors and active site probes,” *Chemistry & biology*, vol. 17, no. 8, pp. 808–819, 2010.

- [91] P. Diaconis *et al.*, “Spearman’s footrule as a measure of disarray,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 262–268, 1977.
- [92] E. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [93] B. Ding *et al.*, “Fast set intersection in memory,” *Proceedings of the VLDB Endowment*, vol. 4, no. 4, pp. 255–266, 2011.
- [94] P. Domingos, “Metacost: A general method for making classifiers cost-sensitive,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 155–164.
- [95] C. Dominguez *et al.*, “Haddock: a protein-protein docking approach based on biochemical or biophysical information,” *Journal of the American Chemical Society*, vol. 125, no. 7, pp. 1731–1737, 2003.
- [96] M. Doyle *et al.*, “Drug target prediction and prioritization: using orthology to predict essentiality in parasite genomes,” *BMC Genomics*, vol. 11, no. 1, p. 222, 2010.
- [97] W. Duch *et al.*, “Feature selection based on information theory filters,” in *Neural Networks and Soft Computing*. Springer, 2003, pp. 173–178.
- [98] C. Dwork *et al.*, “Rank aggregation methods for the web,” in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 613–622.
- [99] T. Eitrich *et al.*, “Classification of highly unbalanced cyp450 data of drugs using cost sensitive machine learning techniques,” *Journal of chemical information and modeling*, vol. 47, no. 1, pp. 92–103, 2007.
- [100] N. Emanuele *et al.*, “Consequences of alcohol use in diabetics,” *Alcohol Health and Research World*, vol. 22, pp. 211–219, 1998.

- [101] D. Emig *et al.*, “Drug target prediction and repositioning using an integrated network-based approach,” *PLoS ONE*, vol. 8, no. 4, p. e60618, 04 2013. [Online]. Available: <http://dx.doi.org/10.1371/journal.pone.0060618>
- [102] J. Engelfriet *et al.*, “A comparison of boundary graph grammars and context-free hypergraph grammars,” *Information and Computation*, vol. 84, no. 2, pp. 163 – 206, 1990.
- [103] M. Erion *et al.*, “Mb06322 (cs-917): a potent and selective inhibitor of fructose 1, 6-bisphosphatase for controlling gluconeogenesis in type 2 diabetes,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 22, pp. 7970–7975, 2005.
- [104] I. Espinosa *et al.*, “Tagatose: from a sweetener to a new diabetic medication?” *Expert opinion on investigational drugs*, vol. 19, no. 2, pp. 285–294, 2010.
- [105] N. Eungdamrong *et al.*, “Compartment-specific feedback loop and regulated trafficking can result in sustained activation of ras at the golgi,” *Biophysical Journal*, vol. 92, no. 3, pp. 808–815, 02 2007.
- [106] P. Evangelista *et al.*, “Taming the curse of dimensionality in kernels and novelty detection,” in *Applied soft computing technologies: The challenge of complexity*. Springer, 2006, pp. 425–438.
- [107] G. Fagiolo, “Clustering in complex directed networks,” *Phys. Rev. E*, vol. 76, pp. 026 107–026 114, Aug 2007. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevE.76.026107>
- [108] W. Farmer *et al.*, “Reduction of aerobic acetate production by escherichia coli.” *Applied and Environmental microbiology*, vol. 63, no. 8, pp. 3205–3210, 1997.
- [109] B. Farrell *et al.*, “Managing clinical trials,” *Trials*, vol. 11, no. 1, p. 78, 2010.

- [110] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, pp. 1–38, 2004.
- [111] A. Fielding *et al.*, "A review of methods for the assessment of prediction errors in conservation presence/absence models," *Environmental conservation*, vol. 24, no. 01, pp. 38–49, 1997.
- [112] P. Fishburn, "Letter to the editoradditive utilities with incomplete product sets: Application to priorities and assignments," *Operations Research*, vol. 15, no. 3, pp. 537–542, 1967.
- [113] J. Fitzgerald *et al.*, "Systems biology and combination therapy in the quest for clinical efficacy," *Nat Chem Biol*, vol. 2, no. 9, pp. 458–466, Sep 2006.
- [114] E. Floyd *et al.*, "Development and use of biomarkers in oncology drug development," *Toxicologic pathology*, vol. 32, no. 1 suppl, pp. 106–115, 2004.
- [115] J. Fontecilla-Camps *et al.*, "Structure–function relationships of anaerobic gas-processing metalloenzymes," *Nature*, vol. 460, no. 7257, pp. 814–822, 2009.
- [116] Food, D. Administration, U. D. of Health, H. Services *et al.*, "Innovation or stagnation: Challenge and opportunity on the critical path to new medical products, 31pp," 2004.
- [117] M. Fredman *et al.*, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM (JACM)*, vol. 34, no. 3, pp. 596–615, 1987.
- [118] L. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.
- [119] C. Fremin *et al.*, "From basic research to clinical development of mek1/2 inhibitors for cancer therapy," *J. Hematol. Oncol.*, vol. 3, no. 1, p. 8, 2010. [Online]. Available: <http://www.jhonline.org/content/3/1/8>

- [120] E. Freuder *et al.*, “Systematic versus stochastic constraint satisfaction,” in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, ser. IJCAI’95, San Francisco, CA, USA, 1995, pp. 2027–2032. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1643031.1643160>
- [121] N. Friedman *et al.*, “Bayesian network classifiers,” *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.
- [122] T. Furey *et al.*, “Support vector machine classification and validation of cancer tissue samples using microarray expression data,” *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.
- [123] G. Fussmann *et al.*, “Crossing the hopf bifurcation in a live predator-prey system,” *Science*, vol. 290, no. 5495, pp. 1358–1360, 2000.
- [124] C. Gaibisso, “A partially persistent data structure for the set-union problem with backtracking,” in *Machines, Languages, and Complexity*. Springer, 1989, pp. 128–135.
- [125] M. Garnett *et al.*, “Systematic identification of genomic markers of drug sensitivity in cancer cells,” *Nature*, vol. 483, no. 7391, pp. 570–575, 2012.
- [126] E. Gilchrist *et al.*, “Reverse genetics techniques: engineering loss and gain of gene function in plants,” *Briefings in functional genomics*, p. elp059, 2010.
- [127] C. Gille *et al.*, “Metannogen: annotation of biological reaction networks,” *Bioinformatics*, vol. 27, no. 19, pp. 2763–2764, 2011.
- [128] M. Girvan *et al.*, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [129] S. Glaysher *et al.*, “Targeting egfr and pi3k pathways in ovarian cancer,” *British journal of cancer*, vol. 109, no. 7, pp. 1786–1794, 2013.

- [130] W. Goffe *et al.*, “Global optimization of statistical functions with simulated annealing,” *Journal of Econometrics*, vol. 60, no. 1, pp. 65–99, 1994.
- [131] J. Gollob *et al.*, “Role of raf kinase in cancer: therapeutic potential of targeting the raf/mek/erk signal transduction pathway.” *Semin Oncol*, vol. 33, no. 4, pp. 392–406, Aug 2006.
- [132] T. Gonzalez *et al.*, “An efficient algorithm for the kolmogorov-smirnov and lilliefors tests,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 1, pp. 60–64, 1977.
- [133] S. Gordo *et al.*, “Knitting and untying the protein network: modulation of protein ensembles as a therapeutic strategy.” *Protein Sci*, vol. 18, no. 3, pp. 481–493, Mar 2009. [Online]. Available: <http://dx.doi.org/10.1002/pro.43>
- [134] K. Gotink *et al.*, “Anti-angiogenic tyrosine kinase inhibitors: what is their mechanism of action?” *Angiogenesis*, vol. 13, no. 1, pp. 1–14, Mar 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10456-009-9160-6>
- [135] S. Grant *et al.*, “Roles of erbb family receptor tyrosine kinases, and downstream signaling pathways, in the control of cell growth and survival.” *Frontiers in bioscience: a journal and virtual library*, vol. 7, p. d376, 2002.
- [136] S. Gunn, “Support vector machines for classification and regression,” *ISIS technical report*, vol. 14, 1998.
- [137] A. Gupta *et al.*, “Compliance, safety, and effectiveness of fixed-dose combinations of antihypertensive agents a meta-analysis,” *Hypertension*, vol. 55, no. 2, pp. 399–407, 2010.

- [138] Ö. Gürcan, “An emergent model for mimicking human neuronal pathways in silico,” in *Advances in Artificial Life, ECAL*, vol. 12, 2013, pp. 1172–1173.
- [139] P. Gustafson *et al.*, “Local sensitivity analysis,” *Bayesian statistics*, vol. 5, pp. 197–210, 1996.
- [140] M. A. Hall, “Correlation-based feature selection for machine learning,” Ph.D. dissertation, The University of Waikato, 1999.
- [141] J. Harrold *et al.*, “Network-based approaches in drug discovery and early development,” *Clinical Pharmacology & Therapeutics*, vol. 94, no. 6, pp. 651–658, 2013.
- [142] S. Hasan *et al.*, “Prioritizing genomic drug targets in pathogens: application to mycobacterium tuberculosis,” *PLoS Computational Biology*, vol. 2, no. 6, p. e61, 2006.
- [143] M. Hatakeyama *et al.*, “A computational model on the modulation of mitogen-activated protein kinase (MAPK) and Akt pathways in heregulin-induced ErbB signalling,” *Biochem J*, vol. 373, no. Pt 2, pp. 451–463, Jul 2003.
- [144] P. Hawkins *et al.*, “Conformer generation with omega: algorithm and validation using high quality structures from the protein databank and cambridge structural database,” *Journal of chemical information and modeling*, vol. 50, no. 4, pp. 572–584, 2010.
- [145] C. H.E. *et al.*, “PANI: an interactive data-driven tool for target prioritization in signaling networks,” in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, ser. IHI '12. New York, NY, USA: ACM, 2012, pp. 851–854. [Online]. Available: <http://doi.acm.org/10.1145/2110363.2110471>
- [146] X. He *et al.*, “Why do hubs tend to be essential in protein networks?” *PLoS Genetics*, vol. 2, no. 6, p. e88, 06 2006.

- [147] D. Henderson *et al.*, “The theory and practice of simulated annealing,” in *Handbook of metaheuristics*. Springer, 2003, pp. 287–319.
- [148] M. Heo *et al.*, “Protein abundances and interactions coevolve to promote functional complexes while suppressing non-specific binding,” *arXiv preprint arXiv:1007.2668*, 2010.
- [149] R. Herbrich *et al.*, “Support vector learning for ordinal regression,” 1999.
- [150] P. Ho *et al.*, “Medication adherence its importance in cardiovascular outcomes,” *Circulation*, vol. 119, no. 23, pp. 3028–3035, 2009.
- [151] U. Hofer *et al.*, “Pre-clinical modeling of ccr5 knockout in human hematopoietic stem cells by zinc finger nucleases using humanized mice,” *Journal of Infectious Diseases*, vol. 208, no. suppl 2, pp. S160–S164, 2013.
- [152] S. Hoops *et al.*, “Copasia complex pathway simulator,” *Bioinformatics*, vol. 22, no. 24, pp. 3067–3074, 2006.
- [153] D. Hosmer Jr. *et al.*, *Applied Logistic Regression. Second Edition*. John Wiley & Sons, 2004.
- [154] A. Howes *et al.*, “The phosphatidylinositol 3-kinase inhibitor, px-866, is a potent inhibitor of cancer cell motility and growth in three-dimensional cultures,” *Molecular cancer therapeutics*, vol. 6, no. 9, pp. 2505–2514, 2007.
- [155] C.-W. Hsu *et al.*, “A practical guide to support vector classification,” 2003.
- [156] D. Hu *et al.*, “Time-dependent sensitivity analysis of biological networks: Coupled MAPK and PI3K signal transduction pathways,” *The Journal of Physical Chemistry A*, vol. 110, no. 16, pp. 5361–5370, 2006.

- [157] E. Hu *et al.*, “White rice consumption and risk of type 2 diabetes: meta-analysis and systematic review,” *BMJ: British Medical Journal*, vol. 344, 2012.
- [158] J. Hua *et al.*, “Optimal number of features as a function of sample size for various classification rules,” *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, 2005.
- [159] S. Hua *et al.*, “Support vector machine approach for protein subcellular localization prediction,” *Bioinformatics*, vol. 17, no. 8, pp. 721–728, 2001.
- [160] W. Hwang *et al.*, “Bridging centrality: graph mining from element level to group level,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 336–344.
- [161] S. Iadevaia *et al.*, “Identification of optimal drug combinations targeting cellular networks: Integrating phospho-proteomics and computational network analysis,” *Cancer Res.*, vol. 70, no. 17, pp. 6704–6714, 2010.
- [162] R. Inc. (2013) Rxlist – the internet drug index. Accessed 3 Sept 2013. [Online]. Available: <http://www.rxlist.com>
- [163] J. Infante *et al.*, “Safety and efficacy results from the first-in-human study of the oral mek 1/2 inhibitor gsk1120212,” *J Clin Oncol*, vol. 28, no. 15, p. 2503, 2010.
- [164] R. Ito *et al.*, “Current advances in humanized mouse models,” *Cellular & molecular immunology*, vol. 9, no. 3, pp. 208–214, 2012.
- [165] A. Ivanova *et al.*, “A non-parametric approach to the design and analysis of two-dimensional dose-finding trials,” *Statistics in medicine*, vol. 23, no. 12, pp. 1861–1870, 2004.
- [166] P. Jaccard, “The distribution of the flora in the alpine zone. 1,” *New phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

- [167] S. Jahn *et al.*, “A role for eiantr in controlling fluxes in the central metabolism of e. coli k12,” *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, vol. 1833, no. 12, pp. 2879 – 2889, 2013.
- [168] N. Jiang *et al.*, “A kinetic core model of the glucose-stimulated insulin secretion network of pancreatic cells,” *Mammalian Genome*, vol. 18, no. 6-7, pp. 508–520, 2007. [Online]. Available: <http://dx.doi.org/10.1007/s00335-007-9011-y>
- [169] C. S. Johnston and C. A. Gaas, “Vinegar: medicinal uses and antiglycemic effect,” *Medscape General Medicine*, vol. 8, no. 2, p. 61, 2006.
- [170] M. Kabir *et al.*, “Metabolic regulation analysis of icd-gene knockout escherichia coli based on 2d electrophoresis with maldi-tof mass spectrometry and enzyme activity measurements,” *Applied microbiology and biotechnology*, vol. 65, no. 1, pp. 84–96, 2004.
- [171] T. Kacprowski *et al.*, “Networkprioritizer: a versatile tool for network-based prioritization of candidate disease genes or other molecules,” *Bioinformatics*, vol. 29, no. 11, pp. 1471–1473, 2013.
- [172] R. Kaminski *et al.*, “Minimal intervention strategies in logical signaling networks with asp,” *Theory and Practice of Logic Programming*, vol. 13, no. 4-5, pp. 675–690, 2013.
- [173] K. Kandasamy *et al.*, “Netpath: a public resource of curated signal transduction pathways,” *Genome biology*, vol. 11, no. 1, p. R3, 2010.
- [174] U. Kang *et al.*, “Centralities in large networks: Algorithms and observations.” in *SDM*, vol. 2011. SIAM, 2011, pp. 119–130.
- [175] N. Kanthamneni *et al.*, “Nanoparticulate delivery of novel drug combination regimens for the chemoprevention of colon cancer,” *International journal of oncology*, vol. 37, no. 1, p. 177, 2010.

- [176] J. Kemeny, "Mathematics without numbers," *Daedalus*, vol. 88, no. 4, pp. 577–591, 1959.
- [177] E. Keogh *et al.*, "Derivative dynamic time warping," in *In First SIAM International Conference on Data Mining (SDM2001)*, 2001.
- [178] I. Kim *et al.*, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural and multidisciplinary optimization*, vol. 29, no. 2, pp. 149–158, 2005.
- [179] J. Kim *et al.*, "Genia corpora semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, no. suppl 1, pp. i180–i182, 2003.
- [180] Y. Kim *et al.*, "Substrate-dependent control of mapk phosphorylation in vivo." *Mol Syst Biol*, vol. 7, p. 467, Feb 2011. [Online]. Available: <http://dx.doi.org/10.1038/msb.2010.121>
- [181] C. Kiss and M. Bichler, "Identification of influencersmeasuring influence in customer networks," *Decision Support Systems*, vol. 46, no. 1, pp. 233–253, 2008.
- [182] S. Klamt *et al.*, "Hypergraphs and cellular networks," *PLoS Comput Biol*, vol. 5, no. 5, p. e1000385, May 2009.
- [183] A. Klip *et al.*, "Cellular mechanism of action of metformin," *Diabetes Care*, vol. 13, no. 6, pp. 696–704, 1990.
- [184] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, no. 2, 1995, pp. 1137–1145.
- [185] C. Kohl *et al.*, "Effects of benfluorex on fatty acid and glucose metabolism in isolated rat hepatocytes: From metabolic fluxes to gene expression," *Diabetes*, vol. 51, no. 8,

- pp. 2363–2368, 2002. [Online]. Available: <http://diabetes.diabetesjournals.org/content/51/8/2363.abstract>
- [186] S. Köhler *et al.*, “Walking the interactome for prioritization of candidate disease genes,” *The American Journal of Human Genetics*, vol. 82, no. 4, pp. 949–958, 2008.
- [187] A. Kołcz *et al.*, “Svm-based filtering of emmail spam with contentmspecific misclassification costs,” 2001.
- [188] S. Kondapaka *et al.*, “Perifosine, a novel alkylphospholipid, inhibits protein kinase b activation.” *Mol Cancer Ther*, vol. 2, no. 11, pp. 1093–1103, Nov 2003.
- [189] D. Koschützki *et al.*, “Centrality analysis methods for biological networks and their application to gene regulatory networks,” *Gene regulation and systems biology*, vol. 2, p. 193, 2008.
- [190] D. Koschützki *et al.*, “Centrality analysis methods for biological networks and their application to gene regulatory networks,” *Gene Regul Syst Bio*, vol. 2, pp. 193–201, 2008.
- [191] M. Kotlyar *et al.*, “Network-based characterization of drug-regulated genes, drug targets, and toxicity,” *Methods*, vol. 57, no. 4, pp. 499–507, 2012.
- [192] O. Kotte *et al.*, “Bacterial adaptation through distributed sensing of metabolic fluxes,” *Molecular systems biology*, vol. 6, no. 1, 2010.
- [193] J. Kotz, “Phenotypic screening, take two,” *SciBX: Science-Business eXchange*, vol. 5, no. 15, 2012.
- [194] C. Kreutz *et al.*, “An error model for protein quantification.” *Bioinformatics*, vol. 23, no. 20, pp. 2747–2753, Oct 2007.

- [195] O. Kuchaiev *et al.*, “Topological network alignment uncovers biological function and phylogeny,” *Journal of the Royal Society Interface*, vol. 7, no. 50, pp. 1341–1354, 2010.
- [196] C. Kuhn *et al.*, “Monte carlo analysis of an ode model of the sea urchin endomesoderm network,” *BMC Systems Biology*, vol. 3, no. 1, p. 83, 2009. [Online]. Available: <http://www.biomedcentral.com/1752-0509/3/83>
- [197] S. KwanáSze *et al.*, “Proteome profiling reveals potential cellular targets of staurosporine using a clickable cell-permeable probe,” *Chemical Communications*, vol. 47, no. 40, pp. 11 306–11 308, 2011.
- [198] Y.-K. Kwon *et al.*, “Coherent coupling of feedback loops: a design principle of cell signaling networks,” *Bioinformatics*, vol. 24, no. 17, pp. 1926–1932, Sep 2008.
- [199] J. Ladbury *et al.*, “Noise in cellular signaling pathways: causes and effects,” *Trends in biochemical sciences*, vol. 37, no. 5, pp. 173–178, 2012.
- [200] C. Lavigne *et al.*, “Prevention of skeletal muscle insulin resistance by dietary cod protein in high fat-fed rats,” *American Journal of Physiology-Endocrinology And Metabolism*, vol. 281, no. 1, pp. E62–E71, 2001.
- [201] N. Le Novère *et al.*, “Biomodels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems,” *Nucleic Acids Res*, vol. 34, no. Database issue, pp. D689–D691, Jan 2006.
- [202] G. Lebedeva *et al.*, “Model-based global sensitivity analysis as applied to identification of anti-cancer drug targets and biomarkers of drug resistance in the erbb2/3 network,” *European journal of pharmaceutical sciences*, vol. 46, no. 4, pp. 244–258, 2012.
- [203] J. Lee *et al.*, “Target deconvolution techniques in modern phenotypic profiling,” *Current opinion in chemical biology*, vol. 17, no. 1, pp. 118–126, 2013.

- [204] I. Leong *et al.*, “In vivo testing of microrna-mediated gene knockdown in zebrafish,” *BioMed Research International*, vol. 2012, 2012.
- [205] H. Li *et al.*, “Effects of raf kinase inhibitor protein expression on metastasis and progression of human epithelial ovarian cancer,” *Molecular Cancer Research*, vol. 6, no. 6, pp. 917–928, 2008.
- [206] P. Li *et al.*, “Characterization of regulatory features of housekeeping and tissue-specific regulators within tissue regulatory networks,” *BMC systems biology*, vol. 7, no. 1, p. 112, 2013.
- [207] T. Li *et al.*, “Performance of batch membrane reactor: Glycerol-3-phosphate synthesis coupled with adenosine triphosphate regeneration,” *Biotechnology and Bioengineering*, vol. 74, no. 4, pp. 326–334, 2001. [Online]. Available: <http://dx.doi.org/10.1002/bit.1123>
- [208] Z. Li *et al.*, “Drug target identification based on flux balance analysis of metabolic networks,” in *International Conference on Computational Systems Biology*, 2010, pp. 331–338.
- [209] D. Lidke *et al.*, “Erk nuclear translocation is dimerization-independent but controlled by the rate of phosphorylation,” *Journal of biological chemistry*, vol. 285, no. 5, pp. 3092–3102, 2010.
- [210] X. Lin *et al.*, “A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information,” *Journal of Chromatography B*, vol. 910, pp. 149–155, 2012.
- [211] C. Ling *et al.*, “Class imbalance problem,” in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 171–171.

- [212] X.-Y. Liu *et al.*, “Docking and molecular dynamics simulations of peroxisome proliferator activated receptors interacting with pan agonist sodelglitazar,” *Protein and peptide letters*, vol. 18, no. 10, pp. 1021–1027, 2011.
- [213] H.-Y. Lo *et al.*, “Cost-sensitive multi-label learning for audio tag annotation and retrieval,” *Multimedia, IEEE Transactions on*, vol. 13, no. 3, pp. 518–529, 2011.
- [214] J. Logue *et al.*, “Complexity in the signaling network: insights from the use of targeted inhibitors in cancer therapy,” *Genes & development*, vol. 26, no. 7, pp. 641–650, 2012.
- [215] G. Lohmann and others, “Eigenvector centrality mapping for analyzing connectivity patterns in fmri data of the human brain,” *PloS one*, vol. 5, no. 4, p. e10232, 2010.
- [216] F. López-Muñoz *et al.*, “Analgesic effects of combinations containing opioid drugs with either aspirin or acetaminophen in the rat,” *Drug Develop Res*, vol. 29, no. 4, pp. 299–304, 1993. [Online]. Available: <http://dx.doi.org/10.1002/ddr.430290408>
- [217] M. Los *et al.*, *Apoptotic pathways as targets for novel therapies in cancer and other diseases*. Springer, 2005.
- [218] J. Luo *et al.*, “Targeting the PI3K-Akt pathway in human cancer: Rationale and promise,” *Cancer Cell*, vol. 4, no. 4, pp. 257 – 262, 2003.
- [219] S. Ma *et al.*, “Penalized feature selection and classification in bioinformatics,” *Briefings in bioinformatics*, vol. 9, no. 5, pp. 392–403, 2008.
- [220] X. Ma *et al.*, “Cgi: a new approach for prioritizing genes by combining gene expression and protein–protein interaction data,” *Bioinformatics*, vol. 23, no. 2, pp. 215–221, 2007.
- [221] F. Machicao *et al.*, “Pleiotropic neuroprotective and metabolic effects of actovegin’s mode of action,” *Journal of the neurological sciences*, 2012.

- [222] A. Madiraju *et al.*, “Metformin suppresses gluconeogenesis by inhibiting mitochondrial glycerophosphate dehydrogenase,” *Nature*, 2014.
- [223] S.-M. Maira *et al.*, “Identification and characterization of nvp-bez235, a new orally available dual phosphatidylinositol 3-kinase/mammalian target of rapamycin inhibitor with potent in vivo antitumor activity,” *Molecular cancer therapeutics*, vol. 7, no. 7, pp. 1851–1863, 2008.
- [224] N. Mamoulis *et al.*, “Efficient top-k aggregation of ranked inputs,” *ACM Trans. Database Syst.*, vol. 32, no. 3, p. 19, 2007.
- [225] S. Mandrekar, “Dose-finding trial designs for combination therapies in oncology,” *Journal of Clinical Oncology*, vol. 32, no. 2, pp. 65–67, 2014.
- [226] H. Mann *et al.*, “On a test of whether one of two random variables is stochastically larger than the other,” *The annals of mathematical statistics*, vol. 18, no. 1, pp. 50–60, 1947.
- [227] T. Marill and D. Green, “On the effectiveness of receptors in recognition systems,” *Information Theory, IEEE Transactions on*, vol. 9, no. 1, pp. 11–17, 1963.
- [228] B. Markman *et al.*, “Status of pi3k inhibition and biomarker development in cancer therapeutics.” *Ann Oncol*, Aug 2009.
- [229] M. Marra *et al.*, “The genome sequence of the sars-associated coronavirus,” *Science*, vol. 300, no. 5624, pp. 1399–1404, 2003.
- [230] A. Masoudi-Nejad *et al.*, “Drug-target and disease networks: polypharmacology in the post-genomic era,” *In Silico Pharmacology*, vol. 1, no. 1, p. 17, 2013.
- [231] U. Matulonis *et al.*, “Cediranib (azd2171) is an active agent in recurrent epithelial ovarian cancer,” *Journal of Clinical Oncology*, vol. 26, no. 15, p. 5501, 2008.

- [232] M. Mauro *et al.*, “Sti571: a paradigm of new agents for cancer therapeutics,” *Journal of clinical oncology*, vol. 20, no. 1, pp. 325–334, 2002.
- [233] G. Mazandu *et al.*, “Generation and analysis of large-scale data-driven mycobacterium tuberculosis functional networks for drug target identification,” *Advances in bioinformatics*, vol. 2011, 2011.
- [234] J. McDermott *et al.*, “Topological analysis of protein co-abundance networks identifies novel host targets important for hcv infection and pathogenesis,” *BMC systems biology*, vol. 6, no. 1, p. 28, 2012.
- [235] M. McPherson *et al.*, “Drug receptor identification from multiple tissues using cellular-derived mrna display libraries,” *Chemistry & biology*, vol. 9, no. 6, pp. 691–698, 2002.
- [236] A. Mehta *et al.*, “Formal trials versus observational studies,” 2006.
- [237] N. Metropolis *et al.*, “Equation of state calculations by fast computing machines,” vol. 21, no. 6, pp. 1087–1092, 1953. [Online]. Available: <http://dx.doi.org/10.1063/1.1699114>
- [238] D. Meyer, “Support vector machines,” *The Interface to libsvm in package e1071. e1071 Vignette*, 2012.
- [239] T. Milenković *et al.*, “Uncovering biological network function via graphlet degree signatures.” *Cancer informatics*, no. 6, 2008.
- [240] T. Millward *et al.*, “Regulation of protein kinase cascades by protein phosphatase 2a.” *Trends Biochem Sci*, vol. 24, no. 5, pp. 186–191, May 1999.
- [241] O. Mirzoeva *et al.*, “Basal subtype and mapk/erk kinase (mek)-phosphoinositide 3-kinase feedback signaling determine susceptibility of breast cancer cells to mek inhibition,” *Cancer research*, vol. 69, no. 2, pp. 565–572, 2009.

- [242] R. Mishra *et al.*, “Jiao gu lan (gynostemma pentaphyllum): The chinese rasayan-current research scenario,” *International Journal of Research In Pharmaceutical and Biomedical Science*, vol. 2, no. 4, p. 1483, 2011.
- [243] W. Miskimins *et al.*, “Synergistic anti-cancer effect of phenformin and oxamate,” *PloS one*, vol. 9, no. 1, p. e85576, 2014.
- [244] M. Mitchell, “Combinations of anticancer drugs and immunotherapy,” *Cancer Immunology, Immunotherapy*, vol. 52, no. 11, pp. 686–692, 2003.
- [245] A. Mitra *et al.*, “Challenges and opportunities in achieving bioequivalence for fixed-dose combination products,” *The AAPS journal*, vol. 14, no. 3, pp. 646–655, 2012.
- [246] C. Mitsiades *et al.*, “The akt pathway: molecular targets for anti-cancer drug development,” *Current cancer drug targets*, vol. 4, no. 3, pp. 235–256, 2004.
- [247] S. Mizutani *et al.*, “Relating drug–protein interaction network with drug side effects,” *Bioinformatics*, vol. 28, no. 18, pp. i522–i528, 2012.
- [248] Y. Moreau *et al.*, “Computational tools for prioritizing candidate genes: boosting disease gene discovery,” *Nature Reviews Genetics*, vol. 13, no. 8, pp. 523–536, 2012.
- [249] S. Morgan *et al.*, “The cost of drug development: a systematic review,” *Health Policy*, vol. 100, no. 1, pp. 4–17, 2011.
- [250] A. Murarka *et al.*, “Metabolic analysis of wild-type escherichia coli and a pyruvate dehydrogenase complex (pdhc)-deficient derivative reveals the role of pdhc in the fermentative metabolism of glucose,” *Journal of Biological Chemistry*, vol. 285, no. 41, pp. 31 548–31 558, 2010.

- [251] N. Nachar, “The mann-whitney u: A test for assessing whether two independent samples come from the same distribution,” *Tutorials in Quantitative Methods for Psychology*, vol. 4, no. 1, pp. 13–20, 2008.
- [252] J. Nacher *et al.*, “A global view of drug-therapy interactions,” *BMC pharmacology*, vol. 8, no. 1, p. 5, 2008.
- [253] N. Nagarajan *et al.*, “Reliability and efficiency of algorithms for computing the significance of the mann–whitney test,” *Computational Statistics*, vol. 24, no. 4, pp. 605–622, 2009.
- [254] N. Nakayama *et al.*, “Kras or braf mutation status is a useful predictor of sensitivity to mek inhibition in ovarian cancer,” *British journal of cancer*, vol. 99, no. 12, pp. 2020–2028, 2008.
- [255] T. Namba *et al.*, “Suppression of expression of heat shock protein 70 by gefitinib and its contribution to pulmonary fibrosis,” *PloS one*, vol. 6, no. 11, p. e27296, 2011.
- [256] A. Nasevicius *et al.*, “Effective targeted gene knockdownin zebrafish,” *Nature genetics*, vol. 26, no. 2, pp. 216–220, 2000.
- [257] NCI. (2014, September) Nci drug dictionary. <http://www.cancer.gov/drugdictionary/>. National Cancer Institute. [Online]. Available: <http://www.cancer.gov/drugdictionary/?CdrID=558909>
- [258] M. Newman, “The mathematics of networks.”
- [259] NIH. (2014) Clinicaltrials.gov. Accessed 5 May 2014. [Online]. Available: <http://www.clinicaltrials.gov>
- [260] H. Noh *et al.*, “The role of protein kinase c activation in diabetic nephropathy,” *Kidney International*, vol. 72, pp. S49–S53, 2007.

- [261] A. Nolte *et al.*, “A note on the finite time behavior of simulated annealing,” *Mathematics of Operations Research*, vol. 25, no. 3, pp. pp. 476–484, 2000. [Online]. Available: <http://www.jstor.org/stable/3690480>
- [262] N. Normanno *et al.*, “Target-based agents against erbb receptors and their ligands: a novel approach to cancer treatment.” *Endocrine-related cancer*, vol. 10, no. 1, pp. 1–21, 2003.
- [263] C. Ogu *et al.*, “Drug interactions due to cytochrome p450,” *Proceedings (Baylor University. Medical Center)*, vol. 13, no. 4, p. 421, 2000.
- [264] T. Okuzumi *et al.*, “Inhibitor hijacking of akt activation.” *Nat Chem Biol*, vol. 5, no. 7, pp. 484–493, Jul 2009. [Online]. Available: <http://dx.doi.org/10.1038/nchembio.183>
- [265] M. Olayioye *et al.*, “The erbb signaling network: receptor heterodimerization in development and cancer,” *The EMBO journal*, vol. 19, no. 13, pp. 3159–3167, 2000.
- [266] J. O’Madadhain *et al.*, “Analysis and visualization of network data using jung.”
- [267] J. O’Madadhain *et al.*, “Analysis and visualization of network data using jung,” *Journal of Statistical Software*, vol. 10, no. 2, pp. 1–35, 2005.
- [268] J. Owens, “Determining druggability,” *Nature Reviews Drug Discovery*, vol. 6, no. 3, pp. 187–187, 2007.
- [269] R. Pal *et al.*, “A kinase inhibition map approach for tumor sensitivity prediction and combination therapy design for targeted drugs,” in *Pacific Symposium on Biocomputing*. World Scientific, 2012, pp. 351–62.
- [270] C. Park *et al.*, “Functional knowledge transfer for high-accuracy prediction of understudied biological processes,” *PLoS computational biology*, vol. 9, no. 3, p. e1002957, 2013.

- [271] R. Patnaik *et al.*, “Stimulation of glucose catabolism in escherichia coli by a potential futile cycle.” *Journal of bacteriology*, vol. 174, no. 23, pp. 7527–7532, 1992.
- [272] G. Pavlopoulos *et al.*, “Using graph theory to analyze biological networks,” *BioData Mining*, vol. 4, no. 1, pp. 1–27, 2011. [Online]. Available: <http://dx.doi.org/10.1186/1756-0381-4-10>
- [273] D. Pe’er *et al.*, “Principles and strategies for developing network models in cancer,” *Cell*, vol. 144, no. 6, pp. 864–873, 2011.
- [274] M. Peerzada *et al.*, “Pulmonary toxicities of tyrosine kinase inhibitors,” *Clin Adv Hematol Oncol*, vol. 9, no. 11, pp. 824–836, 2011.
- [275] A. Perrenoud *et al.*, “Impact of global transcriptional regulation by arca, arcB, cra, crp, cya, fnr, and mlc on glucose catabolism in escherichia coli,” *Journal of bacteriology*, vol. 187, no. 9, pp. 3171–3179, 2005.
- [276] J.-N. Phue *et al.*, “Transcription levels of key metabolic genes are the cause for different glucose utilization pathways in e. coli b (bl21) and e. coli k (jm109),” *Journal of Biotechnology*, vol. 109, no. 12, pp. 21 – 30, 2004.
- [277] —, “Glucose metabolism at high density growth of e. coli b and e. coli k: Differences in metabolic pathways are responsible for efficient glucose utilization in e. coli b as determined by microarrays and northern blot analyses,” *Biotechnology and Bioengineering*, vol. 90, no. 7, pp. 805–820, 2005.
- [278] A. Picon *et al.*, “Reducing the glucose uptake rate in escherichia coli affects growth rate but not protein production,” *Biotechnology and Bioengineering*, vol. 90, no. 2, pp. 191–200, 2005.

- [279] W. Piegorsch *et al.*, “Combining information,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 3, pp. 354–360, 2009.
- [280] V. Pihur *et al.*, “Rankaggreg, an r package for weighted rank aggregation,” *BMC bioinformatics*, vol. 10, no. 1, p. 62, 2009.
- [281] J. Platt, “Advances in kernel methods,” B. Schölkopf *et al.*, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208.
- [282] G. Pohl *et al.*, “Inactivation of the mitogen-activated protein kinase pathway as a potential target-based therapy in ovarian serous tumors with kras or braf mutations,” *Cancer Res*, vol. 65, no. 5, pp. 1994–2000, 2005. [Online]. Available: <http://cancerres.aacrjournals.org/content/65/5/1994.abstract>
- [283] M. Pontil *et al.*, “Support vector machines for 3d object recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 6, pp. 637–646, 1998.
- [284] R. Prill *et al.*, “Dynamic properties of network motifs contribute to biological network organization,” *PLoS biology*, vol. 3, no. 11, p. e343, 2005.
- [285] Y. Pritykin *et al.*, “Simple topological features reflect dynamics and modularity in protein interaction networks,” *PLoS computational biology*, vol. 9, no. 10, p. e1003243, 2013.
- [286] F. Provost, “Machine learning from imbalanced data sets 101,” in *Proceedings of the AAAI2000 workshop on imbalanced data sets*, 2000.
- [287] N. Pržulj, “Biological network comparison using graphlet degree distribution,” *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

- [288] T. Przytycka *et al.*, “Toward the dynamic interactome: it’s about time,” *Briefings in bioinformatics*, vol. 11, no. 1, pp. 15–29, 2010.
- [289] C. Qiu *et al.*, “Mechanism of activation and inhibition of the her4/erbb4 kinase.” *Structure*, vol. 16, no. 3, pp. 460–467, Mar 2008.
- [290] S. Ramón-García *et al.*, “Synergistic drug combinations for tuberculosis therapy identified by a novel high-throughput screen,” *Antimicrobial agents and chemotherapy*, vol. 55, no. 8, pp. 3861–3869, 2011.
- [291] J. Ramsay, *Functional data analysis*. Wiley Online Library, 2006.
- [292] J. Ramsay *et al.*, “Parameter estimation for differential equations: a generalized smoothing approach,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 5, pp. 741–796, 2007.
- [293] A. Ratsimamanga *et al.*, “Sodium oxamate for the treatment of diabetic conditions,” Oct. 2001, uS Patent 6,300,375.
- [294] J. Ray *et al.*, “Non-transcriptional regulatory processes shape transcriptional network dynamics,” *Nature Reviews Microbiology*, vol. 9, no. 11, pp. 817–828, 2011.
- [295] E. Rego *et al.*, “Retinoic acid (ra) and as2o3 treatment in transgenic models of acute promyelocytic leukemia (apl) unravel the distinct nature of the leukemogenic process induced by the pml-rar α and plzf-rar α oncoproteins,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 18, pp. 10 173–10 178, 2000.
- [296] A. Reibman *et al.*, “Numerical transient analysis of markov models,” *Computers & Operations Research*, vol. 15, no. 1, pp. 19 – 36, 1988. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0305054888900263>

- [297] K. Reinhardt, "Sorting in-place with a worst case complexity of $n \log n - 1.3 n + o(\log n)$ comparisons and $\epsilon n \log n + o(1)$ transports," in *Algorithms and Computation*. Springer, 1992, pp. 489–498.
- [298] S. Ritika *et al.*, "Formulation tactics for the delivery of poorly soluble drugs," *Int J Pharm Tech Res*, vol. 4, no. 3, pp. 914–23, 2012.
- [299] A. Ritz *et al.*, "Signaling hypergraphs," *Trends in biotechnology*, 2014.
- [300] M.-K. Riviere *et al.*, "Competing designs for drug combination in phase i dose-finding clinical trials," *Statistics in medicine*, 2014.
- [301] P. Roberts *et al.*, "Targeting the raf-mek-erk mitogen-activated protein kinase cascade for the treatment of cancer." *Oncogene*, vol. 26, no. 22, pp. 3291–3310, May 2007. [Online]. Available: <http://dx.doi.org/10.1038/sj.onc.1210422>
- [302] S. Robinette *et al.*, "Cluster analysis statistical spectroscopy using nuclear magnetic resonance generated metabolic data sets from perturbed biological systems," *Analytical Chemistry*, vol. 81, no. 16, pp. 6581–6589, 2009, PMID: 19624161.
- [303] L. Romano *et al.*, "Endo16 is required for gastrulation in the sea urchin lytechinus variegatus." *Dev Growth Differ*, vol. 48, no. 8, pp. 487–497, Oct 2006.
- [304] A. Roses, "Pharmacogenetics in drug discovery and development: a translational perspective," *Nature Reviews Drug Discovery*, vol. 7, no. 10, pp. 807–817, 2008.
- [305] D. Ruths *et al.*, "Rapidly exploring structural and dynamic properties of signaling networks using PathwayOracle," *BMC Syst Biol*, vol. 2, p. 76, 2008.
- [306] D. Saari *et al.*, "The copeland method," *Economic Theory*, vol. 8, no. 1, pp. 51–76, 1996.

- [307] Y. Saeys *et al.*, “A review of feature selection techniques in bioinformatics,” *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [308] S. Safavian *et al.*, “A survey of decision tree classifier methodology,” *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [309] M. Safran *et al.*, “Genecards version 3: the human gene integrator,” *Database*, vol. 2010, p. baq020, 2010.
- [310] S. Sahle *et al.*, “Simulation of biochemical networks using copasi: a complex pathway simulator,” in *Proceedings of the 38th conference on Winter simulation*, ser. WSC ’06. Winter Simulation Conference, 2006, pp. 1698–1706. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1218112.1218421>
- [311] S. Salvatore *et al.*, “Fastdtw: Toward accurate dynamic time warping in linear time and space,” in *3rd Workshop on Mining Temporal and Sequential Data*, 2004.
- [312] F. Sams-Dodd, “Target-based drug discovery: is something wrong?” *Drug discovery today*, vol. 10, no. 2, pp. 139–147, 2005.
- [313] G. Scardoni *et al.*, *New Frontiers in Graph Theory*. InTech, 2012, ch. 16.
- [314] J. Schlessinger, “Ligand-induced, receptor-mediated dimerization and activation of egf receptor,” *Cell*, vol. 110, no. 6, pp. 669–672, 2002.
- [315] B. Schoeberl *et al.*, “Therapeutically targeting erbb3: a key node in ligand-induced activation of the erbb receptor-pi3k axis,” *Science Signaling*, vol. 2, no. 77, p. ra31, 2009.
- [316] B. Schölkopf *et al.*, “Support vector regression with automatic accuracy control,” in *ICANN 98*. Springer, 1998, pp. 111–116.

- [317] L. Schriml *et al.*, “Disease ontology: a backbone for disease semantic integration,” *Nucleic acids research*, vol. 40, no. D1, pp. D940–D946, 2012.
- [318] M. Schulze, “Morphologie, syntax und semantik im rahmen der linksassoziativen grammatik,” in *GLDV-Jahrestagung*, 1997, pp. 133–149.
- [319] E. Sequeira, “Pubmed central—three years old and growing stronger,” *ARL*, vol. 228, no. 5, p. 9, 2003.
- [320] P. Shannon *et al.*, “Cytoscape: a software environment for integrated models of biomolecular interaction networks,” *Genome research*, vol. 13, no. 11, pp. 2498–2504, 2003.
- [321] V. Sheng and C. Ling, “Thresholding for making classifiers cost-sensitive,” in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 476.
- [322] H. Shi *et al.*, “Cell-based proteome profiling of potential dasatinib targets by use of affinity-based probes,” *Journal of the American Chemical Society*, vol. 134, no. 6, pp. 3001–3014, 2012.
- [323] R. Shoemaker, “The nci60 human tumour cell line anticancer drug screen,” *Nature Reviews Cancer*, vol. 6, no. 10, pp. 813–823, 2006.
- [324] Z. Shu *et al.*, “Importance evaluation of individuals in online social networks based on fuzzy ahp algorithm,” *AISS: Advances in Information Sciences and Service Sciences*, vol. 4, no. 8, pp. 84–92, 2012.
- [325] N. Sieben *et al.*, “Differential gene expression in ovarian tumors reveals dusp 4 and serpin 5 as key regulators for benign behavior of serous borderline tumors,” *J Clin Oncol*, vol. 23, no. 29, pp. 7257–7264, 2005. [Online]. Available: <http://jco.ascopubs.org/content/23/29/7257.abstract>

- [326] J. Sierra *et al.*, “Molecular mechanisms of acquired resistance to tyrosine kinase targeted therapy,” *Mol Cancer*, vol. 9, no. 1, p. 75, 2010. [Online]. Available: <http://www.molecular-cancer.com/content/9/1/75>
- [327] N. Singh *et al.*, “Molecular modeling and molecular dynamics studies of hydralazine with human dna methyltransferase 1.” *ChemMedChem*, vol. 4, no. 5, pp. 792–799, May 2009.
- [328] R. Singh *et al.*, “Global alignment of multiple protein interaction networks with application to functional orthology detection,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, pp. 12 763–12 768, 2008.
- [329] R. Smith *et al.*, “Intermine: a flexible data warehouse system for the integration and analysis of heterogeneous biological data,” *Bioinformatics*, vol. 28, no. 23, pp. 3163–3165, 2012.
- [330] A. Smola *et al.*, “A tutorial on support vector regression,” *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [331] I. Sobolá, “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates,” *Math. Comput. Simul.*, vol. 55, no. 1-3, pp. 271–280, 2001.
- [332] L. Solnica-Krezel, “Conserved patterns of cell movements during vertebrate gastrulation.” *Curr Biol*, vol. 15, no. 6, pp. R213–R228, Mar 2005.
- [333] H. Sondermann *et al.*, “Structural analysis of autoinhibition in the Ras activator Son of sevenless,” *Cell*, vol. 119, no. 3, pp. 393–405, Oct 2004.
- [334] F. Song *et al.*, “Feature selection using principal component analysis,” in *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 27–30.

- [335] S. Soundarajan *et al.*, “Which network similarity measure should you choose: An empirical study,” 2013.
- [336] A. Spencer *et al.*, “Novel akt inhibitor afuresertib in combination with bortezomib and dexamethasone demonstrates favorable safety profile and significant clinical activity in patients with relapsed/refractory multiple myeloma,” *Blood*, vol. 122, no. 21, pp. 283–283, 2013.
- [337] P. Sridhar *et al.*, “Opmet: A metabolic network-based algorithm for optimal drug target identification,” in *Pacific Symposium on Biocomputing*, 2008.
- [338] C. Stark *et al.*, “The biogrid interaction database: 2011 update,” *Nucleic Acids Research*, vol. 39, no. suppl 1, pp. D698–D704, 2011.
- [339] R. Steinman *et al.*, “Activation of Stat3 by cell confluence reveals negative regulation of Stat3 by cdk2,” *Oncogene*, vol. 22, no. 23, pp. 3608–3615, Jun 2003.
- [340] R. Steinmetz *et al.*, “Mechanisms Regulating the Constitutive Activation of the Extracellular Signal-Regulated Kinase (ERK) Signaling Pathway in Ovarian Cancer and the Effect of Ribonucleic Acid Interference for ERK1/2 on Cancer Cell Proliferation,” *Mol Endocrinol*, vol. 18, no. 10, pp. 2570–2582, 2004. [Online]. Available: <http://mend.endojournals.org/cgi/content/abstract/18/10/2570>
- [341] M. Stephens, “Edf statistics for goodness of fit and some comparisons,” *Journal of the American statistical Association*, vol. 69, no. 347, pp. 730–737, 1974.
- [342] R. Steriti, “Berberine for diabetes mellitus type 2,” *Natural Medicine Journal*, Oct 2010.
- [343] S. Stouffer *et al.*, “The american soldier: combat and its aftermath.(studies in social psychology in world war ii, vol. 2.)” 1949.

- [344] Y. Sun *et al.*, “Cost-sensitive boosting for classification of imbalanced data,” *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.
- [345] J. Sussman *et al.*, “Protein data bank (pdb): database of three-dimensional structural information of biological macromolecules,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 54, no. 6, pp. 1078–1084, 1998.
- [346] J. Suykens, “Nonlinear modelling and support vector machines,” in *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, vol. 1. IEEE, 2001, pp. 287–294.
- [347] F. Takes *et al.*, “Computing the eccentricity distribution of large graphs,” *Algorithms*, vol. 6, no. 1, pp. 100–118, 2013.
- [348] R. Tallarida, “Quantitative methods for assessing drug synergism,” *Genes & cancer*, vol. 2, no. 11, pp. 1003–1008, 2011.
- [349] J. Tang *et al.*, “Target inhibition networks: predicting selective combinations of drug-gable targets to block cancer survival pathways,” *PLoS computational biology*, vol. 9, no. 9, p. e1003226, 2013.
- [350] Y. Tang *et al.*, “Svms modeling for highly imbalanced classification,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 1, pp. 281–288, 2009.
- [351] R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.
- [352] J. Thomas *et al.*, “An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles,” *Genome Research*, vol. 11, no. 7, pp. 1227–1236, 2001.

- [353] K. Ting, *Inducing cost-sensitive trees via instance weighting*. Springer, 1998.
- [354] G. Tiwari *et al.*, “Drug delivery systems: an updated review,” *International journal of pharmaceutical investigation*, vol. 2, no. 1, p. 2, 2012.
- [355] L. Torgo *et al.*, “Regression by classification,” in *Advances in artificial intelligence*. Springer, 1996, pp. 51–60.
- [356] S. Trißl *et al.*, “Fast and practical indexing and querying of very large graphs,” in *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2007, pp. 845–856.
- [357] I. Tsang *et al.*, “Core vector machines: Fast svm training on very large data sets,” *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.
- [358] J. Tyson *et al.*, “Network dynamics and cell physiology,” *Nature Reviews Molecular Cell Biology*, vol. 2, no. 12, pp. 908–916, 2001.
- [359] L. Valerio Jr, “In silico toxicology models and databases as fda critical path initiative toolkits,” *Human genomics*, vol. 5, no. 3, p. 200, 2011.
- [360] M. van de Walle *et al.*, “Proposed mechanism of acetate accumulation in two recombinant escherichia coli strains during high density fermentation,” *Biotechnology and bioengineering*, vol. 57, no. 1, pp. 71–78, 1998.
- [361] P. van Poelje *et al.*, “Fructose-1, 6-bisphosphatase inhibitors for reducing excessive endogenous glucose production in type 2 diabetes,” in *Diabetes-Perspectives in Drug Therapy*. Springer, 2011, pp. 279–301.
- [362] N. van Riel, “Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments,” *Brief Bioinform*, vol. 7, no. 4, pp. 364–374, 2006. [Online]. Available: <http://bib.oxfordjournals.org/content/7/4/364.abstract>

- [363] V. Vapnik *et al.*, “Pattern recognition using generalized portrait method,” *Automation and Remote Control*, vol. 24, pp. 774–780, 1963.
- [364] A. Venkatesan *et al.*, “Bis(morpholino-1,3,5-triazine) derivatives: Potent adenosine 5'-triphosphate competitive phosphatidylinositol-3-kinase/mammalian target of rapamycin inhibitors: Discovery of compound 26 (pki-587), a highly efficacious dual inhibitor,” *Journal of Medicinal Chemistry*, vol. 53, no. 6, pp. 2636–2645, 2010. [Online]. Available: <http://pubs.acs.org/doi/abs/10.1021/jm901830p>
- [365] D. Verbanac *et al.*, “Combined in silico and in vitro approach to drug screening,” *Croatica chemica acta*, vol. 78, no. 2, pp. 133–139, 2005.
- [366] B. Viollet *et al.*, “Cellular and molecular mechanisms of metformin: an overview,” *Clinical Science*, vol. 122, no. 6, pp. 253–270, 2012. [Online]. Available: <http://www.clinsci.org/cs/122/cs1220253.htm>
- [367] D. Vitagliano *et al.*, “The tyrosine kinase inhibitor zd6474 blocks proliferation of ret mutant medullary thyroid carcinoma cells,” *Endocr Relat Cancer*, vol. 18, no. 1, pp. 1–11, 2011. [Online]. Available: <http://erc.endocrinology-journals.org/cgi/content/abstract/18/1/1>
- [368] F. Vitali *et al.*, “Network-based target ranking for polypharmacological therapies,” *Journal of biomedical informatics*, vol. 46, no. 5, pp. 876–881, 2013.
- [369] H. Vogel, *Drug Discovery and Evaluation: Pharmacological Assays*. Springer Science & Business Media, 2008, vol. 1.
- [370] P. Waage *et al.*, “Studies concerning affinity,” *Journal of chemical education*, vol. 63, no. 12, p. 1044, 1986.

- [371] N. Wages *et al.*, “Continual reassessment method for partial ordering,” *Biometrics*, vol. 67, no. 4, pp. 1555–1563, 2011.
- [372] F. Wang *et al.*, “The synergistic in vitro and in vivo antitumor effect of combination therapy with salinomycin and 5-fluorouracil against hepatocellular carcinoma,” *PloS one*, vol. 9, no. 5, p. e97414, 2014.
- [373] H. Wang *et al.*, “Dual labeling: Answering graph reachability queries in constant time,” in *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2006, p. 75.
- [374] Y. Wang *et al.*, “Reconstruct gene regulatory network using slice pattern model,” *BMC Genomics*, vol. 10, no. Suppl 1, p. S2, 2009. [Online]. Available: <http://www.biomedcentral.com/1471-2164/10/S1/S2>
- [375] F. Warnberg *et al.*, “Effect of a farnesyl transferase inhibitor (r115777) on ductal carcinoma in situ of the breast in a human xenograft model and on breast and ovarian cancer cell growth in vitro and in vivo,” *Breast Cancer Res*, vol. 8, no. 2, p. R21, 2006.
- [376] S. Wasserman *et al.*, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [377] J. Watson *et al.*, “Target selection and determination of function in structural genomics,” *IUBMB life*, vol. 55, no. 4-5, pp. 249–255, 2003.
- [378] D. Watts *et al.*, “Collective dynamics of small-world networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [379] R. Weinberg, *The biology of cancer*. Garland Science, 2013.
- [380] C. Whelan *et al.*, “Computing graphlet signatures of network nodes and motifs in cytoscape with graphletcounter,” *Bioinformatics*, vol. 28, no. 2, pp. 290–291, 2012.

- [381] P. Whetzel *et al.*, “The mged ontology: a resource for semantics-based description of microarray experiments,” *Bioinformatics*, vol. 22, no. 7, pp. 866–873, 2006.
- [382] D. Whitley, “The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best,” 1989. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.18.8195>
- [383] S. Wilhelm *et al.*, “Discovery and development of sorafenib: a multikinase inhibitor for treating cancer,” *Nat Rev Drug Discov*, vol. 5, no. 10, pp. 835–844, Oct 2006.
- [384] V. Williams, “Multiplying matrices faster than coppersmith-winograd,” in *Proceedings of the 44th symposium on Theory of Computing*. ACM, 2012, pp. 887–898.
- [385] D. Wishart *et al.*, “Drugbank: a knowledgebase for drugs, drug actions and drug targets.” *Nucleic Acids Res*, vol. 36, no. Database issue, pp. D901–D906, Jan 2008.
- [386] P. Wong *et al.*, “Closed-loop control of cellular functions using combinatory drugs guided by a stochastic search algorithm,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 5105–5110, 2008.
- [387] P. Workman *et al.*, “Drugging cancer genomes,” *Nature Reviews Drug Discovery*, vol. 12, no. 12, pp. 889–890, 2013.
- [388] S. Wuchty *et al.*, “Centers of complex networks,” *Journal of Theoretical Biology*, vol. 223, no. 1, pp. 45 – 53, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022519303000717>
- [389] M. Wymann *et al.*, “Phosphoinositide 3-kinase signalling which way to target?” *Trends in Pharmacological Sciences*, vol. 24, no. 7, pp. 366 – 376, 2003.

- [390] R. Xu *et al.*, “Large-scale extraction of accurate drug-disease treatment pairs from biomedical literature for drug repurposing,” *BMC bioinformatics*, vol. 14, no. 1, p. 181, 2013.
- [391] X. Yan *et al.*, “The identification of novel targets of mir-16 and characterization of their biological functions in cancer cells,” *Mol Cancer*, vol. 12, p. 92, 2013.
- [392] K. Yang *et al.*, “Finding multiple target optimal intervention in disease-related molecular network,” *Mol Syst Biol*, vol. 4, p. 228, 2008.
- [393] X. Yang *et al.*, “From xenomouse technology to panitumumab (abx-egf),” in *The Oncogenomics Handbook*, ser. Cancer Drug Discovery and Development, W. J. LaRochelle and R. A. Shimkets, Eds. Humana Press, 2005, pp. 647–657. [Online]. Available: <http://dx.doi.org/10.1385/1-59259-893-5:647>
- [394] T. Yap *et al.*, “First-in-man clinical trial of the oral pan-akt inhibitor mk-2206 in patients with advanced solid tumors,” *Journal of Clinical Oncology*, vol. 29, no. 35, pp. 4688–4695, 2011.
- [395] T. Yeh *et al.*, “Biological characterization of arry-142886 (azd6244), a potent, highly selective mitogen-activated protein kinase kinase 1/2 inhibitor,” *Clin Cancer Res*, vol. 13, no. 5, pp. 1576–1583, Mar 2007.
- [396] M. Yıldırım *et al.*, “Drug-target network,” *Nature Biotechnology*, vol. 25, no. 25, pp. 1119–1126, October 2007.
- [397] L. Yu *et al.*, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *ICML*, vol. 3, 2003, pp. 856–863.
- [398] H.-Y. Yuan *et al.*, “Fastsnp: an always up-to-date and extendable service for snp function analysis and prioritization,” *Nucleic acids research*, vol. 34, no. suppl 2, pp. W635–W641, 2006.

- [399] A. Zaczek *et al.*, “The diverse signaling network of egfr, her2, her3 and her4 tyrosine kinase receptors and the consequences for therapeutic approaches,” 2005.
- [400] B. Zadrozny *et al.*, “Cost-sensitive learning by cost-proportionate example weighting,” in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on.* IEEE, 2003, pp. 435–442.
- [401] A. Zanzoni *et al.*, “A network medicine approach to human disease,” *FEBS letters*, vol. 583, no. 11, pp. 1759–1765, 2009.
- [402] R. Zeineldin *et al.*, “Targeting the egf receptor for ovarian cancer therapy,” *Journal of oncology*, vol. 2010, 2009.
- [403] Q.-G. Zhang *et al.*, “Critical role of PTEN in the coupling between PI3K/Akt and JNK1/2 signaling in ischemic brain injury,” *FEBS Lett*, vol. 581, no. 3, pp. 495–505, Feb 2007.
- [404] X.-W. Zhang *et al.*, “Arsenic Trioxide Controls the Fate of the PML-RARa Oncoprotein by Directly Binding PML,” *Science*, vol. 328, no. 5975, pp. 240–243, 2010. [Online]. Available: <http://www.sciencemag.org/content/328/5975/240.abstract>
- [405] Z. Zhang *et al.*, “Phosphorylated erk is a potential predictor of sensitivity to sorafenib when treating hepatocellular carcinoma: evidence from an in vitro study.” *BMC Med*, vol. 7, p. 41, 2009.
- [406] L. Zhao *et al.*, “Evaluation of combination chemotherapy,” *Clin Cancer Res*, vol. 10, no. 23, pp. 7994–8004, 2004. [Online]. Available: <http://clincancerres.aacrjournals.org/content/10/23/7994.abstract>
- [407] X.-F. Zhao *et al.*, “Treatment with small interfering rna affects the microrna pathway and causes unspecific defects in zebrafish embryos,” *FEBS journal*, vol. 275, no. 9, pp. 2177–2184, 2008.

- [408] Y. Zhou *et al.*, “Kinetic analysis of sequential multistep reactions,” *The Journal of Physical Chemistry B*, vol. 111, no. 48, pp. 13 600–13 610, 2007.
- [409] Z. Zi *et al.*, “SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool.” *BMC Bioinformatics*, vol. 9, p. 342, 2008.
- [410] G. Zimmermann *et al.*, “Multi-target therapeutics: when the whole is greater than the sum of the parts.” *Drug Discov Today*, vol. 12, no. 1-2, pp. 34–42, Jan 2007.
- [411] L. Zon *et al.*, “In vivo drug discovery in the zebrafish,” *Nature reviews Drug discovery*, vol. 4, no. 1, pp. 35–44, 2005.

Appendix A

TENET Prioritization

In this appendix, we present the detailed ranks of the nodes in the test set of C_1 to C_4 and I_3 using the best TENET variant, random prioritization, LSA and *NetworkPrioritizer* using WBF and WASF. The ranks of these approaches are denoted as T , R , L , N_{WBF} and N_{WASF} , respectively. These tables are read from top to bottom and from left to right. Note that in these tables, the known targets are marked † .

Node ID	T	R	L	N_{WBF}	N_{WASF}	Node ID	T	R	L	N_{WBF}	N_{WASF}
Akt †	34	32	19	24	4	RHRG2	4	17	15	19	7
AktPIP3 †	9	20	26	4	3	RPI3K	15	30	16	30	7
AktPIP †	2	8	27	8	3	RPI3Kstar	24	34	17	27	6
AktPIPP †	7	22	14	6	3	RShGS	18	1	30	23	6
ERK	21	28	13	7	2	RShP	20	33	28	28	7
ERKP	6	11	12	3	2	RShc	11	12	24	29	7
ERKPP	10	16	11	1	1	Raf †	33	5	33	17	4
GS	26	18	6	18	6	Rafstar †	17	24	3	5	3
HRG	8	27	21	33	9	RasGDP †	27	9	1	20	5
MEK	14	31	25	16	3	RasGTP †	31	2	32	9	4
MEKP	3	10	7	10	3	ShGS	13	3	31	12	5
MEKPP †	5	15	23	2	2	ShP	30	14	2	26	6
PI3K	22	4	10	22	6	Shc	16	13	29	29	7
PI3Kstar †	35	6	20	13	5	P.I	25	26	4	25	5
PIP3	29	35	8	11	4	internalization	28	21	5	32	7
R	8	23	22	34	9	E	32	7	34	21	4
RP †	12	29	18	10	6	MKP3	19	19	34	15	2
RHRG	1	36	9	31	8	PP2A	23	25	34	14	3

Table A.1: Rank of nodes in test set of C_1 .

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
Glucose ⁺	17	11	1	6	36	Phosphate	23	49	32	5	6
F6P	46	23	2	6	37	Fumarate	20	44	33	6	24
FBP ⁺	28	22	3	6	20	Succinly-CoA	50	6	33	6	32
GAP	15	51	4	5	18	Succinate	34	2	34	6	17
DPG	51	7	5	6	16	GTP	45	40	35	7	42
PEP	30	5	6	6	38	GDP	45	32	36	7	42
PYR _{cyt}	16	46	7	4	5	Alanine	37	43	37	7	42
AMP	18	16	8	6	40	Aspartate	33	30	38	6	35
Lactate	31	4	9	6	36	Glutamate	25	9	39	5	31
G3P ⁺	9	8	10	6	28	H ₂ O	42	45	40	7	42
DHAP	4	48	11	6	28	ETF(red)	21	3	41	6	29
OXA _{cyt}	43	59	12	5	19	ETF(ox)	21	47	42	6	29
Aspartate _{cyt}	36	27	13	6	34	FADH2	27	18	43	6	30
Glutamate _{cyt}	36	25	14	6	34	FAD	8	10	44	6	30
OG _{cyt}	39	29	15	5	26	OG	44	24	45	3	15
Malate _{cyt}	41	12	16	3	11	Malate	5	50	46	2	9
Acetyl-CoA _{cyt} ⁺	40	13	17	7	42	OXA	19	26	47	4	7
CoA _{cyt}	40	42	17	7	42	Citrate	47	17	48	5	23
Isocitrate _{cyt}	52	55	18	5	27	Isocitrate	48	57	49	5	22
Citrate _{cyt}	29	39	19	5	25	ATP	1	1	50	1	1
ATP _{cyt}	11	52	20	2	2	ADP	12	35	51	4	3
ADP _{cyt}	6	14	21	2	10	NADP ⁺	38	37	52	6	33
NAD	22	21	22	4	12	NADPH	38	19	53	5	42
NADH _{cyt}	14	41	22	4	12	NAD ⁺	2	36	54	3	13
NADP _{cyt}	35	28	23	7	42	NADH	26	33	55	3	13
NADPH _{cyt}	53	58	23	6	33	Ubiquinone	24	15	24	4	14
PYR	13	38	28	3	4	Ubiquinol	10	54	25	4	14
CO ₂	49	20	29	4	8	Ferrocytochrome C ⁺	3	56	26	6	39
CoA	7	53	30	4	21	Ferricytochrome C	3	34	27	6	39
Acetyl-CoA ⁺	32	31	31	6	41						

Table A.2: Rank of nodes in test set of C₂.

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
GENE_E_Alx1	197	477	2	535	367	PROTEIN_M_L1	45	505	2	371	352
GENE_E_Apobec	192	199	2	412	380	PROTEIN_M_Lim	116	194	2	333	324
GENE_E_Blimp1 [†]	197	590	2	477	360	PROTEIN_M_Msp130	116	463	2	359	347
GENE_E_Bra [†]	194	272	2	400	367	PROTEIN_M_MspL	114	349	2	306	331
GENE_E_Brn [†]	193	517	2	435	386	PROTEIN_M_Not	116	236	2	313	328
GENE_E_CyP	193	99	2	473	374	PROTEIN_M_Notch [†]	9	50	2	73	68
GENE_E_Delta [†]	194	428	2	512	378	PROTEIN_M_Notch2 [†]	27	255	2	293	103
GENE_E_Dpt	120	41	2	545	389	PROTEIN_M_Nrl	76	367	2	208	152
GENE_E_Dri	192	421	2	532	378	PROTEIN_M_OrCt	116	146	2	341	340
GENE_E_ES	193	562	2	539	380	PROTEIN_M_Otx [†]	50	321	2	9	18
GENE_E_Endo16 [†]	192	485	2	391	357	PROTEIN_M_Pks	116	401	2	303	336
GENE_E_Erg	193	460	2	446	372	PROTEIN_M_Pmar1 [†]	55	468	2	263	150
GENE_E_Ets1 [†]	192	379	2	518	388	PROTEIN_M_Sm27	116	454	2	312	343
GENE_E_Eve [†]	194	358	2	488	368	PROTEIN_M_Sm30	114	113	2	316	319
GENE_E_Ficolin	194	276	2	541	367	PROTEIN_M_Sm50	114	116	2	315	337
GENE_E_FoxA	197	388	2	395	361	PROTEIN_M_Snail	56	293	2	215	154
GENE_E_FoxB	195	396	2	401	365	PROTEIN_M_SoxB1 [†]	86	197	2	65	80
GENE_E_FoxN23	120	192	2	428	387	PROTEIN_M_SoxC	126	482	2	265	281
GENE_E_FoxO	194	284	2	411	367	PROTEIN_M_SuH [†]	13	37	2	62	71
GENE_E_FvMo	192	42	2	534	379	PROTEIN_M_SuHN [†]	83	156	2	163	101
GENE_E_GataC	194	79	2	441	370	PROTEIN_M_SuTx	116	84	2	335	330
GENE_E_GataE [†]	193	589	2	447	374	PROTEIN_M_TBr	177	107	2	33	47
GENE_E_Gcad	192	123	2	527	393	PROTEIN_M_TCF [†]	7	596	2	201	105
GENE_E_Gcm	196	448	2	402	362	PROTEIN_M_Tel	183	577	2	49	58
GENE_E_Gelsolin	120	48	2	517	387	PROTEIN_M_Tgif	176	228	2	21	40
GENE_E_HesC [†]	192	519	2	482	392	PROTEIN_M_UMADelta [†]	148	553	2	213	148
GENE_E_Hex	193	377	2	391	372	PROTEIN_M_UMANrl	179	582	2	111	177
GENE_E_Hnf6	120	397	2	443	398	PROTEIN_M_UMR	116	117	2	346	337
GENE_E_Hox [†]	195	601	2	413	365	PROTEIN_M_UVAOtx [†]	194	11	2	409	373
GENE_E_Kakapo	120	411	2	530	387	PROTEIN_M_UbiqAlx1	195	356	2	503	367
GENE_E_Lim	192	239	2	550	378	PROTEIN_M_UbiqDelta [†]	194	70	2	551	374
GENE_E_Msp130	197	185	2	404	360	PROTEIN_M_UbiqEts1 [†]	192	163	2	437	388
GENE_E_MspL	195	6	2	499	364	PROTEIN_M_UbiqGcad	192	130	2	574	393
GENE_E_Not	120	490	2	420	386	PROTEIN_M_UbiqHesC [†]	192	431	2	471	392
GENE_E_Nrl	196	329	2	386	364	PROTEIN_M_UbiqHnf6	120	85	2	567	398
GENE_E_OrCt	192	314	2	559	380	PROTEIN_M_UbiqSoxB1 [†]	104	291	2	221	256
GENE_E_Otx [†]	194	518	2	392	368	PROTEIN_M_UbiqSoxC	193	96	2	502	384
GENE_E_Pks	192	76	2	500	379	PROTEIN_M_UbiqTel	193	509	2	396	381
GENE_E_Pmar1 [†]	193	549	2	415	372	PROTEIN_M_VEGFR	20	541	2	69	87
GENE_E_Sm27	197	347	2	394	360	PROTEIN_M_VEGFSignal	25	522	2	131	62
GENE_E_Sm30	120	591	2	448	391	PROTEIN_M_Wnt8 [†]	12	311	2	139	111
GENE_E_Sm50	198	470	2	373	358	PROTEIN_M_ΔCB [†]	3	497	2	50	57
GENE_E_Snail	120	157	2	543	387	PROTEIN_M_ΔBTCT [†]	30	327	2	84	15
GENE_E_SoxB1 [†]	192	53	2	398	385	PROTEIN_M_Δ13	112	259	2	301	317
GENE_E_SoxC	193	27	2	546	384	PROTEIN_P_Alx1	147	258	2	10	20
GENE_E_SuTx	192	91	2	389	379	PROTEIN_P_Apobec	116	348	2	317	327
GENE_E_TBr	193	456	2	384	372	PROTEIN_P_Blimp1 [†]	110	234	2	31	46
GENE_E_Tel	193	26	2	426	381	PROTEIN_P_Bra [†]	23	307	2	55	44
GENE_E_Tgif	194	12	2	375	368	PROTEIN_P_Brn [†]	159	385	2	75	90
GENE_E_VEGFR	194	418	2	452	367	PROTEIN_P_CAPK	118	432	2	354	350
GENE_E_Wnt8 [†]	193	242	2	515	373	PROTEIN_P_CyP	116	141	2	307	343
GENE_E_Δ13	193	585	2	492	372	PROTEIN_P_Delta [†]	1	325	2	85	66
GENE_M_Alx1	195	458	2	449	367	PROTEIN_P_Delta2 [†]	5	467	2	285	305
GENE_M_Apobec	193	312	2	390	373	PROTEIN_P_Dpt	116	416	2	364	329
GENE_M_Blimp1 [†]	197	372	2	432	361	PROTEIN_P_Dri	180	34	2	43	50
GENE_M_Bra [†]	194	581	2	383	368	PROTEIN_P_Endo16 [†]	116	564	2	358	326
GENE_M_Brn [†]	120	28	2	406	386	PROTEIN_P_Erg	169	204	2	13	24
GENE_M_CAPK	120	394	2	498	387	PROTEIN_P_Ets1 [†]	48	245	2	6	5
GENE_M_CyP	193	521	2	385	374	PROTEIN_P_Eve [†]	185	150	2	66	85
GENE_M_Delta [†]	194	374	2	496	374	PROTEIN_P_Ficolin	116	424	2	351	341
GENE_M_Dpt	192	403	2	536	379	PROTEIN_P_FoxA	188	288	2	56	76
GENE_M_Dri	192	46	2	542	378	PROTEIN_P_FoxB	191	246	2	80	98
GENE_M_Endo16 [†]	192	535	2	430	381	PROTEIN_P_FoxN23	175	557	2	124	199

Continued on next page.

Table A.3 Rank of nodes in test set of C_3 – continued from previous page

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
GENE_M_Erg	193	219	2	492	372	PROTEIN_P_FoxO	116	251	2	323	340
GENE_M_Ets1 [†]	192	126	2	423	388	PROTEIN_P_FvMo	116	105	2	308	326
GENE_M_Eve [†]	194	256	2	511	369	PROTEIN_P_GataC	152	554	2	260	278
GENE_M_Ficolin	194	205	2	468	367	PROTEIN_P_GataE [†]	19	257	2	18	8
GENE_M_FoxA	197	286	2	440	362	PROTEIN_P_Gcad	111	320	2	321	312
GENE_M_FoxB	195	476	2	442	365	PROTEIN_P_Gcm	46	501	2	20	35
GENE_M_FoxN23	120	452	2	565	387	PROTEIN_P_Gelsolin	116	502	2	319	334
GENE_M_FoxO	194	574	2	443	367	PROTEIN_P_Gro [†]	34	97	2	372	355
GENE_M_FvMo	192	417	2	393	379	PROTEIN_P_GroTCF [†]	130	10	2	51	27
GENE_M_GataC	194	542	2	410	370	PROTEIN_P_GroTFC	197	73	2	424	362
GENE_M_GataE [†]	193	247	2	422	375	PROTEIN_P_HesC [†]	88	30	2	19	30
GENE_M_Gcad	192	35	2	553	393	PROTEIN_P_Hex	80	249	2	17	11
GENE_M_Gcm	196	215	2	428	363	PROTEIN_P_Hnf6	153	548	2	23	37
GENE_M_Gelsolin	192	180	2	458	380	PROTEIN_P_Hox [†]	69	87	2	35	53
GENE_M_HesC [†]	192	447	2	563	392	PROTEIN_P_Kakapo	116	594	2	368	334
GENE_M_Hex	193	566	2	387	372	PROTEIN_P_L1	31	588	2	81	88
GENE_M_Hnf6	120	422	2	479	398	PROTEIN_P_Lim	116	378	2	366	320
GENE_M_Hox [†]	195	207	2	416	366	PROTEIN_P_Msp130	116	277	2	332	346
GENE_M_Kakapo	192	161	2	531	380	PROTEIN_P_MspL	115	531	2	343	334
GENE_M_Lim	192	98	2	474	379	PROTEIN_P_Not	116	287	2	330	325
GENE_M_Msp130	197	570	2	408	360	PROTEIN_P_Notch [†]	6	550	2	378	353
GENE_M_MspL	195	182	2	478	364	PROTEIN_P_Notch2 [†]	5	67	2	296	107
GENE_M_Not	120	213	2	552	387	PROTEIN_P_Nrl	73	208	2	168	119
GENE_M_Nrl	197	457	2	506	360	PROTEIN_P_OrCt	116	415	2	369	327
GENE_M_OrCt	193	586	2	379	373	PROTEIN_P_Otx [†]	52	315	2	22	17
GENE_M_Otx [†]	194	265	2	455	373	PROTEIN_P_Pks	116	139	2	309	326
GENE_M_Pks	193	164	2	449	371	PROTEIN_P_Pmar1 [†]	101	510	2	237	266
GENE_M_Pmar1 [†]	193	584	2	544	373	PROTEIN_P_Sm27	116	296	2	349	345
GENE_M_Sm27	197	342	2	405	360	PROTEIN_P_Sm30	115	7	2	357	323
GENE_M_Sm30	120	198	2	569	391	PROTEIN_P_Sm50	115	282	2	355	339
GENE_M_Sm50	198	151	2	442	359	PROTEIN_P_Snail	56	214	2	254	164
GENE_M_Snail	120	429	2	465	387	PROTEIN_P_SoxB1 [†]	65	252	2	70	84
GENE_M_SoxB1 [†]	192	154	2	556	385	PROTEIN_P_SoxC	146	187	2	257	304
GENE_M_SoxC	193	387	2	519	384	PROTEIN_P_SuH [†]	32	435	2	382	356
GENE_M_SuTx	192	444	2	403	379	PROTEIN_P_SuHN [†]	17	493	2	90	99
GENE_M_TBr	193	2	2	516	372	PROTEIN_P_SuTx	116	423	2	362	326
GENE_M_Tel	193	560	2	477	381	PROTEIN_P_TBr	171	481	2	45	49
GENE_M_Tgif	194	195	2	399	368	PROTEIN_P_TCF [†]	7	571	2	220	106
GENE_M_VEGFR	194	271	2	429	368	PROTEIN_P_Tel	189	25	2	57	59
GENE_M_Wnt8 [†]	193	583	2	397	374	PROTEIN_P_Tgif	172	168	2	15	39
GENE_M_z13	193	147	2	470	372	PROTEIN_P_UMADelta [†]	194	191	2	524	378
GENE_P_Alx1	195	524	2	472	364	PROTEIN_P_UMANrl	196	360	2	485	364
GENE_P_Apobec	192	556	2	562	382	PROTEIN_P_UVAOtx [†]	194	3	2	490	373
GENE_P_Blimp1 [†]	197	69	2	526	362	PROTEIN_P_UbiqAlx1	168	489	2	191	181
GENE_P_Bra [†]	194	313	2	487	368	PROTEIN_P_UbiqDelta [†]	194	300	2	558	378
GENE_P_Brn [†]	120	464	2	494	387	PROTEIN_P_UbiqES	116	203	2	350	337
GENE_P_CyP	193	404	2	520	374	PROTEIN_P_UbiqEts1 [†]	105	445	2	180	191
GENE_P_Delta [†]	194	504	2	418	378	PROTEIN_P_UbiqGcad	192	177	2	495	393
GENE_P_Dpt	120	353	2	417	389	PROTEIN_P_UbiqHesC [†]	107	366	2	253	255
GENE_P_Dri	192	274	2	427	378	PROTEIN_P_UbiqHnf6	49	290	2	246	242
GENE_P_Endo16 [†]	192	399	2	509	381	PROTEIN_P_UbiqSoxB1 [†]	192	122	2	555	392
GENE_P_Erg	193	532	2	537	372	PROTEIN_P_UbiqSoxC	145	38	2	223	233
GENE_P_Ets1 [†]	192	365	2	549	383	PROTEIN_P_UbiqTel	145	235	2	236	225
GENE_P_Eve [†]	194	111	2	510	369	PROTEIN_P_VEGFR	29	193	2	83	95
GENE_P_Ficolin	194	362	2	504	367	PROTEIN_P_VEGFSignal	51	363	2	89	63
GENE_P_FoxA	197	563	2	456	362	PROTEIN_P_Wnt8 [†]	12	382	2	148	110
GENE_P_FoxB	195	373	2	461	365	PROTEIN_P_cB [†]	3	523	2	48	56
GENE_P_FoxN23	120	331	2	566	387	PROTEIN_P_nBTCF [†]	30	508	2	98	14
GENE_P_FoxO	194	240	2	462	367	PROTEIN_P_z13	112	8	2	311	316
GENE_P_FvMo	192	201	2	493	380	mRNA_E_Alx1	165	304	2	146	269
GENE_P_GataC	194	78	2	533	370	mRNA_E_Apobec	124	64	2	248	274
GENE_P_GataE [†]	193	400	2	547	376	mRNA_E_Blimp1 [†]	137	446	2	109	115
GENE_P_Gcad	192	229	2	573	393	mRNA_E_Bra [†]	85	332	2	99	117

Continued on next page.

Table A.3 Rank of nodes in test set of C_3 – continued from previous page

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
GENE_P_Gcm	196	15	2	419	363	mRNA_E_Brn [†]	173	506	2	144	112
GENE_P_Gelsolin	120	533	2	507	390	mRNA_E_CAPK	120	190	2	320	351
GENE_P_HesC [†]	192	301	2	476	385	mRNA_E_CyP	138	409	2	261	264
GENE_P_Hex	193	495	2	523	372	mRNA_E_Delta [†]	59	569	2	95	130
GENE_P_Hnf6	120	109	2	455	392	mRNA_E_Dpt	95	390	2	241	230
GENE_P_Hox [†]	195	419	2	377	366	mRNA_E_Dri	121	200	2	187	292
GENE_P_Kakapo	120	576	2	407	390	mRNA_E_ES	75	425	2	233	218
GENE_P_Lim	192	333	2	469	379	mRNA_E_Endo16 [†]	124	462	2	3	108
GENE_P_Msp130	197	555	2	436	360	mRNA_E_Erg	131	264	2	247	283
GENE_P_MspL	195	593	2	505	364	mRNA_E_Ets1 [†]	61	430	2	122	165
GENE_P_Not	120	544	2	420	387	mRNA_E_Eve [†]	108	32	2	102	144
GENE_P_Nrl	196	212	2	538	364	mRNA_E_Ficolin	144	142	2	156	201
GENE_P_OrCt	192	479	2	554	382	mRNA_E_FoxA	113	405	2	127	197
GENE_P_Otx [†]	194	167	2	380	373	mRNA_E_FoxB	160	188	2	207	301
GENE_P_Pks	192	144	2	524	380	mRNA_E_FoxN23	72	298	2	137	121
GENE_P_Pmar1 [†]	193	600	2	439	373	mRNA_E_FoxO	144	181	2	211	211
GENE_P_Sm27	197	597	2	480	360	mRNA_E_FvMo	124	579	2	155	149
GENE_P_Sm30	120	221	2	561	391	mRNA_E_GataC	122	480	2	237	271
GENE_P_Sm50	198	376	2	444	358	mRNA_E_GataE [†]	97	115	2	157	129
GENE_P_Snail	120	389	2	433	387	mRNA_E_Gcad	22	83	2	105	75
GENE_P_SoxB1 [†]	192	238	2	529	392	mRNA_E_Gcm	94	546	2	136	158
GENE_P_SoxC	193	434	2	421	378	mRNA_E_Gelsolin	95	137	2	184	178
GENE_P_SuTx	192	369	2	508	380	mRNA_E_HesC [†]	60	22	2	252	155
GENE_P_TBr	193	492	2	398	372	mRNA_E_Hex	128	375	2	170	285
GENE_P_Tel	193	267	2	466	377	mRNA_E_Hnf6	53	173	2	249	122
GENE_P_Tgif	194	330	2	376	368	mRNA_E_Hox [†]	125	350	2	78	120
GENE_P_VEGFR	194	512	2	528	368	mRNA_E_Kakapo	95	269	2	176	169
GENE_P_Wnt8 [†]	193	292	2	445	374	mRNA_E_Lim	124	250	2	133	116
GENE_P_z13	193	54	2	381	372	mRNA_E_Msp130	163	392	2	182	246
PRE_E_Gcad	18	66	2	570	397	mRNA_E_MspL	138	88	2	154	221
PRE_E_Notch [†]	18	364	2	576	399	mRNA_E_Not	95	159	2	236	153
PRE_E_Otx [†]	18	134	2	513	394	mRNA_E_Notch [†]	39	222	2	271	188
PRE_E_SoxB1 [†]	18	371	2	481	396	mRNA_E_Nrl	113	178	2	107	142
PRE_E_SuH [†]	18	90	2	521	399	mRNA_E_OrCt	124	337	2	216	170
PRE_E_UMR	18	343	2	547	400	mRNA_E_Otx [†]	38	231	2	64	70
PRE_E_UVAOtx [†]	18	241	2	467	399	mRNA_E_Pks	124	209	2	230	157
PRE_E_UbiqSoxB1 [†]	18	86	2	486	399	mRNA_E_Pmar1 [†]	93	127	2	97	113
PRE_E_VEGF	18	77	2	540	399	mRNA_E_Sm27	163	36	2	120	228
PRE_E_cB [†]	18	143	2	568	399	mRNA_E_Sm30	66	237	2	161	143
PRE_M_Gcad	18	57	2	557	397	mRNA_E_Sm50	151	604	2	110	224
PRE_M_Notch [†]	18	233	2	560	399	mRNA_E_Snail	92	308	2	250	167
PRE_M_Otx [†]	18	536	2	431	394	mRNA_E_SoxB1 [†]	26	230	2	117	92
PRE_M_SoxB1 [†]	18	186	2	514	396	mRNA_E_SoxC	78	95	2	274	253
PRE_M_SuH [†]	18	486	2	434	399	mRNA_E_SuH [†]	39	103	2	283	182
PRE_M_UMADelta [†]	18	573	2	454	399	mRNA_E_SuTx	124	114	2	158	166
PRE_M_UMANrl	18	436	2	489	399	mRNA_E_TBr	132	414	2	218	289
PRE_M_UMR	18	248	2	464	400	mRNA_E_Tel	77	40	2	226	270
PRE_M_UbiqSoxB1 [†]	18	561	2	575	399	mRNA_E_Tgif	140	206	2	249	291
PRE_M_cB [†]	18	266	2	483	399	mRNA_E_UMR	39	393	2	292	262
PRE_P_Ets1 [†]	18	220	2	519	395	mRNA_E_UVAOtx [†]	39	326	2	284	183
PRE_P_Gcad	18	500	2	438	397	mRNA_E_UbiqSoxB1 [†]	39	260	2	275	248
PRE_P_LI	18	384	2	497	399	mRNA_E_VEGF	39	306	2	235	194
PRE_P_Otx [†]	18	1	2	475	394	mRNA_E_VEGFR	134	578	2	134	210
PRE_P_UbiqAlx1	18	383	2	459	399	mRNA_E_Wnt8 [†]	84	81	2	94	114
PRE_P_UbiqES	18	534	2	491	400	mRNA_E_cB [†]	39	494	2	243	151
PRE_P_UbiqEts1 [†]	18	101	2	501	399	mRNA_E_z13	102	224	2	113	126
PRE_P_UbiqHesC [†]	18	351	2	564	399	mRNA_M_Alxl	100	357	2	130	272
PRE_P_UbiqHnf6	18	263	2	507	400	mRNA_M_Apobec	138	498	2	259	234
PRE_P_UbiqSoxC	18	341	2	453	399	mRNA_M_Blimp1 [†]	133	47	2	165	277
PRE_P_UbiqTel	18	176	2	548	399	mRNA_M_Bra [†]	90	302	2	96	161
PRE_P_cB [†]	18	368	2	571	399	mRNA_M_Brn [†]	94	45	2	185	258
PROTEIN_E_Alxl	162	44	2	24	21	mRNA_M_CAPK	95	196	2	270	192
PROTEIN_E_Apobec	116	472	2	363	331	mRNA_M_CyP	138	75	2	231	212

Continued on next page.

Table A.3 Rank of nodes in test set of C_3 – continued from previous page

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
PROTEIN_E_Blimp1 [†]	117	474	2	38	42	mRNA_M_Delta [†]	79	82	2	104	145
PROTEIN_E_Bra [†]	4	461	2	8	6	mRNA_M_Dpt	124	580	2	206	257
PROTEIN_E_Brn [†]	155	567	2	72	64	mRNA_M_Dri	123	227	2	193	279
PROTEIN_E_CAPK	118	310	2	367	350	mRNA_M_Endo16 [†]	124	565	2	201	231
PROTEIN_E_CyP	116	437	2	328	348	mRNA_M_Erg	131	572	2	194	299
PROTEIN_E_Delta [†]	15	14	2	92	83	mRNA_M_Ets1 [†]	61	119	2	114	141
PROTEIN_E_Delta2 [†]	43	149	2	297	307	mRNA_M_Eve [†]	106	595	2	108	223
PROTEIN_E_Dpt	116	370	2	324	322	mRNA_M_Ficolin	144	89	2	138	180
PROTEIN_E_Dri	166	232	2	37	41	mRNA_M_FoxA	109	275	2	128	268
PROTEIN_E_ES	114	459	2	365	318	mRNA_M_FoxB	160	218	2	169	297
PROTEIN_E_Endo16 [†]	116	592	1	1	1	mRNA_M_FoxN23	71	202	2	183	136
PROTEIN_E_Erg	167	129	2	14	23	mRNA_M_FoxO	144	473	2	116	203
PROTEIN_E_Ets1 [†]	42	538	2	12	4	mRNA_M_FvMo	124	278	2	193	244
PROTEIN_E_Eve [†]	182	344	2	60	65	mRNA_M_GataC	122	359	2	251	273
PROTEIN_E_Ficolin	116	21	2	318	338	mRNA_M_GataE [†]	99	297	2	210	227
PROTEIN_E_FoxA	186	552	2	58	74	mRNA_M_Gcad	22	545	2	126	73
PROTEIN_E_FoxB	190	547	2	81	96	mRNA_M_Gcm	91	59	2	91	172
PROTEIN_E_FoxN23	174	160	2	195	229	mRNA_M_Gelsolin	124	443	2	212	232
PROTEIN_E_FoxO	116	133	2	352	335	mRNA_M_HesC [†]	62	499	2	159	147
PROTEIN_E_FvMo	116	24	2	340	315	mRNA_M_Hex	128	124	2	224	295
PROTEIN_E_GataC	150	334	2	277	303	mRNA_M_Hnf6	53	172	2	177	123
PROTEIN_E_GataE [†]	11	273	2	5	7	mRNA_M_Hox [†]	124	138	2	172	184
PROTEIN_E_Gcad	111	514	2	353	310	mRNA_M_Kakapo	124	488	2	209	202
PROTEIN_E_Gcm	37	575	2	34	31	mRNA_M_Lim	124	19	2	200	193
PROTEIN_E_Gelsolin	116	469	2	348	342	mRNA_M_Msp130	163	189	2	171	215
PROTEIN_E_Gro [†]	33	294	2	425	354	mRNA_M_MspL	138	295	2	152	240
PROTEIN_E_GroTCF [†]	129	346	2	52	26	mRNA_M_Not	95	55	2	240	206
PROTEIN_E_GroTFC	197	20	2	463	361	mRNA_M_Notch [†]	39	170	2	276	187
PROTEIN_E_HesC [†]	53	475	2	28	19	mRNA_M_Nrl	161	503	2	132	216
PROTEIN_E_Hex	98	507	2	11	10	mRNA_M_OrCt	138	174	2	227	207
PROTEIN_E_Hnf6	149	516	2	16	33	mRNA_M_Otx [†]	35	395	2	88	77
PROTEIN_E_Hox [†]	70	100	2	27	52	mRNA_M_Pks	138	440	2	217	190
PROTEIN_E_Kakapo	116	322	2	326	342	mRNA_M_Pmar1 [†]	87	305	2	129	132
PROTEIN_E_LI	45	108	2	347	352	mRNA_M_Sm27	163	484	2	115	213
PROTEIN_E_Lim	116	145	2	304	309	mRNA_M_Sm30	66	336	2	173	138
PROTEIN_E_Msp130	116	525	2	331	342	mRNA_M_Sm50	151	451	2	125	205
PROTEIN_E_MspL	114	559	2	327	333	mRNA_M_Snail	92	558	2	192	156
PROTEIN_E_Not	116	52	2	360	314	mRNA_M_SoxB1 [†]	26	61	2	106	94
PROTEIN_E_Notch [†]	10	18	2	76	69	mRNA_M_SoxC	78	169	2	229	173
PROTEIN_E_Notch2 [†]	28	427	2	262	102	mRNA_M_SuH [†]	39	352	2	265	189
PROTEIN_E_Nrl	74	29	2	202	186	mRNA_M_SuTx	124	118	2	238	219
PROTEIN_E_OrCt	116	303	2	345	332	mRNA_M_TBr	135	17	2	267	296
PROTEIN_E_Otx [†]	57	426	2	24	16	mRNA_M_Tel	77	31	2	279	293
PROTEIN_E_Pks	116	483	2	344	315	mRNA_M_Tgif	142	102	2	186	302
PROTEIN_E_Pmar1 [†]	55	381	2	149	127	mRNA_M_UMADelta [†]	39	355	2	273	235
PROTEIN_E_Sm27	116	438	2	338	340	mRNA_M_UMANrl	39	420	2	258	245
PROTEIN_E_Sm30	114	16	2	337	321	mRNA_M_UMR	39	136	2	287	262
PROTEIN_E_Sm50	114	5	2	322	338	mRNA_M_UbiqSoxB1 [†]	39	153	2	278	247
PROTEIN_E_Snail	56	442	2	166	135	mRNA_M_VEGFR	134	450	2	162	185
PROTEIN_E_SoxB1 [†]	86	602	2	74	89	mRNA_M_Wnt8 [†]	81	33	2	112	128
PROTEIN_E_SoxC	126	551	2	288	267	mRNA_M_cB [†]	39	128	2	245	160
PROTEIN_E_SuH [†]	14	165	2	67	67	mRNA_M_z13	95	539	2	150	131
PROTEIN_E_SuHN [†]	82	106	2	123	100	mRNA_P_Alx1	154	402	2	222	280
PROTEIN_E_SuTx	116	380	2	334	315	mRNA_P_Apobec	124	537	2	239	263
PROTEIN_E_TBr	170	323	2	40	48	mRNA_P_Blimp1 [†]	133	270	2	175	275
PROTEIN_E_TCF [†]	8	412	2	188	104	mRNA_P_Bra [†]	88	68	2	164	162
PROTEIN_E_Tel	183	408	2	63	60	mRNA_P_Brn [†]	94	43	2	266	265
PROTEIN_E_Tgif	164	132	2	41	38	mRNA_P_CAPK	120	158	2	370	350
PROTEIN_E_UMADelta [†]	194	71	2	522	378	mRNA_P_CyP	138	135	2	244	239
PROTEIN_E_UMANrl	196	526	2	388	364	mRNA_P_Delta [†]	58	289	2	143	125
PROTEIN_E_UMR	116	439	2	310	337	mRNA_P_Dpt	95	540	2	203	195
PROTEIN_E_UVAOtx [†]	156	254	2	103	118	mRNA_P_Dri	123	120	2	280	290
PROTEIN_E_UbiqAlx1	193	74	2	451	367	mRNA_P_Endo16 [†]	124	513	2	174	200

Continued on next page.

Table A.3 Rank of nodes in test set of C_3 – continued from previous page

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
PROTEIN_E_UbiqDelta [†]	194	281	2	414	378	mRNA_P_Erg	131	309	2	196	298
PROTEIN_E_UbiqES	193	4	2	484	380	mRNA_P_Ets1 [†]	24	345	2	86	86
PROTEIN_E_UbiqEts1 [†]	192	407	2	525	388	mRNA_P_Eve [†]	106	520	2	168	174
PROTEIN_E_UbiqGcad	192	398	2	572	393	mRNA_P_Ficolin	144	605	2	145	204
PROTEIN_E_UbiqHesC [†]	192	317	2	457	392	mRNA_P_FoxA	109	324	2	151	237
PROTEIN_E_UbiqHnf6	120	226	2	559	398	mRNA_P_FoxB	160	49	2	225	284
PROTEIN_E_UbiqSoxB1 [†]	104	319	2	178	209	mRNA_P_FoxN23	71	515	2	141	134
PROTEIN_E_UbiqSoxC	193	131	2	450	384	mRNA_P_FoxO	144	449	2	242	250
PROTEIN_E_UbiqTel	193	171	2	483	381	mRNA_P_FvMo	124	318	2	198	196
PROTEIN_E_VEGF	2	316	2	30	32	mRNA_P_GataC	122	175	2	204	294
PROTEIN_E_VEGFR	20	104	2	61	78	mRNA_P_GataE [†]	41	299	2	93	137
PROTEIN_E_VEGFSignal	25	56	2	101	61	mRNA_P_Gcad	22	184	2	119	72
PROTEIN_E_Wnt8 [†]	12	65	2	135	109	mRNA_P_Gcm	47	455	2	100	139
PROTEIN_E_cB [†]	4	162	2	42	55	mRNA_P_Gelsolin	95	72	2	232	163
PROTEIN_E_nBTFC [†]	36	9	2	79	13	mRNA_P_HesC [†]	113	211	2	263	261
PROTEIN_E_zl3	112	339	2	339	313	mRNA_P_Hex	127	210	2	234	300
PROTEIN_GCM	194	453	2	256	306	mRNA_P_Hnf6	87	216	2	264	259
PROTEIN_M_Alx1	162	183	2	29	22	mRNA_P_Hox [†]	124	391	2	140	214
PROTEIN_M_Apobec	116	51	2	356	340	mRNA_P_Kakapo	95	478	2	190	175
PROTEIN_M_Blimp1 [†]	103	354	2	44	45	mRNA_P_L1	39	58	2	291	220
PROTEIN_M_Bra [†]	96	80	2	36	43	mRNA_P_Lim	124	528	2	153	171
PROTEIN_M_Brn [†]	158	465	2	68	82	mRNA_P_Msp130	163	268	2	118	236
PROTEIN_M_CAPK	116	39	2	314	342	mRNA_P_MspL	143	121	2	214	222
PROTEIN_M_CyP	116	125	2	361	349	mRNA_P_Not	95	13	2	286	208
PROTEIN_M_Delta [†]	16	413	2	87	81	mRNA_P_Nrl	113	62	2	181	179
PROTEIN_M_Delta2 [†]	44	527	2	299	308	mRNA_P_OrCt	124	283	2	228	263
PROTEIN_M_Dpt	116	361	2	325	339	mRNA_P_Otx [†]	35	94	2	77	79
PROTEIN_M_Dri	181	338	2	53	51	mRNA_P_Pks	124	599	2	205	176
PROTEIN_M_Endo16 [†]	116	543	2	342	330	mRNA_P_Pmar1 [†]	89	155	2	147	133
PROTEIN_M_Erg	167	410	2	46	25	mRNA_P_Sm27	163	166	2	181	238
PROTEIN_M_Ets1 [†]	54	511	2	7	3	mRNA_P_Sm30	68	93	2	219	146
PROTEIN_M_Eve [†]	184	386	2	82	93	mRNA_P_Sm50	157	487	2	167	217
PROTEIN_M_Ficolin	116	441	2	298	340	mRNA_P_Snail	92	279	2	181	168
PROTEIN_M_FoxA	187	606	2	59	91	mRNA_P_SoxB1 [†]	64	406	2	289	198
PROTEIN_M_FoxB	190	262	2	71	97	mRNA_P_SoxC	139	280	2	199	286
PROTEIN_M_FoxN23	178	568	2	189	260	mRNA_P_SuTx	124	110	2	227	176
PROTEIN_M_FoxO	116	253	2	305	341	mRNA_P_TBr	132	225	2	253	282
PROTEIN_M_FvMo	116	179	2	336	330	mRNA_P_Tel	136	328	2	197	287
PROTEIN_M_GataC	150	63	2	255	276	mRNA_P_Tgif	141	148	2	179	288
PROTEIN_M_GataE [†]	21	603	2	4	9	mRNA_P_UbiqAlx1	39	244	2	269	243
PROTEIN_M_Gcad	111	587	2	329	311	mRNA_P_UbiqES	39	223	2	290	262
PROTEIN_M_Gcm	63	285	2	26	34	mRNA_P_UbiqEts1 [†]	39	529	2	281	241
PROTEIN_M_Gelsolin	116	140	2	302	344	mRNA_P_UbiqHesC [†]	39	340	2	294	254
PROTEIN_M_Gro [†]	34	112	2	374	355	mRNA_P_UbiqHnf6	39	92	2	295	251
PROTEIN_M_GroTCF [†]	130	433	2	54	28	mRNA_P_UbiqSoxC	39	261	2	268	252
PROTEIN_M_GroTFC	197	491	2	460	362	mRNA_P_UbiqTel	39	217	2	282	249
PROTEIN_M_HesC [†]	67	335	2	32	29	mRNA_P_VEGFR	40	60	2	142	226
PROTEIN_M_Hex	98	23	2	39	12	mRNA_P_Wnt8 [†]	81	598	2	121	124
PROTEIN_M_Hnf6	149	496	2	25	36	mRNA_P_cB [†]	39	152	2	272	159
PROTEIN_M_Hox [†]	119	466	2	47	54	mRNA_P_zl3	95	530	2	160	140
PROTEIN_M_Kakapo	116	243	2	300	344	none	2	471	2	2	2

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
BM	29	30	36	18	18	Enolase	1	30	35	35	35
Acetate [†]	36	8	46	1	1	Fdp [†]	30	14	34	28	28
Glucose [†]	22	13	47	2	2	Citrate synthase	29	32	39	26	26
ACoA	27	1	23	8	8	Icd [†]	15	37	12	13	13
AKG	46	23	21	9	9	Icd_P [†]	16	1	18	21	21
cAMP	40	7	30	22	22	Mdh	7	3	10	36	36
FBP	17	36	37	12	12	MaeAB	25	45	11	29	29
G6P [†]	21	18	7	15	15	PckA	6	12	24	34	34
Glyoxylate	25	26	25	7	7	Pdh [†]	24	7	22	30	30
Isocitrate [†]	43	20	15	14	14	PfkA	23	15	8	28	28
Malate	20	28	26	16	16	Ppc [†]	4	23	40	33	33
OAA	44	43	17	10	10	PpsA [†]	18	38	9	32	32
PEP [†]	11	3	16	5	5	PykF	20	18	4	27	27
PG3	41	46	14	11	11	Cra	34	24	41	4	4
Pyruvate	6	17	28	3	3	Cra-FBP	36	10	42	42	42
AceA	19	15	19	23	23	Crp	37	42	6	43	43
AceB [†]	9	35	38	25	25	Crp-cAMP	22	31	45	6	6
AceK	8	12	20	17	17	IclR	43	35	3	20	20
Acoa2act [†]	5	16	13	39	39	PdhR	41	26	1	41	41
Acs	33	24	32	19	19	PdhR-PYR	38	16	43	44	44
Akg2mal	39	40	29	37	37	EIIA [†]	45	4	44	38	38
Cya [†]	13	5	27	45	45	EIIA_P	13	28	5	24	24
Emp	2	6	31	31	31	EIICB [†]	44	34	2	40	40

Table A.4: Rank of nodes in test set of C₄.

Node ID	T	R	L	N _{WBF}	N _{WASF}	Node ID	T	R	L	N _{WBF}	N _{WASF}
GENE_E_Apobec	56	19	1	43	51	PROTEIN_M_LL1	9	47	1	39	38
GENE_E_ES	45	58	1	57	52	PROTEIN_M_Pmar1 [†]	11	41	1	25	14
GENE_E_Kakapo	47	32	1	56	57	PROTEIN_M_Sm27	44	37	1	33	37
GENE_E_OrCt	56	29	1	60	53	PROTEIN_M_SoxB1 [†]	20	18	1	6	8
GENE_E_Otx [†]	27	50	1	42	44	PROTEIN_M_UbiqSoxC	32	11	1	52	55
GENE_E_Pmar1 [†]	22	56	1	44	49	PROTEIN_P_Delta2 [†]	1	40	1	31	30
GENE_E_Sm50	59	42	1	40	39	PROTEIN_P_Dpt	36	33	1	38	32
GENE_E_SuTx	57	10	1	41	50	PROTEIN_P_Dri	19	5	1	4	4
GENE_M_Alx1	58	39	1	47	43	PROTEIN_P_FoxA	42	26	1	5	7
GENE_M_CAPK	46	31	1	51	56	PROTEIN_P_Pks	35	15	1	32	31
GENE_M_Eve [†]	17	23	1	55	48	PROTEIN_P_SoxB1 [†]	15	21	1	7	9
GENE_M_FoxO	51	60	1	46	42	mRNA_E_Dri	53	20	1	17	26
GENE_M_Nrl	55	38	1	53	41	mRNA_E_ES	28	34	1	22	20
GENE_M_VEGFR	38	24	1	45	45	mRNA_E_Ets1 [†]	30	35	1	11	17
GENE_P_Bra [†]	23	28	1	50	46	mRNA_E_Gcm	8	55	1	13	16
GENE_P_Eve [†]	14	12	1	54	47	mRNA_M_Gcad	5	54	1	12	6
GENE_P_OrCt	26	45	1	59	54	mRNA_M_HesC [†]	18	46	1	15	13
GENE_P_Sm27	34	61	1	48	40	mRNA_M_Hex	48	14	1	20	28
PRE_E_UbiqSoxB1 [†]	12	9	1	49	58	mRNA_M_Not	29	6	1	24	19
PRE_E_VEGF	16	8	1	58	59	mRNA_M_Snail	43	57	1	19	15
PRE_P_cB [†]	6	30	1	61	60	mRNA_M_SoxB1 [†]	4	7	1	10	10
PROTEIN_E_Apobec	41	43	1	37	33	mRNA_M_TBr	54	2	1	26	29
PROTEIN_E_Ficolin	50	3	1	34	35	mRNA_M_Tel	21	4	1	28	27
PROTEIN_E_HesC [†]	7	44	1	2	3	mRNA_M_UbiqSoxB1 [†]	25	16	1	27	22
PROTEIN_E_Hex	10	48	1	1	1	mRNA_M_z13	13	53	1	14	12
PROTEIN_E_OrCt	39	27	1	36	34	mRNA_P_Apobec	31	52	1	23	24
PROTEIN_E_Sm27	49	36	1	35	36	mRNA_P_Dri	37	13	1	29	25
PROTEIN_E_UVAOtx [†]	40	22	1	9	11	mRNA_P_Endo16 [†]	33	49	1	16	18
PROTEIN_E_cB [†]	3	17	1	3	5	mRNA_P_OrCt	31	25	1	21	24
PROTEIN_E_nBTCTF [†]	2	1	1	8	2	mRNA_P_UbiqEts1 [†]	24	51	1	30	21
PROTEIN_M_FoxN23	52	59	1	18	23						

Table A.5: Rank of nodes in test set of I₁.

Appendix B

Structural Features: Algorithm Analysis

In this appendix, we provide the time and space complexity analysis for algorithms used to obtain some of the structural features. Those analysis not provided here are covered in published literature in Table 5.4.

B.1 Degree Centrality

Theorem B.9 *Computation of degree centrality requires $O(|V|)$ time in the worst case.*

Proof: It takes $O(|V|)$ time to iterate through all the nodes in the graph since it requires constant time to retrieve the number of edges associated to each node.

Theorem B.10 *Computation of degree centrality requires $O(|V| + |E|)$ space in the worst case.*

Proof: $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. The computed degree centrality values require another $O(|V|)$ space. Taken together, a maximum of $O(|V| + |E|)$ space is needed.

B.2 Eigenvector Centrality

Theorem B.11 *Computation of eigenvector centrality requires $O(|V|^2 + |E|)$ space in the worst case.*

Proof: $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. Using Perron-Frobenius theorem to find the eigenvalue λ [258] necessitates the storage of the adjacency matrix which requires $O(|V|^2)$ space. The computed eigenvector centrality values require another $O(|V|)$ space. Taken together, a maximum of $O(|V|^2 + |E|)$ space is needed.

B.3 Closeness and Eccentricity Centralities

Theorem B.12 *Computation of closeness and eccentricity centralities require $O(|V|^2 + |E|)$ space in the worst case.*

Proof: For both centralities, $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. The shortest path distance are used in the definitions (Definitions 4.3 and 4.4) of both centralities. Using Dijkstra's algorithm [92], the shortest path computation requires $O(|V|^2)$ space for storage of the adjacency matrix. Another $O(|V|)$ space is required for the computed centrality values. Hence, a maximum of $O(|V|^2 + |E|)$ space is needed.

B.4 Bridging Coefficient

Theorem B.13 *Computation of bridging coefficient requires $O(|V|^2)$ time in the worst case.*

Proof: For each node in the network, the computation of the bridging coefficient iterates through all the neighbours of the node (Definition 4.6). In the worst case, the network is a single strongly connected component where every node is a neighbour of all other nodes. Hence, calculating the bridging coefficient of all nodes in the network requires $O(|V|^2)$ in the worst case.

Theorem B.14 *Computation of bridging coefficient requires $O(|V| + |E|)$ space in the worst case.*

Proof: $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. The computed bridging coefficient values require another $O(|V|)$ space. Taken together, a maximum of $O(|V| + |E|)$ space is needed.

B.5 Bridging Centrality

Theorem B.15 *Computation of bridging centrality requires $O(|V|^2)$ time in the worst case.*

Proof: For each node in the network, the computation of the bridging centrality is a product of the inverse of the betweenness rank and the bridging coefficient rank (Definition 4.7). Hence, computation requires $O(|V|^2 + |V| + |E|)$ time in the worst case. This can be further simplified into $O(|V|^2 + |E|)$.

Theorem B.16 *Computation of bridging centrality requires $O(|V| + |E|)$ space in the worst case.*

Proof: The space requirement for computing the bridging centrality is a sum of that for betweenness centrality and bridging coefficient. Hence, it is $O(2|V| + 2|E|)$ and can be simplified further to $O(|V| + |E|)$ space.

B.6 Clustering Coefficient

Theorem B.17 *Computation of clustering coefficient requires $O(|V| + |E|)$ space in the worst case.*

Proof: $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. According to Definition 4.8, computation of the clustering coefficient requires the use of the adjacency matrix which takes $O(|V|^2)$ space. The computed clustering coefficient values require another $O(|V|)$ space. Taken together, a maximum of $O(|V|^2 + |E|)$ space is needed.

B.7 Proximity Prestige

Theorem B.18 *Computation of proximity prestige requires $O(|V|^2 + |E|)$ time in the worst case.*

Proof: In Definition 4.9, the set of nodes having at least one path leading to node u (I_u) and the shortest path distance are needed for calculating the prestige value. Using the reachability-based approach described in Section 5.5.2 to obtain I_u requires $O(|V|^2 + |E|)$ time. space is required to store the node and edge information of the input graph. The shortest path distance can be found using Dijkstra's algorithm [92] and the computation requires $O(|E| + |V|\log_2|V|)$ time using Fibonacci heaps [117]. In the worst case, $O(|V|^2 + |E|)$ time is needed since $O(|V|^2) > O(|V|\log_2|V|)$.

Theorem B.19 *Computation of proximity prestige requires $O(|V| + |E|)$ space in the worst case.*

Proof: $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. The computed proximity prestige values require another $O(|V|)$ space. Taken together, a maximum of $O(|V| + |E|)$ space is needed.

B.8 Target Downstream Effect

Theorem B.20 *Computation of target downstream effect requires $O(|V|^2 + |E|)$ time in the worst case.*

Proof: According to Definition 4.10, the computation of the target downstream effect for each node requires iterating through each of its downstream nodes. The downstream nodes can be found by using the reachability-based approach described in Section 5.5.2. The time required to compute the reachability of the nodes is $O(|V|^2 + |E|)$. In the worst case, the

network is a single strongly connected component where every node has a path leading to all other nodes. Hence, calculating the target downstream effect of all nodes in the network requires $O(2|V|^2 + |E|)$ in the worst case and can be simplified to $O(|V|^2 + |E|)$.

Theorem B.21 *Computation of target downstream effect requires $O(|V| + |E|)$ space in the worst case.*

Proof: $O(|V| + |E|)$ space is required to store the node and edge information of the input graph. The computed target downstream effect values require another $O(|V|)$ space. Taken together, a maximum of $O(|V| + |E|)$ space is needed.

Appendix C

Best SVM Model Parameters

This appendix documents the best SVM model parameters of the various models that we experimented with. Table C.1 contains the parameters of the best models of different signaling networks using the TRENCENTO-naïve approach with different kernels; Tables C.2 and C.3 and contain the C parameters of the best models of the individual and the combined networks, respectively, using linear kernel. Table C.4 provides the best C parameters and other associated model evaluation criteria values associated with the best models of the combined network formed from all learning networks.

Best model Parameters	MAPK-PI3K	Ras Activation	Glucose-Stimulated Insulin Secretion	Endomesoderm Gene Regulatory	Glucose Metabolism
RBF Kernel					
Best model C	$2^{-0.4}$	2^2	2^{-10}	$2^{10.8}$	2^8
Best model γ	2^{-12}	2^{-10}	2^{-10}	2^{-12}	2^{-10}
Sigmoid Kernel					
Best model C	$2^{-0.4}$	2^2	2^{-10}	$2^{10.8}$	2^8
Best model γ	2^{-12}	2^{-10}	2^{-10}	2^{-12}	2^{-10}
Best model $Coef0$	2^{-12}	2^{-10}	2^{-10}	2^{-12}	2^{-10}
Polynomial Kernel					
Best model C	$2^{-0.4}$	2^2	2^{-10}	$2^{10.8}$	2^8
Best model γ	2^{-12}	2^{-10}	2^{-10}	2^{-12}	2^{-10}
Best model $Coef0$	2^{-12}	2^{-10}	2^{-10}	2^{-12}	2^{-10}
Best model $Degree$	2	2	2	2	2

Table C.1: Best model parameters for the various signaling networks using the TRENCENTO-naïve approach with different kernels. The parameters for the TRENCENTO-naïve approach with linear kernel is found in Table C.2.

Best model <i>C</i>	MAPK-PI3K	Ras Activation	Glucose-Stimulated Insulin Secretion	Endomesoderm Gene Regulation	Glucose Metabolism
TRENCENTO-W					
WMC Ratio ID=1	2^{-10}	$2^{4.08}$	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^{-10}	$2^{3.2}$	2^{-10}	2^8	2^{10}
WMC Ratio ID=3	2^{-10}	2^2	2^{-10}	$2^{3.2}$	2^{10}
WMC Ratio ID=4	2^0	$2^{2.08}$	2^{-10}	2^{-2}	2^{10}
WMC Ratio ID=5 (TRENCENTO-naïve)	$2^{-0.4}$	2^2	2^{-10}	$2^{10.8}$	2^8
WMC Ratio ID=6	$2^{-4.08}$	$2^{1.6}$	2^{-10}	$2^{2.8}$	$2^{5.6}$
WMC Ratio ID=7	$2^{-5.76}$	2^{-2}	2^{-10}	$2^{0.08}$	2^6
WMC Ratio ID=8	2^0	2^0	2^{-10}	$2^{5.6}$	$2^{6.8}$
WMC Ratio ID=9	$2^{0.8}$	2^2	2^6	2^8	2^8
TRENCENTO-WB					
WMC Ratio ID=1	$2^{7.28}$	$2^{5.2}$	2^{-10}	2^{-10}	$2^{8.4}$
WMC Ratio ID=2	$2^{6.4}$	$2^{4.08}$	2^{-10}	2^8	$2^{11.6}$
WMC Ratio ID=3	2^6	$2^{3.2}$	2^{-10}	$2^{0.16}$	$2^{10.8}$
WMC Ratio ID=4	$2^{2.4}$	$2^{1.84}$	2^{-10}	2^{-2}	$2^{10.4}$
WMC Ratio ID=5 (TRENCENTO-B)	2^4	$2^{3.2}$	2^{-10}	$2^{8.8}$	2^{12}
WMC Ratio ID=6	2^4	2^0	2^{-10}	2^6	$2^{10.4}$
WMC Ratio ID=7	2^2	2^0	2^{-10}	$2^{0.8}$	$2^{8.4}$
WMC Ratio ID=8	2^4	2^{-2}	2^{-10}	2^{10}	$2^{11.2}$
WMC Ratio ID=9	$2^{4.8}$	2^0	2^4	$2^{7.6}$	2^{10}
TRENCENTO-WR					
WMC Ratio ID=1	2^{-10}	2^4	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^{-10}	2^4	2^{-10}	2^6	2^{-10}
WMC Ratio ID=3	2^{-10}	2^{-10}	2^{-10}	$2^{7.52}$	2^{-10}
WMC Ratio ID=4	2^4	2^2	2^{-10}	$2^{3.6}$	2^{-10}
WMC Ratio ID=5 (TRENCENTO-R)	2^6	2^2	2^{-10}	$2^{8.4}$	2^{-10}
WMC Ratio ID=6	$2^{-3.6}$	2^{-10}	2^{-10}	$2^{-5.52}$	2^{-10}
WMC Ratio ID=7	$2^{-3.2}$	2^{-10}	2^0	$2^{-0.24}$	2^4
WMC Ratio ID=8	$2^{1.84}$	2^{-2}	2^{-10}	$2^{4.96}$	2^{-10}
WMC Ratio ID=9	2^{10}	2^2	2^8	2^6	2^{-10}
TRENCENTO-WH					
WMC Ratio ID=1	2^{-10}	2^6	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^6	2^4	2^{-10}	2^6	2^{-10}
WMC Ratio ID=3	2^4	2^4	2^{-10}	$2^{6.8}$	2^{-10}
WMC Ratio ID=4	$2^{8.08}$	2^{-10}	2^{-10}	$2^{2.4}$	2^{-10}
WMC Ratio ID=5 (TRENCENTO-H)	2^{10}	2^2	2^8	$2^{11.6}$	2^{-10}
WMC Ratio ID=6	2^{10}	2^{-10}	2^2	2^4	2^{-10}
WMC Ratio ID=7	2^{10}	2^{-10}	2^2	$2^{-1.76}$	2^4
WMC Ratio ID=8	2^{10}	2^{-2}	2^4	$2^{6.4}$	2^{-10}
WMC Ratio ID=9	2^8	2^4	2^{10}	2^4	2^{-10}

Table C.2: Best model *C* parameter for the various signaling networks using various approaches with linear kernel.

Best model C	C ₁	C ₂	C ₃	C ₄
TRENCENTO-W				
WMC Ratio ID=1	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=3	2^{-10}	2^{-10}	2^{-10}	2^2
WMC Ratio ID=4	$2^{0.4}$	$2^{-1.92}$	2^{-10}	$2^{9.2}$
WMC Ratio ID=5 (TRENCENTO-naïve)	$2^{10.4}$	$2^{5.6}$	2^{-10}	$2^{7.6}$
WMC Ratio ID=6	2^0	$2^{11.92}$	2^{-10}	$2^{1.04}$
WMC Ratio ID=7	$2^{-0.64}$	$2^{10.4}$	2^{-10}	$2^{-0.08}$
WMC Ratio ID=8	2^8	$2^{6.4}$	$2^{2.8}$	2^8
WMC Ratio ID=9	$2^{9.6}$	2^6	2^6	2^8
TRENCENTO-WB				
WMC Ratio ID=1	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=3	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=4	2^4	$2^{8.8}$	2^{-10}	$2^{3.6}$
WMC Ratio ID=5 (TRENCENTO-B)	$2^{9.2}$	$2^{5.36}$	2^{-10}	$2^{7.6}$
WMC Ratio ID=6	$2^{6.4}$	2^4	2^{-10}	$2^{-0.32}$
WMC Ratio ID=7	2^2	$2^{6.4}$	2^{-10}	$2^{10.8}$
WMC Ratio ID=8	$2^{4.4}$	$2^{9.6}$	2^{-10}	2^8
WMC Ratio ID=9	2^6	2^6	2^{-10}	$2^{7.6}$
TRENCENTO-WR				
WMC Ratio ID=1	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=3	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=4	2^4	2^{-10}	2^{-10}	2^8
WMC Ratio ID=5 (TRENCENTO-W)	2^0	2^6	2^{-10}	$2^{7.6}$
WMC Ratio ID=6	$2^{-0.16}$	$2^{-5.76}$	2^{-10}	2^{-2}
WMC Ratio ID=7	$2^{0.70}$	$2^{-3.6}$	2^{-10}	2^{10}
WMC Ratio ID=8	$2^{0.62}$	2^4	2^2	$2^{10.8}$
WMC Ratio ID=9	$2^{0.35}$	2^6	2^{-10}	$2^{4.8}$
TRENCENTO-WH				
WMC Ratio ID=1	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=2	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=3	2^{-10}	2^{-10}	2^{-10}	2^{-10}
WMC Ratio ID=4	2^{10}	2^{-10}	2^{-10}	$2^{4.4}$
WMC Ratio ID=5 (TRENCENTO-W)	2^6	2^8	2^{-10}	$2^{7.6}$
WMC Ratio ID=6	$2^{-2.56}$	$2^{-5.6}$	2^{-10}	$2^{-0.32}$
WMC Ratio ID=7	$2^{-1.36}$	$2^{-3.92}$	2^{-10}	2^0
WMC Ratio ID=8	$2^{2.4}$	2^0	2^2	2^8
WMC Ratio ID=9	2^{-10}	2^{-10}	2^{-10}	2^{12}

Table C.3: Best model C parameter for the various combined signaling network variants (C₁ to C₄) using different approaches with linear kernel.

	Best model C	$\bar{\phi}(val)$	$\phi(test)$	TPR	TNR	PPV
TRENCENTO-W						
WMC Ratio ID=1	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=2	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=3	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=4	$2^{1.84}$	0.699	0.848	0	0.951	0
WMC Ratio ID=5 (TRENCENTO-naïve)	2^8	0.734	0.783	0	0.878	0
WMC Ratio ID=6	$2^{11.2}$	0.75	0.739	0	0.829	0
WMC Ratio ID=7	$2^{1.28}$	0.732	0.674	0	0.756	0
WMC Ratio ID=8	2^{12}	0.605	0.587	0.6	0.585	0.15
WMC Ratio ID=9	2^{10}	0.493	0.435	1	0.366	0.161
TRENCENTO-WB						
WMC Ratio ID=1	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=2	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=3	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=4	$2^{2.8}$	0.693	0.87	0	0.976	0
WMC Ratio ID=5 (TRENCENTO-B)	$2^{10.4}$	0.754	0.804	0	0.902	0
WMC Ratio ID=6	$2^{2.4}$	0.759	0.696	0.2	0.756	0.091
WMC Ratio ID=7	$2^{1.12}$	0.723	0.609	0.2	0.659	0.067
WMC Ratio ID=8	$2^{4.4}$	0.546	0.435	0.6	0.415	0.111
WMC Ratio ID=9	2^6	0.349	0.109	1	0	0.109
TRENCENTO-WR						
WMC Ratio ID=1	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=2	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=3	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=4	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=5 (TRENCENTO-R)	2^8	0.72	0.783	0.6	0.805	0.273
WMC Ratio ID=6	2^4	0.734	0.565	0.6	0.561	0.143
WMC Ratio ID=7	$2^{-3.76}$	0.706	0.522	0.8	0.488	0.16
WMC Ratio ID=8	$2^{1.6}$	0.531	0.174	0.8	0.098	0.098
WMC Ratio ID=9	2^{-10}	0.32	0.109	1	0	0.109
TRENCENTO-WH						
WMC Ratio ID=1	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=2	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=3	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=4	2^{-10}	0.68	0.891	0	1	0
WMC Ratio ID=5 (TRENCENTO-H)	2^8	0.734	0.826	0.4	0.878	0.286
WMC Ratio ID=6	$2^{-0.8}$	0.733	0.565	0.6	0.561	0.143
WMC Ratio ID=7	$2^{0.96}$	0.725	0.326	0.6	0.293	0.094
WMC Ratio ID=8	2^8	0.571	0.217	0.6	0.171	0.081
WMC Ratio ID=9	2^{-10}	0.32	0.109	1	0	0.109

Table C.4: Best model C parameter and associated model evaluation criteria values for the combined network (formed from all learning networks) trained with a linear kernel.