Biological signal processing based on time-frequency distribution analysis

Yu, Renjun

2007

Yu, R. J. (2007). Biological signal processing based on time-frequency distribution analysis.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2495>

<https://doi.org/10.32657/10356/2495>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:24:21 SGT

Biological Signal Processing Based On Time-Frequency Distribution Analysis

by

Yu Renjun

Submitted in fulfilment of the requirement for the degree of
Doctor of Philosophy

at

School of Computer Engineering
Nanyang Technological University

September, 2006

ABSTRACT

The main objective here is to use the time-frequency techniques for the analysis of biomolecular sequences. The mathematical definitions of Cohen's class time-frequency distributions (CTFDs) are firstly reviewed. Some other useful time-frequency distributions (TFDs), such as the Short-time Fourier Transform (STFT), the wavelet transform (WT), the Choi-Williams Distribution (CWVD), the Wigner-Ville Distribution (WVD), the Smoothed Pseudo Wigner-Ville Distribution (SPWVD) and the Reassigned Smoothed Pseudo Wigner-Ville Distribution (RSPWVD), are introduced. Their performance in revealing the information of different types of signals and computational cost are evaluated. Among them the reassigned and smoothed TFDs (RSTFDs), such as the RSPWVD, can achieve a very satisfying compromise between the sharpness and the 'cleanness' (noise removal) measures of spectrum. However, the computational cost is very high and this makes the RSTFDs a less suitable class of candidates for high-speed applications. To overcome this problem, effective algorithms based on simplified SPWVD computation and recursive-windows are developed in this research. From the results of simulations, these new algorithms are able to reduce the computational cost of complicated RSPWVD algorithms greatly.

In current bioinformatics research, finding desired genes in the DNA sequences of various organisms and developing methods to predict the structure and/or function of newly discovered proteins are two major challenges. These DNA and protein sequence are digital in a real sense; they are represented in such a way that each element can be one out of a finite number of entities. In the case of DNA, the alphabet has a size of four and

consists of the letters A, T, C and G; in the case of proteins, the size of the corresponding alphabet is 20. So they can be digitalized into numerical sequences using corresponding methods just like discrete-time signal segments. Digital signal processing (DSP) techniques can thus be applied to discover useful information buried in these sequences. These digitalized signals are however not ‘natural’; they are nonstationary and have unexpected ‘noises’. So the time-frequency analysis can be an appropriate method to study these signals. As an effective tool of signal processing, the time-frequency analysis provides additional insight into signal representation and reveals information embedded in signals through its complex interaction between frequency and time. It describes how the frequency or spectral content of a signal evolves and to develop the physical and mathematical ideas needed to understand what a time-varying spectrum is.

In one of the investigations, the time-frequency analysis techniques for locating critical amino acid residues in protein sequences are explored. New time-frequency analysis tools based on different TFDs are developed. Protein signals based on amino acid hydrophobicity are constructed and analyzed with the newly developed time-frequency analysis tools. The effectiveness of using the time-frequency analysis techniques together with amino acid hydrophobicity in locating important amino acid domains in proteins is demonstrated; the amino acids critical to the function of these proteins can be clearly located in the resulting spectrograms. An evaluation of the performance of these different time-frequency analysis tools based on different TFDs is also made.

The time-frequency analysis techniques are also implemented to study DNA sequences in order to locate the exon and intron areas by their frequency properties. Tools based on the newly developed time-frequency analysis algorithms are designed for gene

identification. Performance results are compared and evaluated with those obtained from frequently used gene-finding tools such as the GeneMark.hmm, Genscan and HMMgene, using a dataset of HMD132. Combined programs of time-frequency analysis -based gene-finding tools and the HMMgene are developed for higher identification accuracy.

ACKNOWLEDGMENTS

I wish to express my sincere thanks and deep sense of gratitude to my supervisor, Dr. Tan Eng Chong, for his invaluable guidance, constant encouragement and keen interest in my work. I am extremely grateful to him for spending his valuable time with me for discussions in spite of his many other commitments and activities. He has been a great source of inspiration to work with and I shall always cherish my association with him with immense pleasure. His futuristic vision and realistic ideas have created a new enthusiasm and an ever-increasing zeal to work and explore many new things.

I am grateful to Nanyang Technological University for the financial support, which has enabled me to pursue this research.

I am also highly indebted to my parents for their constant encouragement and support in carrying out this research work.

Table of Contents

Abstract	i
Acknowledgements	iv
Table of Contents	v
List of Figures	ix
List of Tables	xiii
List of Acronyms and Abbreviations	xv
List of Symbols	xvii
List of Algorithms	xxi

Chapter 1: Introduction

1.1	The Need for Time-Frequency Analysis	1
1.2	The Time-Frequency Analysis in Bioinformatics Applications	3
1.3	Objectives	5
1.4	Organization of Contents	7

Chapter 2: Time-Frequency Analysis

2.1	Introduction	9
2.2	Time-Frequency Distributions	10
2.2.1	The Linear TFDs	11
2.2.2	Cohen's Class TFDs (CTFDs)	14
2.2.3	Affine Class TFDs and Hyperbolic Class TFDs	21
2.2.4	Adaptive TFDs	22
2.3	Reassigned Smoothed Pseudo WVD (RSPWVD)	25

Table of Contents

2.3.1	Factors Affecting Performance of TFDs	25
2.3.2	Smoothing Function for Reducing Cross-Terms	27
2.3.3	Reassigned Method for Improving Concentration	29
2.4	Summary	31

Chapter 3: Improved Algorithms for Time-Frequency Analysis

3.1	Introduction	33
3.2	Time-Recursive Discrete Fourier Transform (DFT) Algorithm	34
3.3	The RSPWD Algorithms	35
3.3.1	The RSPWVD Computation Based on Simplified SPWVD	35
3.3.2	The RSPWVD Computation Based on Time-Recursive Method	37
3.3.3	The Direct RSPWVD Algorithm	44
3.3.4	The Simplified RSPWVD Algorithm	44
3.3.5	The Time-Recursive RSPWVD Algorithm	47
3.4	Evaluation of RSPWVD Algorithms	49
3.5	Summary	56

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Tools

4.1	Introduction	57
4.2	Biological Background	58
4.2.1	From DNA to Gene	58
4.2.2	From Gene to Protein	60
4.2.3	Determining the Three-Dimensional Structure of Proteins	63
4.2.4	Hemoglobin	64
4.2.5	Myoglobin	65
4.2.6	Prolactin	67
4.2.7	Glucagon	67

Table of Contents

4.2.8	Cytochrome C	67
4.2.9	Lysozyme	68
4.2.10	Epidermal Growth Factor	68
4.3	Identification of Active Sites in Protein Sequences Using Time-Frequency Analysis	70
4.4	Results and Analysis	74
4.4.1	Human Hemoglobin	74
4.4.2	Myoglobin	78
4.4.3	Cytochrome C	79
4.4.4	Prolactin	81
4.4.5	Lysozyme	83
4.4.6	Epidermal Growth Factor	84
4.4.7	Glucagon	85
4.4.8	Comparison of Computational time	85
4.5	Quantitative Comparison of Results	88
4.6	Summary	91

Chapter 5: Computational Gene-Finding Using Time-Frequency Analysis

5.1	Introduction	92
5.1.1	Sequence Similarity Search Method	93
5.1.2	The Integrated Compositional and Signal Search Method	94
5.1.3	Gene-Finding Programs	95
5.2	Testing and Training Dataset and Gene-Finding Programs	96
5.2.1	Testing Dataset for All Programs	96
5.2.2	Training Dataset for Gene-Finding Using Time-Frequency Analysis	99
5.2.3	Tests of Gene-Finding Programs	100
5.3	Gene-Finding Programs Using Time-Frequency Analysis Techniques	102
5.3.1	Optimized Parameters Computation	102

Table of Contents

5.3.2	Identification of Gene Sites	107
5.3.3	Results Explanations	109
5.4	Measuring Predictive Accuracy	112
5.4.1	Nucleotide Level Accuracy	112
5.4.2	Exon Level Accuracy	113
5.5	Results and Evaluation	115
5.5.1	G+C Content	118
5.5.2	Exon Length	120
5.5.3	Exon Type and Signal Prediction	122
5.6	Combining Time-Frequency Analysis and Traditional Gene-Finding Programs	123
5.6.1	Motivation	123
5.6.2	Results and Evaluation	125
5.7	Summary	126
 Chapter 6: Conclusions and Future Work		
6.1	Conclusions	128
6.2	Future Work	130
 Appendix A: The HMD132 Dataset		133
 Appendix B: the THMD32 Dataset		137
 List of Author's Publications		139
 References		140

List of Figures

Figure 1.1	Time, frequency, and time-frequency representations of a low frequency signal followed by a high frequency signal.	4
Figure 1.2	Time, frequency, and time-frequency representations of a high frequency signal followed by a low frequency signal.	4
Figure 2.1	Classification of TFDs: STFT=Short-time Fourier transform, WT=Wavelet transform, CC=Cohen's Class, AC=Affine class, HC=Hyperbolic class, WVD=Wigner-Ville distribution, SP=Spectrograms, SC=Scalograms, RGK=Radially Gaussian kernel, LWVD=L-Wigner-Ville distributions, PWVD=polynomial Wigner-Ville distributions, WHOMS=Wigner higher order moment spectra.	11
Figure 2.2	Energy spectral density, real part and STFT of a 128 point truncated sinusoid, where n is the index of time samples.	12
Figure 2.3	Energy spectral density, real part and STFT of a 128 point truncated sinusoid with 64 point Hamming window, where n is the index of time samples.	13
Figure 2.4	Energy spectral density, real part and STFT of a 128 point truncated sinusoid with 7 point Hamming window, where n is the index of time samples.	13
Figure 2.5	WVD and IF of a linear FM signal: (a) WVD, (b) IF, where n is the index of time samples.	16
Figure 2.6	Illustration of cross terms in a monocomponent signal, where n is the index of time samples.	18
Figure 2.7	The interference geometry in the time-frequency plane.	18
Figure 2.8	Bat signal and its adaptive optimal kernel time-frequency representation: (a) a bat signal, (b) adaptive optimal kernel time-frequency representation of the signal.	24
Figure 2.9	STFT representation of the two-components signal, where n is the index of time samples.	26

List of Figures

Figure 2.10	WVD representation of the two-components signal, where n is the index of time samples.	26
Figure 2.11	SPWVD representation of the two-components signal, where n is the index of time samples.	29
Figure 2.12	RSPWVD representation of the two-components signal, where n is the index of time samples.	31
Figure 3.1	The WD representation of the signal.	51
Figure 3.2	The SPWD representation of the signal.	52
Figure 3.3	The RSPWVD representation of the signal based on original RSPWVD algorithm.	52
Figure 4.1	The structure of DNA, DNA is a double helix formed by base pairs attached to a sugar-phosphate backbone. (U.S. National Library of Medicine, http://ghr.nlm.nih.gov/handbook/illustrations/dnastructure)	60
Figure 4.2	Protein Synthesis. (http://www.emc.maricopa.edu/faculty/farabee/BIO-BK/BioBookglossR.html)	62
Figure 4.3	A generic α -amino acid, where the 'R' group is variable, and is the only difference between the 20 common amino acids. This form is called a zwitterion, because it has both positive and negatively charged atoms. The zwitterionic state results from the amine group (NH ₂) gaining a hydrogen atom from solution, and the acidic group (COO) losing one. (http://cnx.org/content/m11614/latest/)	63
Figure 4.4	Hemoglobin (http://fig.cox.miami.edu/~cmallery/150/protein/proteinsb.htm).	66
Figure 4.5	Myoglobin (http://courses.cm.utexas.edu/jrobertus/ch339k/overheads-1.htm).	66
Figure 4.6	Prolactin (http://en.wikipedia.org/wiki/Image:PRL_structure.png).	69
Figure 4.7	Glucagon (http://en.wikipedia.org/wiki/Image:Glucagon.png).	69
Figure 4.8	Cytochrome C (http://bai.nci.nih.gov/).	69

Figure 4.9	Human Lysozyme (http://commons.wikimedia.org/wiki/Image:Lysozyme.png).	70
Figure 4.10	Epidermal Growth Factor (http://www-nmr.cabm.rutgers.edu/photogallery/proteins/htm/page3.htm).	70
Figure 4.11	Protein sequence of human hemoglobin chains is transformed into a numerical sequence by assigning an EIIP value to each amino acid: (a) alpha chain, (b) beta chain.	73
Figure 4.12	Spectrogram of human hemoglobin alpha chain, where n is the index of amino acids.	76
Figure 4.13	RSPWVD of human hemoglobin alpha chain, where n is the index of amino acids.	76
Figure 4.14	Spectrogram of the beta chain of hemoglobin, where n is the index of amino acids.	77
Figure 4.15	RSPWVD of the beta chain of hemoglobin, where n is the index of amino acids.	77
Figure 4.16	Spectrogram of myoglobin, where n is the index of amino acids.	78
Figure 4.17	RSPWVD of myoglobin, where n is the index of amino acids.	79
Figure 4.18	Spectrogram of human cytochrome C, where n is the index of amino acids.	80
Figure 4.19	RSPWVD of human cytochrome C, where n is the index of amino acids.	80
Figure 4.20	Spectrogram of prolactin, where n is the index of amino acids.	82
Figure 4.21	RSPWVD of prolactin, where n is the index of amino acids.	82
Figure 4.22	Spectrogram of lysozyme, where n is the index of amino acids.	83
Figure 4.23	RSWVD of lysozyme, where n is the index of amino acids.	84
Figure 4.24	Spectrogram of human epidermal growth factor, where n is the index of amino acids.	86
Figure 4.25	RSPWVD of human epidermal growth factor, where n is the index of amino acids.	86

List of Figures

Figure 4.26	Spectrogram of glucagons, where n is the index of amino acids.	87
Figure 4.27	RSPWVD of glucagons, where n is the index of amino acids.	87
Figure 5.1	$ W ^2$ of the DNA sequence of AF099730 using STFT.	110
Figure 5.2	$ W ^2$ of the DNA sequence of AF099730 using WVD.	110
Figure 5.3	$ W ^2$ of the DNA sequence of AF099730 using RSPWVD.	110

List of Tables

Table 2.1	The relationship between the properties of Cohen's class TFDs and the properties of their kernels	21
Table 3.1	Rectangular windows, the number of sub-windows is 1.	38
Table 3.2	Hamming windows, the number of sub-windows is 3.	39
Table 3.3	Blackman windows, the number of sub-windows is 5.	39
Table 3.4	Half-Sine windows, the number of sub-windows is 2.	39
Table 3.5	Computational time of 5 algorithms with different window lengths using Rectangular window. (N , L is the length of time and frequency smoothing windows, respectively)	54
Table 3.6	Computational time of 5 algorithms with different window lengths using Hamming window. (N , M is the length of time and frequency smoothing windows, respectively)	54
Table 3.7	Computational time of 5 algorithms with different window lengths using Blackman window. (N , M is the length of time and frequency smoothing windows, respectively)	55
Table 3.8	Computational time of 5 algorithms with different window lengths using Half-Sine window. (N , M is the length of time and frequency smoothing windows, respectively)	55
Table 3.9	<i>ERSD</i> , efficiency ratio of the simplified RSPWVD algorithm against the direct RSPWVD algorithm with different window lengths of different windows. (N , M is the length of time and frequency smoothing windows, respectively)	55
Table 3.10	<i>ERRD</i> , efficiency ratio of the time-recursive RSPWVD algorithm against the direct RSPWVD algorithm with different window lengths of different windows. (N , M is the length of time and frequency smoothing windows, respectively)	56
Table 4.1	Electron-ion interaction pseudopotential (EIIP) of amino acids.	72
Table 4.2	Computational time of the protein analysis (using a 40-point Hamming window).	88

List of Tables

Table 4.3	The value of performance measures over the eight pairs of results from Figure 12-27.	90
Table 5.1	Mean optimized parameters for different TFDs of human and mouse sequences.	106
Table 5.2	Nucleotide level accuracy.	117
Table 5.3	Exon level accuracy.	117
Table 5.4	Nucleotide and Exon Level Accuracy versus G+C content	120
Table 5.5	Exon Accuracy versus exon length	121
Table 5.6	Exon accuracy versus exon type	122
Table 5.7	Results of the combined gene-finding programs	125

List of Acronyms and Abbreviations

Acronyms	Meaning
CTFD	Cohen's class time-frequency distribution
CWD	Choi-Williams distribution
DFT	Discrete Fourier transform
DNA	Deoxyribonucleic acid
EIIP	Electron-ion interaction pseudopotential
EST	Expressed sequence tag
FFT	Fast Fourier transform
GACF	Generalized autocorrelation function
HMM	Hidden Markov model
LWD	L-Wigner-Ville distribution
MAC	Multiply and accumulate
MAD	Maximum amplitude difference
mRNA	Messenger RNA
PLWD	polynomial Wigner-Ville distribution
RGK	Radially Gaussian kernel
RNA	Ribonucleic acid
RRM	Resonant Recognition Model
rRNA	Ribosomal RNA
RSPWVD	Reassigned smoothed pseudo Wigner-Ville distribution
RSTFD	Reassigned and smoothed time-frequency distribution
RWTFD	Running-windowed time-frequency distribution
SPWVD	Smoothed pseudo Wigner-Ville distribution
STFT	Short-time Fourier transform
TFD	Time-frequency distribution
tRNA	Transfer RNA
UTR	Untranslated region

List of Acronyms and Abbreviations

WBS	Wigner-bispectrum
WVD	Wigner-Ville distribution
WHOMS	Wigner higher order moment spectra

List of Symbols

Symbol	Meaning
$A(t, \omega)$	Affine class time-frequency distribution
AC	Approximate correlation
ACP	Average conditional probability
AE	The number of annotated exons
$C(t, \omega)$	Cohen's class time-frequency distribution
$C_{sc}(t, \omega)$	Scaled Cohen's class time-frequency distribution
CC	Correlation coefficient
CR_a	Proportion of annotated exons that are correctly predicted
CR_p	Proportion of predicted exons that are exactly correct
$CWD(t, \omega)$	CWD of a signal $s(t)$
$CWT_s^\psi(t, \omega)$	Wavelet transform of $s(t)$, where ψ is the mother wavelet
$E[k]$	Frequency energy of a certain TFD at frequency k
Ecp	Expectation operator
$ERRD$	Efficiency ratio of the time-recursive RSPWVD algorithm against the direct RSPWVD algorithm
$ERSD$	efficiency ratio of the simplified RSPWVD algorithm against the direct RSPWVD algorithm
ES_n	Exon level sensitivity
ES_p	Exon level specificity
$F(t)$	Mapping function indicating the signal s varies with time t
FN	The number of coding nucleotides predicted incorrectly as non-coding
FP	The number of non-coding nucleotides predicted incorrectly as coding
$g[p]$	Smoothing window in the time domain

$h[k]$	Smoothing window in the frequency domain
$H_h(t)$	Hyperbolic time shift operator
$L(t, \omega)$	Smoothing function
$L_{p(k)}$	Length of a exon, whose frequency energy peak is $p(k)$
M_{ad}	Maximum amplitude difference
ME	Proportion of missed exons
$\hat{m}_s[n, m]$	Frequency reassignment value
$\hat{n}_s[n, m]$	Time reassignment value
OL	Proportion of predicted exons that overlap actual exons
PC_a	Proportion of partially predicted annotated exons
PC_p	Proportion of predicted exons that are partially correct
PE	The number of predicted exons
$r[\tau_k^s : \tau_k^e]$	A segment beginning at τ_k^s and ending at τ_k^e , with τ_k^s included
$s(t)$	Signal at time t
$ s(t) ^2$	Energy or intensity per unit time at time t of signal
$ s(t) ^2 \Delta t$	Fractional energy in the time interval Δt at time t
$S(\omega)$	Fourier coefficient of $s(t)$ at frequency ω
$ S(\omega) ^2$	Energy or intensity per unit frequency at frequency ω (energy density spectrum)
$ S(\omega) ^2 \Delta \omega$	Fractional energy in the frequency interval $\Delta \omega$ at frequency ω
$s_{sc}(t)$	Scaled signal of $s(t)$
$S_{STFT}(t, \omega)$	STFT of $s(t)$
$s[n]$	Discrete time sequence
$S[k]$	Discrete FT of $s[n]$
$S_r[t, k]$	Real part of $S[k]$
$S_j[t, k]$	Imaginary part of $S[k]$

$s_r[n]$	Real part of $s[n]$
$s_j[n]$	Imaginary part of $s[n]$
$SC[k]$	Total spectral content of sequences at frequency k
$SPC_h(t, \omega)$	Spectrogram with the kernel being the AF of the analysis window h
$SPWD_{g,h}(t, \omega)$	SPWVD of $s(t)$, where g and h are two real even windows
$SPWD_s^{gh}[n, m]$	Discrete SPWVD of $s[n]$ with the autocorrelation-domain kernels $g[p]h[k]$
$SPWD_s^{Tgh}[n, m]$	Discrete SPWVD of $s[n]$, with the autocorrelation-domain kernels $pg[p]h[k]$
$SPWD_s^{gDh}[n, m]$	Discrete SPWVD of $s[n]$, with the autocorrelation-domain kernels $g[p]h'[k]$
std	Standard deviation
t	Time domain factor
Δt	Time interval
$t_s(\omega)$	Group delay $S(\omega)$
TE	True exons, the number of exactly predicted exons
TD	Total computation time using direct RSPWVD algorithm
TN	The number of non-coding nucleotides predicted correctly as non-coding
TP	The number of coding nucleotides predicted correctly as coding
TR	Total computation time using the time-recursive RSPWVD algorithm
TS	Total computation time using the simplified RSPWVD algorithm
WE	Proportion of wrong exons
$w[n]$	Window function
$W[n, k]$	Discrete WVD of $s[n]$
$W(t, \omega)$	WD of a signal $s(t)$
$W_{x+y}(t, \omega)$	Cross WVD of the two signals, $s_x(t)$ and $s_y(t)$
$SW(t, \omega)$	Smoothed WVD

List of Symbols

Y_n	Proportion of coding nucleotides that are correctly predicted as coding
Y_p	Proportion of nucleotides predicted as coding that are actually coding
ω	Frequency domain factor
$\omega_s(t)$	Instantaneous frequency of signal $s(t)$
$\Delta\omega$	Frequency interval
$\phi(\theta, \tau)$	Kernel of Cohen's class time-frequency distribution
θ	Threshold for MAD algorithm

List of Algorithms

Algorithm 3.1	Time-Recursive Discrete Fourier Transform (DFT) Algorithm	35
Algorithm 3.2	The Direct RSPWVD algorithm proposed by Auger and Flandrin (1995)	45
Algorithm 3.3	The Simplified RSPWVD algorithm	46
Algorithm 3.4	The Time-Recursive RSPWVD algorithm	47
Algorithm 4.1	The optimized parameters computation algorithm	103
Algorithm 4.2	The Peak-Searching Algorithm	108
Algorithm 4.3	The Frame-Merging Algorithm	109
Algorithm 4.4	The combined gene-finding algorithm	124

Chapter 1: Introduction

1.1 The Need for Time-Frequency Analysis

The most important and fundamental variables to describe a given signal in nature are time and frequency (John and Dimitris 1996). While the time domain function indicates how the amplitude of a signal changes over time, the frequency domain function shows how often such changes take place.

Obviously the description in time-domain is the most fundamental description of signals. Most signals appear in the forms of a temporal variable, such as electromagnetic field, pressure, and voltage changing with time. So mathematically a signal can be described as a function of one variable as

$$s(t) = F(t) \tag{1.1}$$

where t stands for the time domain factor, and the signal s varies with t through a mapping function $F(t)$. The square of the signal, $|s(t)|^2$, is called the energy density or the instantaneous power, and it indicates how the signal energy is distributed in the time domain. Let

$$|s(t)|^2 = \text{the energy or intensity per unit time at time } t \text{ of a signal } s.$$

$$|s(t)|^2 \Delta t = \text{the fractional energy of a signal } s \text{ in the time interval } \Delta t \text{ at time } t.$$

The most common representation of a signal in frequency domain is the Fourier transform (FT). The standard FT allows the decomposition of a signal into individual frequency components and establishes the relative intensity of each component. The Fourier expansion can be expressed in terms of sinusoids of different frequencies as

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega) e^{j\omega t} d\omega \quad (1.2)$$

where

$$S(\omega) = \int_{-\infty}^{+\infty} s(t) e^{-j\omega t} dt \quad (1.3)$$

$S(\omega)$ is the measure of the similarity between the signal $s(t)$ and the complex sinusoidal functions.

One of the most important features of the FT is that the basis function in (1.2) and the dual function in (1.3) have the same form $e^{j\omega t}$. Moreover, $e^{j\omega t}$ corresponds to an impulse at frequency ω . Hence, the ruler used to measure the frequency properties of a signal processes the finest tick marks. Consequently, $S(\omega)$, the projection of a signal on the basis function, precisely reflects the signal's behavior at frequency ω . The square of the FT, $|S(\omega)|^2$, is called power spectrum, which indicates how the signal energy is distributed in the frequency domain. Let

$$|S(\omega)|^2 = \text{energy per unit frequency at frequency } \omega \text{ (energy density spectrum).}$$

$$|S(\omega)|^2 \Delta\omega = \text{the fractional energy in the frequency interval } \Delta\omega \text{ at frequency } \omega.$$

However, neither the signal nor its FT indicates how the energy is distributed simultaneously in time and frequency. For example, consider the representations of two synthetic signals shown in Fig. 1.1 and 1.2. The two signals appear identical only if the time or the frequency energy distribution is considered. However, the two signals are clearly not identical from its time-frequency distribution (TFD). Hence, to distinguish such signals and to provide a more revealing picture of characteristics of the signal, a joint time-frequency representation is necessary. TFDs are two-dimensional functions that indicate the joint time-frequency energy content of a signal. They have been used in a wide range of

signals, including speech, music and other acoustic signals, biological signals, radar and sonar signals, and geographical signals (Cohen 1995, Qian and Chen 1996).

1.2 The Time-Frequency Analysis in Bioinformatics Applications

In bioinformatics research, finding genes in DNA sequences of various organisms and developing methods to predict the structure and/or function of newly discovered proteins are two challenging problems (Burset and Guigo 1996, Burge and Karlin 1997, Claverie 1997, Alberts *et al.* 1998, Cosic *et al.* 2000). These DNA and protein sequences are digital in a very real sense. They are represented in the form of sequences of which each element can be one out of a finite number of entities. In the case of DNA, the alphabet is size four and consists of the letters A, T, C and G. In the case of proteins, the size of the corresponding alphabet is 20 (Anthony *et al.* 1999, Anastassiou 2002). So they can be digitalized into numerical sequences using corresponding methods just like discrete-time signal segments. Digital signal processing techniques can be applied for discovering useful information buried in these sequences. These digitalized signals are not ‘natural’; they are nonstationary and have unexpected ‘noises’. So it is suitable to use time-frequency analysis to analyze these signals (Cosic *et al.* 2000, de Trad *et al.* 2001). As an effective tool of signal processing, the time-frequency analysis provides additional insight into signal analysis and reveals information embedded in signals through its complex interaction between frequency and time. It can help to develop tools for bioinformatics research.

Fig. 1.1 Time, frequency, and time-frequency representations of a low frequency signal followed by a high frequency signal.

Fig. 1.2 Time, frequency, and time-frequency representations of a high frequency signal followed by a low frequency signal.

1.3 Objectives

This project is to use the time-frequency techniques for the analysis of biomolecular sequences. Not only should the biomolecular sequences, which are represented by characters, be translated to digital sequence but also the time-frequency analysis techniques need to be altered to meet the requirement of large scale bioinformatics data processing. Furthermore, the digitalized biomolecular signals are not the ‘natural’ signals; special techniques should be developed for the analysis. Thus, the objectives are to

(A). Propose New Efficient Time-Frequency Analysis Algorithms

The time-frequency analysis generates much more data than the original signal, which means the computational and storage cost is much higher than that of the separate analysis in time space and frequency space. Furthermore, optimized TFDs, such as the Reassigned Smoothed Pseudo Wigner-Ville Distribution (RSPWVD), which give better time-frequency representation of signals, cause much higher computational and storage cost. It will be a critical problem preventing the time-frequency analysis techniques from bioinformatics applications, which always deal with large scale biomolecular data. To overcome this problem, more effective algorithms should be developed to reduce computational cost. Using running-windows, the time-recursive approach is used to develop the new time-frequency analysis algorithms. New algorithms to compute WV, RSPWV are proposed. All of them reduce the computational cost greatly. More details are given in Chapter 3.

(B). Develop Detection Programs Based on Time-Frequency Analysis for Predicting the Structure and/or Function of Newly Discovered Proteins

Proteins have a site on them that initiate, mediate or terminate a particular biological action. Cosic *et al.* (1991) has established the Resonant Recognition Model (RRM), which is a physicomathematical model for protein analysis. It has been shown that a particular function in a protein is represented by one RRM characteristic frequency that can be determined by Fourier analysis (Cosic *et al.* 1991, Pirogova and Cosic 2001). However, the FT only reveals the frequency properties. It is believed that the time-frequency analysis, which analyzes signal in time and frequency domain simultaneously, can reveal much more useful information. In this study the advantage of the RRM framework is taken and suitable TFDs are chosen to develop time-frequency analysis programs for proteins analysis. These programs are expected to add advantage of presenting information about space (time) that in this case is associated with corresponds to a particular amino acids location in a protein and the active amino acids contributing to the characteristic frequencies of the proteins are consequently identified. With this information, some of the most functionally important amino acids in the protein are believed to be detected. More details are given in Chapter 4.

(C). Develop detection programs based on time-frequency analysis for finding the genes in the DNA sequences of various organisms.

In this era of intensive genomic sequencing, when millions of bases of genomic DNA are sequenced daily in genome centers worldwide and the list of completely sequenced genomes from different organisms is growing rapidly, tools for interpreting the content of

these genomes are more important than ever. Tools such as GeneMark.hmm (Lukashin and Borodovski 1998), Genscan (Burge 1997) and HMMgene (Krogh 1997) have been developed in the past 10 years and their prediction accuracy is constantly improving. However, it's still far away from completely automatic gene discovery with 100% accuracy. Anastassiou (2002) indicates the DNA sequences can be view as character strings, and be digitalized into numerical sets. So that digital signal processing techniques can be applied for searching useful information buried in this sequence. As the time-frequency analysis can reveal much more information of signals, it is believed to be a promising tool for DNA sequences analysis. 'DNA Spectra' can be obtained by time-frequency analysis, and then it can be used for the identification of genes. More details are given in Chapter 5.

1.4 Organization of Contents

The remaining chapters are organized as follows:

Chapter 2 reviews existing methods of representing a signal in time and frequency, including the mathematical definitions and commonly classification of TFDs. Objective assessments of TFDs and optimized TFDs are discussed.

Chapter 3 presents some new efficient time-frequency analysis algorithms. It mainly envisages the usage of time-recursive approach in computing TFDs. Time-recursive algorithm in computing discrete Fourier transform (DFT) is addressed. Moreover, it deals with the recursive algorithm for computing RSPWVD using running-time windows.

Chapter 4 explores the time-frequency analysis techniques for locating critical amino acid residues in protein sequences. New time-frequency analysis tools based on STFT and

Chapter 1: Introduction

RSPWVD are developed. Protein signals are constructed based on amino acid hydrophobicity and analyzed for detecting important amino acid domains. The amino acids critical to the function of these proteins are located in the resulting spectrogram and scalogram.

Chapter 5 implements the time-frequency analysis techniques to perform analysis of DNA sequences and locate the exon and intron areas by their frequency properties. Tools based on the newly developed recursive time-frequency analysis algorithms are designed for exon classification. Furthermore, frequently used exon classification tools such as GenMark.hmm, Genscan and HMMgene are used for comparisons. A small dataset, HMD132, is used to evaluate the performance of these algorithms. Attempts to integrate predictions by combining both the classical exon classification tools and time-frequency analysis tools are made for higher prediction accuracy.

Chapter 6 concludes the main findings of the project and recommends possible future work.

Chapter 2: Time-Frequency Analysis

2.1 Introduction

The origin of time-frequency analysis goes back to 1940s when Gabor's (1946) instrumental work on signal representation based on elementary Gaussian elements for a proper description of the signal in combined time and frequency domains. Later, the work done by Wigner (1932) in the field of quantum mechanics had been applied to signal processing by Ville (1948). Page (1952) had developed the concept of instantaneous power spectrum as the rate of change of energy spectrum in the range $-\infty$ to T , where T is a time constant. Levin (1964) used the same definition for the segment T to ∞ , and defined a new function as the average of both the $-\infty$ to T and T to ∞ types of instantaneous power spectra. The time-frequency analysis of nonstationary process was considered by Flandrin and Martin (1985). In 1968, a fundamental result was published by Rihaczek (1968) that was stemmed from physical observations, now known as the Rihaczek distribution. Claasen and Mecklenbrauker (1980) gave a mathematical treatment to the TFDs and provided an insight into their properties. In particular, the Wigner-Ville distribution (WVD) was rather extensively investigated. Cohen (1989, 1995) employed characteristic functions and operator theory to derive a general class of TFDs. He showed that all the bilinear TFDs can be written in one general form which has been called Cohen's class. Adaptive signal representations have been given a lot of attention to overcome some of the difficulties associated with TFDs, such as the amount of cross terms and the

conflicting frequency and time resolutions (Qian and Chen 1997). These representations were mainly investigated by Jones and Baraniuk (1993), Choi and Williams (1989) and Jones and Parks (1992). Time-frequency reassignment was also introduced as a means of improving the readability of time-frequency representations (Kodera *et al.* 1978), and later it was generalized to any bilinear time frequency or time scale representation by Auger and Flandrin (1995).

2.2 Time-Frequency Distributions

The basic idea of time-frequency analysis is to devise a function that will describe the energy density of a signal simultaneously in time and frequency domain (Cohen 1989 and 1995, Qian and Chen 1996, Chen and Hao 1999). By using such a function, the density of frequency at a particular time and the fraction of the energy in a certain frequency can be calculated. The distribution is named as the time-frequency distribution (TFD).

The TFDs can be classified according to their properties. A general classification is shown in Fig. 2.1 (O'Neill 1997). Two usual types of TFDs are those that are linear or quadratic functions of the signal. Examples of linear TFDs are short-time Fourier transform (STFT) and wavelet transform. Examples of quadratic TFDs are the spectrogram, scalogram and WVD. The Cohen's class, the affine class and the hyperbolic class are three general classes of quadratic distributions. Besides the linear and quadratic TFDs, the adaptive TFDs, such as Radially Gaussian kernel (RGK), L-Wigner-Ville distributions (LWVD), polynomial Wigner-Ville distributions (PLWVD) and Wigner higher order moment spectra (WHOMS), give flexibility in analyzing the signal by depicting them in

domains other than time and frequency.

Fig. 2.1 Classification of TFDs: STFT=Short-time Fourier transform, WT=Wavelet transform, CC=Cohen's Class, AC=Affine class, HC=Hyperbolic class, WVD=Wigner-Ville distribution, SP=Spectrograms, SC=Scalograms, RGK=Radially Gaussian kernel, LWVD=L-Wigner-Ville distributions, PLWVD=polynomial Wigner-Ville distributions, WHOMS=Wigner higher order moment spectra.

2.2.1 The Linear TFDs

(A). Short-Time Fourier Transform

The short-time Fourier transform (STFT) is already known for years for describing any quasi-stationary signals such as speech. The FT does not explicitly show the time localization of the frequency components of a signal. So the time localization can be

obtained by suitably pre-windowing the signal $s(t)$ (Nawab and Quatieri 1988). The STFT is defined as (Qian and Chen 1997)

$$S_{STFT}(t, \omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(\tau) g(\tau - t) e^{-j\omega\tau} d\tau \quad (2.1)$$

It uses the product of a sinusoidal wave with a symmetric finite energy window function g . Multiplication by a relatively short window effectively suppresses the signal outside the neighborhood around the “analysis time”, t . The effect of varying the length of the window is shown in Figs. 2.2, 2.3 and 2.4. The localization and energy concentration of STFT depends only on the window and does not vary unless the window is changed.

Fig. 2.2 Energy spectral density, real part and STFT of a 128 point truncated sinusoid, where n is the index of time samples.

Fig. 2.3 Energy spectral density, real part and STFT of a 128 point truncated sinusoid with 64 point Hamming window, where n is the index of time samples.

Fig. 2.4 Energy spectral density, real part and STFT of a 128 point truncated sinusoid with 7 point Hamming window, where n is the index of time samples.

(B). Wavelet Transform

The continuous wavelet transform was developed as an alternative approach to the STFT to overcome the resolution problem (Qian and Chen 1997). The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the signal is multiplied with a function, ψ , similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and the wavelet transform:

- (i) The single peak of FT will be seen corresponding to a sinusoid.
- (ii) The width of the window of wavelet transform is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

The continuous wavelet transform of signal $s(t)$ is defined as

$$WT(t, \omega) = \frac{1}{\sqrt{|\omega|}} \int s(\tau) \psi^* \left(\frac{\tau - t}{\omega} \right) d\tau \quad (2.2)$$

It is a function of two variables, t and ω , which are the translation and scale parameters, respectively. ψ is the transforming function, and it is called the mother wavelet which is a zero-averaging function centered around zero with finite energy. A good review on wavelet transform, its applications, the filter bank interpretation and fast computations was made by Rao and Bopardikar (1998).

2.2.2 Cohen's Class TFDs (CTFDs)

To distribute the energy of the signal over time and frequency, Cohen (1995) proposed a unified approach as

$$C(t, \omega) = \frac{1}{4\pi^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-j\theta t - j\pi u + j\theta u} \phi(\theta, \tau) s^* \left(u - \frac{1}{2} \tau \right) s \left(u + \frac{1}{2} \tau \right) du d\tau d\theta \quad (2.3)$$

where $\phi(\theta, \tau)$ is an arbitrary function called the kernel by Classsen and Meclenbrauker (1980), and $C(t, \omega)$ is a Cohen's Class TFD (CTFD). The kernel can be a function of time and frequency. In general, it is preferred to be low pass in nature because it acts as a filtering means in the ambiguity function domain (Cohen 1995). The Kernel determines properties of the TFDs build upon them. Stated otherwise, the desired properties get reflected as constraints on the kernel. Some of the TFDs belonging to the Cohen's class are briefly reviewed in the following.

(A). Wigner-Ville Distribution

The Wigner-Ville Distribution (WVD) is a prototype of the distribution that is qualitatively different from spectrogram (magnitude squared of the STFT) (Cohen 1995, Qian and Chen 1996). The WVD has been successfully used in analyzing nonstationary signals (Cohen 1995). The WVD of a signal $s(t)$ is given by

$$WV(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} s \left(t + \frac{\tau}{2} \right) s^* \left(t - \frac{\tau}{2} \right) e^{-j\omega\tau} d\tau \quad (2.4)$$

(2.4) can be obtained by setting the kernel equal to one in (2.3). Most of the properties of WVD can be obtained with this interpretation. Perhaps the most remarkable property of the WVD is that for a Gaussian windowed linear chirp signal, defined as

$$s(t) = e^{-at^2} e^{j(a_0 + a_1 t + a_2 t^2)} \quad (2.5)$$

the energy of the signal is concentrated along the instantaneous frequency and its WVD is given by

$$WV(t, \omega) = e^{-(\omega - a_1 - 2a_2 t)^2} e^{-2a_2^2 t^2} \quad (2.6)$$

The distribution and the instantaneous frequency of the above signal are shown in Fig. 2.5.

Fig. 2.5 WVD and IF of a linear FM signal: (a) WVD, (b) IF, where n is the index of time samples.

The WVD of the sum of two signals, $s_x(t)$ and $s_y(t)$, is not the sum of individual WVDs. Instead, it is the sum of their WVDs plus another component that is the cross WVD of the two signals

$$WV_{x+y}(t, \omega) = WV_x + WV_y + 2\text{Re}[WV_{x,y}(t, \omega)] \quad (2.7)$$

where the $2\text{Re}[WV_{x,y}(t, \omega)]$ is commonly called a cross term, and $WV_{x,y}(t, \omega)$ is the

cross WVD of the two signals, $s_x(t)$ and $s_y(t)$. The $WV_{x,y}(t, \omega)$ is given by

$$WV_{x,y}(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} s_x\left(t + \frac{\tau}{2}\right) s_y^*\left(t - \frac{\tau}{2}\right) e^{-j\omega\tau} d\tau \quad (2.8)$$

All quadratic TFDs, including the spectrogram, will contain cross terms.

Cross terms can also occur within a single component signal, e.g., a non-linear frequency modulated signal which is shown in Fig. 2.6. An auto term is defined, rather vaguely, as parts of the WVD that corresponds to the true spectrum of the signal. The cross terms do not actually represent the energy of the signal and are hence undesirable (Bikdash and Yu 1993, Cohen 1995, Qian and Chen 1996). Suppose the two auto terms are separated in time by Δt and in frequency by Δf , as shown in Fig. 2.7, there will be a cross term centered between the two auto terms in the time-frequency plane and it oscillates in the time direction with a rate of Δf and in the frequency direction with a rate of Δt . Kernel can be used as a filtering means to suppress these cross terms at the expense of degradation in auto term resolution. Even with spurious cross terms, the WVD has often been used because of its capability to resolve multicomponent signals that are time-frequency disjoint. Of all the quadratic representations, WVD alone attains simultaneous resolution in time and frequency. Besides, it satisfies most of the properties of a TFD (Cohen 1995, Qian and Chen 1997). A more detailed discussion can be found in (Hlawatsch and Boudreaux-Bartles 1992).

Fig. 2.6 Illustration of cross terms in a monocomponent signal, where n is the index of time samples.

Fig. 2.7 The interference geometry in the time-frequency plane.

WVD equally weights all times of the future and future and past with respect to the center of the analysis time window. By contract, the Pseudo Wigner-Ville Distribution (PWVD) emphasizes the properties near the time of interest. A window function that is peaked around $\tau = 0$ is used to multiply the WVD to emphasize the signal around time t

(Cohen 1995). Therefore, the PWVD of signal $s(t)$ is defined as

$$PWV(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} h(\tau) s\left(t + \frac{\tau}{2}\right) s^*\left(t - \frac{\tau}{2}\right) e^{-j\omega\tau} d\tau \quad (2.9)$$

where h denotes a smoothing window.

(B). Choi-Williams Distribution

The associated inherent cross terms in all bilinear distributions are of major concern in spectral estimation and in multicomponent signal separation. There are kernels with filtering characteristics which can reduce these cross terms. However, only those kernels constrained to construct distributions of desired properties are of interest. In the ambiguity function domain, the locus of auto terms falls around the origin while those of cross terms lies away from the origin. Choi and Williams (1989) have identified this property of the ambiguity function and have chosen a kernel that has larger weights in the vicinity of the auto terms and smaller weights farther away from the origin in the ambiguity function domain; it has the form of

$$\phi(\theta, \tau) = e^{-\frac{\theta^2 \tau^2}{\sigma}} \quad (2.10)$$

Substituting (2.9) into (2.3), CWD can be obtained as (Choi and Williams 1989, Cohen 1995)

$$CWD(t, \omega) = \frac{1}{4\pi^{3/2}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\tau^2/\sigma}} \exp\left[-\frac{(u-t)^2}{4\tau^2/\sigma} - j\tau\omega\right] s\left(u + \frac{\tau}{2}\right) s^*\left(u - \frac{\tau}{2}\right) du d\tau \quad (2.11)$$

The capability of CWD in suppressing the cross terms without much degradation in auto term can be controlled by a proper choice of the kernel spread, σ . A resolution comparison of several distributions and a detailed discussion of CWD were made by Jones

and Parks (1992).

(C). Spectrogram

The classical definition of spectrogram can be considered as the squared magnitude of the STFT (Nawab and Quatieri 1988). However, the spectrogram can also be considered as a member of Cohen's class with the kernel being the ambiguity function of the analysis window, and it is given by

$$SPC(t, \omega) = \left| \int_{-\infty}^{+\infty} s(\tau) h(t - \tau) e^{-j\omega\tau} d\tau \right|^2 \quad (2.12)$$

Where h is a time-window.

The interesting property of the spectrogram is that, unlike other bilinear TFDs, it is always nonnegative. Apparently it seems that the spectrogram does not suffer from cross terms, but strictly speaking the cross terms in this representation exactly fall into the auto term region. Hence, the spectrogram can be considered as a smoothed version of the WVD to suppress cross terms. Unfortunately, as a natural consequence, it reduces the auto term resolution.

(D). Relationship between Cohen's Class TFDs and Their Kernels

The properties of Cohen's class TFDs are related to the properties of their kernels. Qian and Chen (1997) summarized the relationship in Table. 2.1. Many other properties, such as finite time support, strong time support and inversion, have been discussed with proofs by Giridhar (1998).

Table 2.1 The relationship between the properties of Cohen's class TFDs and the properties of their kernels

No.	Properties of CTFD	Properties of Kernel
1	Time-shift invariant	Independent of time variable t
2	Frequency-shift invariant	Independent of frequency variable ω
3	Realness	$\phi(\theta, \tau) = \phi(-\theta, -\tau)$
4	Time marginal	$\phi(\theta, 0) = 1$
5	Frequency marginal	$\phi(0, \tau) = 1$
6	Instantaneous frequency property	$\phi(\theta, 0) = 1$ and $\left. \frac{\partial}{\partial \tau} \phi(\theta, \tau) \right _{\tau=0} = 0$
7	Group delay property	$\phi(0, \tau) = 1$ and $\left. \frac{\partial}{\partial \theta} \phi(\theta, \tau) \right _{\theta=0} = 0$
8	Positivity	$\phi(\theta, \tau)$ is an ambiguity function or signal-dependent

2.2.3 Affine Class TFDs and Hyperbolic Class TFDs

The Cohen's class includes all the quadratic distributions that are invariant to shifts in time and frequency. An operator can be defined and distributions that are covariant to shifts on these operators can be constructed. When scale and time shifts are chosen as the operators, the affine class is obtained (O'Neil 1997). TFDs in the affine class can be computed through smoothing the WVD, as WVD is covariant to both shift in frequency and scale. For example, the affine class can be written as

$$A(t, \omega) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(t', \omega') \phi(\omega(t-t'), \omega'/\omega) dt' d\omega' \quad (2.13)$$

where $W(t, \omega)$ is the WVD of the signal, and ϕ is an affine smoothing (O'Neil 1997).

The relationship between the affine class and the wavelet transform is similar to the relationship between Cohen's class TFDs and the STFT. The scalogram is defined as the squared magnitude of the wavelet transform, and scalograms are a subset of the affine class (O'Neil 1997). Innumerable distributions, which fundamentally differ from Cohen's class TFDs, can be constructed using the same general structure by defining a new set of operators on which the class would be covariant.

Papandreou *et al.* (1993) have defined another class of quadratic class called the hyperbolic class. This class consists of all quadratic TFDs that are covariant to scales and hyperbolic shifts in time. The hyperbolic time shift operator is defined as

$$H_h(t) = e^{-jh \log t} s(t) \quad (2.14)$$

Where h determines the shape of the hyperbola, and is called as the hyperbolic parameter. The relationship between the scale and hyperbolic time-shift operators is identical to the relationship between the time-shift and frequency-shift operators (and different from the relationship between the time-shift and scale operators). More information on the existence and construction of arbitrary operators can be found in (Baraniuk and Jones 1996). Higher order distributions were proposed to match very specific signals that have nonlinear frequency modulation or polynomial instantaneous frequency, e.g., Wigner-bispectrum (WBS), polynomial Wigner-Ville distribution (PLWVD) and L-Wigner-Ville distribution (LWVD) (Stankovic 1994).

2.2.4 Adaptive TFDs

The TFDs in the Cohen's class can be obtained by convolving the WVD with the

kernel function. The purpose of this kernel is to filter out cross terms and maintain the resolution of the auto terms. Since the structure of the auto terms in the WVD changes with signals, the kernels that work well for one signal may not work well for other signals. A fixed kernel results in good performance for only certain configurations of ambiguity function auto terms and cross terms. Since the locations of the auto components and cross components depend on the signals to be analyzed, it is expected to obtain good performance for a broad class of signals by using only the signal dependent kernel. Jones and Baraniuk (1993) have proposed methods for creating signal adaptive kernels. The optimal kernel design is formulated in the ambiguity plane because of its property to distinguish the auto components and cross components, and can be formulated as

$$\max_{\phi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |A(\theta, \tau) \phi(\theta, \tau)|^2 d\theta d\tau \quad (2.15)$$

which is subject to the constraints: $\phi(0,0)=1$ and $\phi(\theta, \tau)$ is radially non-increasing. TFD of a bat signal using the adaptive optimal kernel design is shown in Fig. 2.8. A fast algorithm to compute the above representation is proposed in (Jones and Baraniuk 1994).

Further, constraining the kernel to be a radially Gaussian kernel is proposed in (Jones and Baraniuk 1993). Jeong and Williams (1992) have investigated the kernel design for reduced interference. Jones and Parks (1992) have considered the problem of finding a best estimate using STFT by choosing a concentration measure and then adapting the window parameter in each time and frequency bin using this measure. They claim that it outperforms all representations in terms of concentration, but it is computationally very intensive.

(a)

(b)

Fig. 2.8 Bat signal and its adaptive optimal kernel time-frequency representation: (a) a bat signal, (b) adaptive optimal kernel time-frequency representation of the signal. (Jones and Baraniuk 1994)

2.3 Reassigned Smoothed Pseudo WVD (RSPWVD)

2.3.1 Factors Affecting Performance of TFDs

Just like some spectral estimates are better than others, some TFDs also outperform others when they are used to analyze certain classes of signals (Choi and Williams 1989, Cohen 1989 and 1995, Zaveri *et al.* 1992, Andria and Savino 1994, Qian and Chen 1996, Chen and Hao 1999). The WVD is optimal for monocomponent signals since it achieves the best energy concentration around the signal. The STFT, on the other hand, results in an undesirable smoothing of the signal energy around its instantaneous frequency (Cohen 1989 and 1995, Qian and Chen 1996). Choosing the right TFD to analyze a given signal is not straightforward, even for monocomponent signals. However, the task appears to be more complex when one deals with multicomponent signals.

As an illustration, let us consider a multicomponent signal that contains two linear frequency modulations in different time and frequency. This signal is represented in the time-frequency domain using the STFT and WVD in Figs. 2.9 and 2.10. One can easily note that the result of STFT in Fig. 2.9 is ‘cleaner’ than that of the WVD in Fig. 2.10. There are no cross terms in the STFT. The cross terms in the middle of the two components of the WVD results look like a third component in the time-frequency plane, and they are likely to lead to error for signal identification. However, the STFT also has its own shortcoming. The two components spread out and occupy much more area in the time-frequency plane. So the result of STFT cannot reveal the properties of the signal accurately. Thus, it indicates two main factors affecting the performance of TFDs. The concentration of a TFD determines the resolution of the representation and the cross terms of a TFD decide the

‘cleanness’ of the representation.

Fig. 2.9 STFT representation of the two-components signal, where n is the index of time samples.

Fig. 2.10 WVD representation of the two-components signal, where n is the index of time samples.

2.3.2 Smoothing Function for Reducing Cross-Terms

One practical attempt to reduce the effect of the cross terms is to use smoothing functions (Andria and Savino 1994, Cohen 1995, Chen and Hao 1999). Here the WVD is used to illustrate the principle of smoothing function. The fundamental idea is to smooth the WVD by double convolution so that

$$W_s(t, \omega) = \int_{-\infty}^{+\infty} L(t-t', \omega-\omega') W(t', \omega') dt' d\omega' \quad (2.15)$$

where L is a smoothing function. A judicious choice of L will result in a new distribution with desirable properties. If L is taken to be independent of the signal, then the only way to obtain a positive distribution is by sacrificing the marginals. The most common smoothing function used is a Gaussian function (Cohen 1995) defined as

$$L(t, \omega) = \frac{1}{\alpha\beta} e^{-t^2/\alpha - \omega^2/\beta} \quad (2.16)$$

where α and β are two parameters of the Gaussian function. It has been known for a long time in literature (Debruijn 1967) that for certain values of α and β , a positive distribution is obtained. The condition is

$$\alpha\beta \geq 1 \quad (2.17)$$

The smoothed pseudo Wigner Ville distribution (SPWVD) applies a bidimensional filter to WVD. Its separable kernel allows the time and frequency smoothings to be adjusted independently. Let $H(\Omega) = \int_{-\infty}^{+\infty} h(\tau) d\tau$, $G(\Theta) = \int_{-\infty}^{+\infty} g(u) du$ and $\phi(u, \Omega) = g(u)H(\Omega)$. Thus,

$$SPWV(t, \omega) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(u) h(\tau) s\left(t - u + \frac{\tau}{2}\right) s^*\left(t - u - \frac{\tau}{2}\right) e^{-j\omega\tau} du d\tau \quad (2.18)$$

where g and h are two real even windows with $h(0) = G(0) = 1$ (Auger and Flandrin

1995).

SPWVD can be used to reduce the cross terms in WVD. For practical usage the discrete form of the SPWVD is used (Andria and Savino 1996, Bernasconi *et al.* 2004); i.e.,

$$SPWV^{gh}[n,m] = 2 \sum_{k=-N+1}^{N-1} \sum_{p=-M+1}^{M-1} g[p]h[k]s[n+p+k]s^*[n+p-k]e^{-2j\pi km/N} \quad (2.19)$$

where $h[k]$ is the smoothing window in the frequency domain and $g[p]$ is the smoothing window in the time domain. Generally the window length for $h[k]$ is $2N-1$ and is therefore reasonably large, and the smoothing window length for $g[p]$ is $2M-1$ and is generally of short duration, because if it is too large it smoothes too much of the required SPWVD signal. As seen from (2.19), the choice of the two windows can be done independently, thus enabling an efficient search for the best resolution both in time and frequency domains. This property makes SPWVD particularly suited to analyze large-scale nonstationary signals with multicomponent, when compared with the classic linear TFDs such as STFT and wavelet transform, and other quadratic TFDs, such as the WVD and the CWD. In particular, the STFT enables the choice of a time window only; therefore, the frequency resolution is inversely proportional to time resolution over time-frequency plane. Among the TFDs in the Cohen's class, as before mentioned, the WVD presents heavy cross terms. The majority of the smoothed TFDs, such as the Margenau-Hill and CWD, do not have a separable kernel, thus forcing a tradeoff search between time and frequency resolution (Auger and Flandrin 1995, Cohen 1995).

The signal in Figs. 2.9 and 2.10 can be represented by the SPWVD in Fig. 2.11. The removal of cross terms effect from the WVD is obvious. However, the two signal

components are wider than those of WVD. This means that the result is not as accurate as that of WVD. In other words, although cross terms are eliminated, accuracy of WVD declines.

Fig. 2.11 SPWVD representation of the two-components signal, where n is the index of time samples.

2.3.3 Reassigned Method for Improving Concentration

Although the SPWVD can achieve cross term degradation in the time-frequency plane, it also brings a less accurate localization of the signal auto-components. Auger and Flandrin (1995) showed that the reassignment method could be advantageously applied to certain CTFDs (particularly to SPWVDs) by careful choice of the given distribution's parameters, to increase signal concentration. So the following expressions are proposed to compute relocation positions $(\hat{n}, \hat{m})$ as a function of the time-frequency coordinates $[n, m]$ of RSPWVD:

$$\hat{n}[n, m] = n - \frac{SPWV^{Tgh}[n, m]}{SPWV^{gh}[n, m]} \quad (2.20)$$

$$\hat{m}[n, m] = m + \frac{SPWV^{gDh}[n, m]}{SPWV^{gh}[n, m]} \quad (2.21)$$

where gh , Tgh and gDh represent the autocorrelation-domain kernels $g[p]h[k]$, $pg[p]h[k]$ and $g[p]h'[k]$ ($h'[k]$ is the derivative of $h[k]$ at k), respectively. So

$$SPWV^{Tgh}[n, m] = 2 \sum_{k=-N+1}^{N-1} \sum_{p=-M+1}^{M-1} pg[p]h[k]s[n+p+k]s^*[n+p-k]e^{-2j\pi km/N} \quad (2.22)$$

$$SPWV^{gDh}[n, m] = 2 \sum_{k=-N+1}^{N-1} \sum_{p=-M+1}^{M-1} g[p]h'[k]s[n+p+k]s^*[n+p-k]e^{-2j\pi km/N} \quad (2.23)$$

The reassigned method creates a modified version of the time-frequency representation by moving the representation values away from where they are computed. It provides a higher concentration of TFDs in the time-frequency plane. Applying the reassigned method, the properties of signal are reserved and its TFDs are perfectly localized in time-frequency plane. The reassigned method can be associated with a properly chosen smoothing function to yield simultaneously a high concentration of signal components and cross term removal. If the chosen smoothing function of a representation fits the analyzed signal, its modified version will be more readable. The reassigned method is combined with the smoothing function used in Fig. 2.11 to produce the RSPWVD. The RSPWVD is used to analyze the same signal represented in Fig. 2.12. It is obvious that both the concentration of signal components and the removal of cross term are satisfied.

Fig. 2.12 RSPWVD representation of the two-components signal, where n is the index of time samples.

2.4 Summary

In this chapter a brief introduction of the principle of time-frequency analysis including its common theory and its operator algebra is given. A general classification of the TFDs according to their properties is introduced. Linear and quadratic TFDs are two usual types of TFDs. Examples of both kinds of the TFDs, such as the STFT and wavelet transform for linear TFDs and spectrogram, scalogram and WVD for quadratic TFDs, are given. Three general classes of quadratic distributions, the Cohen's class, the affine class and the hyperbolic class, are discussed.

Facts for the objective assessment of the performance of TFDs are evaluated. Attempts to improve the performance of TFD are discussed. The concentration of distributions and

Chapter 2: Time-Frequency Analysis

the effect of cross terms are the two main factors affecting the performance of TFDs. The smoothing function is introduced to reduce the effect of cross terms and the reassigned method is used to improve the concentration of certain TFDs. Combining these two techniques, a new distribution, RSPWVD, is obtained. The experiment results show the RSPWVD brings satisfying time-frequency representation of signals.

Chapter 3: Improved Algorithms for Time-Frequency Analysis

3.1 Introduction

As the time-frequency representation of a certain signal can be considered as a projection of the signal from two-dimension time-space into a three-dimension time-frequency space, it is obvious that the time-frequency analysis generates much more data than the original signal. This means that the computational and storage cost of time-frequency analysis is much higher than the separate analysis in time space and frequency space. Although the time-frequency analysis reveals desired useful information embedded in the original signal, its application in real-time cases will require more efficient algorithms to be developed.

Time-recursive approach has been developed for computing the discrete Fourier transform (Goertzel 1958, Bitmead 1982, Unser 1983, Zhang and Yao 1989) discrete cosine/ sine/ Hartley transforms (Liu and Chiu 1993), as well as the time-varying Fourier transform (Chen *et al.*). All of the time-recursive algorithms proposed in these papers share a common characteristic, i.e., when the moving-window moves forward one step, a new sample is included while excluding the oldest sample. So the time-recursive approach can be used to compute the TFDs using running-windows (Amin 1995 and 1996). It is similar to the multiply-and-accumulate (MAC) operation as in many digital signal processors. It does not decimate the sequence in either time or frequency but directly operates on the

Chapter 3: Improved Algorithms for Time-Frequency Analysis

sequence and acts as a mere summation involving multiplications. In the following, the time-recursive approach to compute the discrete Fourier transform (DFT) is reviewed and then the algorithms to compute the RSPWVD using the time-recursive approach are developed. The reassignment method of certain TFD, the RSPWVD (introduced in section 2.3, chapter 2) can greatly improve the sharpness of the location of the signal components by relocating its TFD in the time-frequency space. However, its computational cost is very high. New algorithms based on simplified SPWVD computation and the recursive method are developed and the computational cost of the RSPWVD is shown to reduce greatly when applied to a speech signal segment. Analysis on reduction of computational cost is also made.

3.2 Time-Recursive Discrete Fourier Transform (DFT) Algorithm

The DFT of a sequence, $s[n]$, is defined as (John and Dimitris, 1996)

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-j \frac{2\pi nk}{N}}, \quad k = 0, 1, \dots, N-1 \quad (3.1)$$

where N is the length of the sequence, $s[n]$.

(3.1) can be computed using time-recursive approach (Amin 1987). Denote $S[n', k]$ as the DFT of the sequence $s[N-1-n'], \dots, s[N-1]$, $n' = 0, 1, \dots, N-1$

$$S[n', k] = \sum_{m=0}^{n'} s[N-1-n'+m] e^{-j \frac{2\pi mk}{N}}, \quad k = 0, 1, \dots, N-1 \quad (3.2)$$

then

$$S[n'+1, k] = \sum_{m=0}^{n'+1} s[N-1-n'-1+m] e^{-j \frac{2\pi mk}{N}} \quad (3.3)$$

Let $m' = m - 1$. Then (3.3) can be rewritten as

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$$\begin{aligned}
 S[n' + 1, k] &= \sum_{m'=-1}^{n'} s[N - 1 - n' + m'] e^{-j \frac{2\pi(m'+1)k}{N}} \\
 &= \sum_{m'=0}^{n'} s[N - 1 - n' + m'] e^{-j \frac{2\pi(m'+1)k}{N}} + s[N - 2 - n']
 \end{aligned} \tag{3.4}$$

so

$$S[n' + 1, k] = e^{-j \frac{2\pi k}{N}} S[n', k] + s[N - 2 - n'] \tag{3.5}$$

$$S[N - 1, k] = S[k] \tag{3.6}$$

So it can be computed using the following algorithm:

Algorithm 3.1: Time-Recursive Discrete Fourier Transform (DFT) Algorithm

For each k ($0 \leq k \leq N - 1$)

Step (0): Initialize: Set $n' = -1$ and $S[-1, k] = 0$.

Step (1): Compute $S[n' + 1, k] = e^{-j \frac{2\pi k}{N}} S[n', k] + s[N - 2 - n']$, $n' \leq N - 1$.

Step (2): Assign $S[k] = S[N - 1, k]$.

Each element, $S[k]$ ($0 \leq k \leq N - 1$), of the DFT of the sequence $s[n]$ can be obtained after N such iterations.

3.3 The RSPWVD Algorithms

3.3.1 The RSPWVD Computation Based on Simplified SPWVD

In Chapter 2 the RSPWVD is proved to be a very satisfying solution for time-frequency representation of a certain signal. However, reassignment of SPWVD uses the current distribution value and requires the calculation of two additional SPWVDs. As a result the computational process of RSPWVD is very time-consuming. So some efficient algorithms should be developed to complete the reassignment operators. Based on the symmetry property of WVD algorithms, attempts can be made. Firstly (2.19) can be

rewritten as

$$\begin{aligned}
 SPWV^{gh}[n,m] &= 2 \sum_{p=-M+1}^{M-1} g[p] \sum_{k=1-N}^{N-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \\
 &= 2 \sum_{p=-M+1}^{M-1} g[p] \left[\sum_{k=1}^{N-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right. \\
 &\quad \left. + h[0] s[n+p] s^*[n+p] + 2 \sum_{k=1-N}^{-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right]
 \end{aligned} \tag{3.7}$$

Let $k' = -k$. Then (3.7) becomes

$$\begin{aligned}
 SPWV^{gh}[n,m] &= 2 \sum_{p=-M+1}^{M-1} g[p] \left[h[0] s[n+p] s^*[n+p] \right. \\
 &\quad \left. + \sum_{k=1}^{N-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right. \\
 &\quad \left. + \sum_{k'=1}^{N-1} h[-k'] s[n+p-k'] s^*[n+p+k'] e^{2j\pi km/N} \right]
 \end{aligned} \tag{3.8}$$

That is,

$$\begin{aligned}
 SPWV^{gh}[n,m] &= 2 \sum_{p=-M+1}^{M-1} g[p] h[0] s[n+p] s^*[n+p] \\
 &\quad + 2 \sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \\
 &\quad + 2 \sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} h[-k] s[n+p-k] s^*[n+p+k] e^{2j\pi km/N}
 \end{aligned} \tag{3.9}$$

For most practical proposes, the distribution needs to be real-valued, so the smooth function $h[m]$ should satisfy $h^*[m] = h[-m]$. Then (3.9) can be written as

$$\begin{aligned}
 SPWV^{gh}[n,m] &= 2 \sum_{p=-M+1}^{M-1} g[p] h[0] s[n+p] s^*[n+p] \\
 &\quad + 2 \sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \\
 &\quad + 2 \sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} \left[h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right]^* \\
 &= 4 \operatorname{Re} \left[\sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right] \\
 &\quad + 2 \sum_{p=-M+1}^{M-1} h[0] g[p] s[n+p] s^*[n+p]
 \end{aligned} \tag{3.10}$$

Similar decompositions as in (3.7)-(3.10) can be applied to (2.22) and (2.23). It can be

written after some manipulations as

$$SPWV^{Tgh}[n, m] = 4 \operatorname{Re} \left[\sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} pg[p]h[k]s[n+p+k]s^*[n+p-k]e^{-2j\pi km/N} \right] + 2 \sum_{p=-M+1}^{M-1} ph[0]g[p]s[n+p]s^*[n+p] \quad (3.11)$$

$$SPWV^{gDh}[n, m] = 4 \operatorname{Im} \left[\sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} g[p]h'[k]s[n+p+k]s^*[n+p-k]e^{-2j\pi km/N} \right] + 2 \sum_{p=-M+1}^{M-1} h'[0]g[p]s[n+p]s^*[n+p] \quad (3.12)$$

Comparing (3.9), (3.10) and (3.11) with (2.19), (2.22) and (2.23), the range of variable k changes from $-N+1 \leq k \leq N-1$ to $1 \leq k \leq N-1$, which corresponds to only about half the number of the original additions. So the computational cost is reduced.

3.3.2 The RSPWVD Computation Based on Time-Recursive Method

Amin (1995) introduces the running kernels that yield recursive structures for TFDs (TFDs). The running kernels offer important properties, which allow invariance in computations with respect to the extent of the kernel in the time or the lag variable. It is also shown that the commonly used smoothing windows, such as the Hamming window, the Blackman window and the Half-Sine window satisfy the desirable properties and simultaneously allow recursive implementations of the local autocorrelation (Amin 1996). Inspired by his work, the recursive method is believed to be efficient for reducing time when applied to SPWVD computation. After a straightforward manipulation, the recursive expression of (3.10) is given by

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$$SPWV^{gh}[n+1, m] = 4 \operatorname{Re} \left[\sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} g[p] h[k] s[n+1+p+k] s^*[n+1+p-k] e^{-2j\pi km/N} \right] + 2 \sum_{p=-M+1}^{M-1} h[0] g[p] s[n+1+p] s^*[n+1+p] \quad (3.13)$$

It can be rewritten as

$$SPWV^{gh}[n+1, m] = 4 \operatorname{Re} \left[\sum_{p=-M+2}^M \sum_{k=1}^{N-1} g[p-1] h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right] + 2 \sum_{p=-M+2}^M h[0] g[p-1] s[n+p] s^*[n+p] \quad (3.14)$$

Amin (1996) shows that the recursive method can be used if a certain smoothing window $g[p]$ belongs to the following window family:

$$G = \left\{ g \left| g[p] = \sum_{q=1}^Q g_q[p], \quad g_q[p] = \alpha_q \gamma_q^{-p} e^{-j\theta_q p} \right. \right\} \quad (3.15)$$

where $g_q[p]$ is the sub-window of $g[p]$, Q is the number of $g_q[p]$ and α_q and γ_q are complex parameters. The Rectangular, the Hamming, the Blackman and the Half-Sine windows belong to G . They can be decomposed into several sub-windows, $g_q[p]$, which satisfy the property of $g_q[p-1] = A_q g_q[p]$, where A_q is a complex constant (Amin 1996). Table 3.1-3.4 show the decomposition of the Rectangular, the Hamming, the Blackman and the Half-Sine windows.

Table 3.1 Rectangular windows, the number of sub-windows is 1.

Expression	$g[p] = \begin{cases} 1, & \text{for } 1 \leq p \leq N \\ 0, & \text{elsewhere} \end{cases}$
$g_q[p]$	$g[p] = 1$
A_q	$A_1 = 1$

Chapter 3: Improved Algorithms for Time-Frequency Analysis

Table 3.2 Hamming windows, the number of sub-windows is 3.

Expression	$g(p) = \begin{cases} \alpha + (\alpha - 1)\cos\left[\frac{2\pi k}{N+1}\right], & \text{for } 1 \leq p \leq N \\ 0, & \text{elsewhere} \end{cases}$
$g_q[p]$	$g_1[p] = \alpha, g_2[p] = \frac{1}{2}(\alpha - 1)e^{j2\pi p/(N+1)}, g_3[p] = \frac{1}{2}(\alpha - 1)e^{-j2\pi p/(N+1)}$
A_q	$A_1 = 1, A_2 = e^{-j2\pi/(N+1)}, A_3 = e^{j2\pi/(N+1)}$

Table 3.3 Blackman windows, the number of sub-windows is 5.

Expression	$g[p] = \begin{cases} 0.42 - 0.5\cos\left[\frac{2\pi k}{N+1}\right] + 0.08\cos\left[\frac{4\pi k}{N+1}\right], & \text{for } 1 \leq p \leq N \\ 0, & \text{elsewhere} \end{cases}$
$g_q[p]$	$g_1[p] = 0.42, g_2[p] = -0.25e^{j2\pi p/(N+1)}, g_3[p] = -0.25e^{-j2\pi p/(N+1)},$ $g_4[p] = 0.04e^{j4\pi p/(N+1)}, g_5[p] = 0.04e^{-j4\pi p/(N+1)}$
A_q	$A_1 = 1, A_2 = e^{-j2\pi/(N+1)}, A_3 = e^{j2\pi/(N+1)}, A_4 = e^{-j4\pi/(N+1)}, A_5 = e^{j4\pi/(N+1)}$

Table 3.4 Half-Sine windows, the number of sub-windows is 2.

Expression	$g[p] = \begin{cases} \sin\left[\frac{\pi k}{N+1}\right], & \text{for } 1 \leq p \leq N \\ 0, & \text{elsewhere} \end{cases}$
$g_q[p]$	$g_1[p] = 0.5je^{j\pi p/(N+1)}, g_2[p] = -0.5e^{-j\pi p/(N+1)}$
A_q	$A_1 = e^{-j\pi/(N+1)}, A_2 = e^{j\pi/(N+1)}$

Thus, the SPWVD computation can be executed in sub-windows as

$$SPWV_q^{gh}[n, m] = 2 \sum_{k=-N+1}^{N-1} \sum_{p=-M+1}^{M-1} g_q[p] h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \quad (3.16)$$

$$SPWV_q^{Tgh}[n, m] = 2 \sum_{k=-N+1}^{N-1} \sum_{p=-m+1}^{M-1} p g_q[p] h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \quad (3.17)$$

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$$SPWV^{gDh}_q[n, m] = 2 \sum_{k=-N+1}^{N-1} \sum_{p=-M+1}^{M-1} g_q[p] h'[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \quad (3.18)$$

Then the $SPWV^{gh}[n, m]$, $SPWV^{Tgh}[n, m]$ and $SPWV^{gDh}[n, m]$ can be computed as

$$SPWV^{gh}[n, m] = \sum_{q=1}^Q SPWV^{gh}_q[n, m] \quad (3.19)$$

$$SPWV^{Tgh}[n, m] = \sum_{q=1}^Q SPWV^{Tgh}_q[n, m] \quad (3.20)$$

$$SPWV^{gDh}[n, m] = \sum_{q=1}^Q SPWV^{gDh}_q[n, m] \quad (3.21)$$

Like (3.14), $SPWV^{gh}_q[n+1, m]$ can be written as

$$SPWV^{gh}_q[n+1, m] = 4A_q \operatorname{Re} \left[\sum_{p=-M+2}^M \sum_{k=1}^{N-1} g_q[p] h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right] \\ + 2A_q \sum_{p=-M+2}^M h[0] g_q[p] s[n+p] s^*[n+p] \quad (3.22)$$

$$SPWV^{gh}_q[n+1, m] = 4A_q \operatorname{Re} \left[\sum_{p=-M+1}^{M-1} \sum_{k=1}^{N-1} g_q[p] h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right] \\ + 4A_q \operatorname{Re} \left[\sum_{k=1}^{N-1} g_q[M] h[k] s[n+M+k] s^*[n+M-k] e^{-2j\pi km/N} \right] \\ - 4A_q \operatorname{Re} \left[\sum_{k=1}^{N-1} g_q[-M+2] h[k] s[n-M+2+k] s^*[n-M+2-k] e^{-2j\pi km/N} \right] \quad (3.23) \\ + 2A_q \sum_{p=-M+1}^{M-1} g[p] h[0] s[n+p] s^*[n+p] \\ + 2A_q g_q[M] h[0] s[n+M] s^*[n+M] \\ - 2A_q g_q[-M+2] h[0] s[n-M+2] s^*[n-M+2]$$

So

$$SPWV^{gh}_q[n+1, m] = A_q SPWV^{gh}_q[n, m] + \xi_q[n, m] \quad (3.24)$$

where

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$$\begin{aligned}
\xi_q[n, m] = & 4A_q \operatorname{Re} \left[g_q[M] \sum_{k=1}^{N-1} h[k] s[n+M+k] s^*[n+M-k] e^{-2j\pi km/N} \right. \\
& \left. - g_q[-M+2] \sum_{k=1}^{N-1} h[k] s[n-M+2+k] s^*[n-M+2-k] e^{-2j\pi km/N} \right] \\
& + 2A_q h[0] [g_q[L+1] s[n+M] s^*[n+M] \\
& - g_q[-M+2] s[n-M+2] s^*[n-M+2]]
\end{aligned} \quad (3.25)$$

Similarly, the recursive expression of (3.11) and (3.12) can be written as

$$\begin{aligned}
SPWV_q^{Tgh}[n+1, m] = & 4 \operatorname{Re} \left[\sum_{p=-M+2}^M \sum_{k=1}^{N-1} [p-1] g_q[p-1] h[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right. \\
& \left. + 2 \sum_{p=-M+2}^M [p-1] g_q[p-1] h[0] s[n+p] s^*[n+p] \right]
\end{aligned} \quad (3.26)$$

$$\begin{aligned}
SPWV_q^{gDh}[n+1, m] = & 4 \operatorname{Im} \left[\sum_{p=-M+2}^M \sum_{k=1}^{N-1} g_q[p-1] h'[k] s[n+p+k] s^*[n+p-k] e^{-2j\pi km/N} \right] \\
& + 2 \sum_{p=-M+2}^M g_q[p-1] h'[0] s[n+p] s^*[n+p]
\end{aligned} \quad (3.27)$$

After some manipulations, $SPWV_q^{Tgh}[n+1, m]$ and $SPWV_q^{gDh}[n+1, m]$ can be expressed

as

$$SPWV_{s=}^{Tgh}_q[n+1, m] = A_q [SPWV_q^{Tgh}[n, m] - SPWV_q^{gh}[n, m]] + \psi_q[n, m] \quad (3.28)$$

$$SPWV_q^{gDh}[n+1, m] = A_q SPWV_q^{gDh}[n, m] + \zeta_q[n, m] \quad (3.29)$$

where

$$\begin{aligned}
\psi_q[n, m] = & 4A_q [M-1] \operatorname{Re} \left[g_q[M] \sum_{k=1}^{N-1} h[k] s[n+M+k] s^*[n+M-k] e^{-2j\pi km/N} \right. \\
& \left. + g_q[-M+2] \sum_{k=1}^{N-1} h[k] s[n-M+2+k] s^*[n-M+2-k] e^{-2j\pi km/N} \right] \\
& + 2A_q [M-1] h[0] [g_q[M] s[n+M] s^*[n+M] \\
& + g_q[-M+2] s[n-M+2] s^*[n-M+2]]
\end{aligned} \quad (3.30)$$

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$$\begin{aligned}
\zeta_q[n, m] = & 4A_q \operatorname{Im} \left[g_q[M] \sum_{k=1}^{N-1} h'[k] s[n+M+k] s^*[n+M-k] e^{-2j\pi km/N} \right. \\
& - g_q[-M+2] \sum_{k=1}^{N-1} h'[k] s[n-M+2+k] s^*[n-M+2-k] e^{-2j\pi km/N} \left. \right] \\
& + 2A_q h'(0) [g_q[M] s[n+M] s^*[n+M] \\
& - g_q[-M+2] s[n-M+2] s^*[n-M+2]]
\end{aligned} \quad (3.31)$$

So $SPWV_q^{gh}[n+1, m]$ can be determined by the value of $SPWV_q^{gh}[n, m]$ and $\xi_q[n, m]$.

$SPWV_q^{Tgh}[n+1, m]$ can be determined by the value of $SPWV_q^{gh}[n, m]$, $SPWV_q^{Tgh}[n, m]$ and $\psi_q[n, m]$. $SPWV_q^{gDh}[n+1, m]$ can be determined by the value of $SPWV_q^{gDh}[n, m]$ and $\zeta_q[n, m]$.

For each frequency variable m , only $3Q$ initial SPWVDs, $SPWV_q^{gh}[0, m]$, $SPWV_q^{Tgh}[0, m]$ and $SPWV_q^{gDh}[0, m]$, ($1 \leq q \leq Q$), have to be computed using (3.10), (3.11) and (3.12). The following $SPWV_q^{gh}[n, m]$, $SPWV_q^{Tgh}[n, m]$ and $SPWV_q^{gDh}[n, m]$ with $n \geq 1$ can be obtained using the recursive method. Finally, $SPWV_q^{gh}[n, m]$, $SPWV_q^{Tgh}[n, m]$ and $SPWV_q^{gDh}[n, m]$ are obtained using (3.19), (3.20) and (3.21).

Let

$$\eta_{q,1}[n, m] = 4A_q g_q[M] \sum_{k=1}^{N-1} h[k] s[n+M+k] s^*[n+M-k] e^{-2j\pi km/N} \quad (3.32)$$

$$\eta_{q,2}[n, m] = 4A_q g_q[-M+2] \sum_{k=1}^{N-1} h[k] s[n-M+2+k] s^*[n-M+2-k] e^{-2j\pi km/N} \quad (3.33)$$

$$\delta_{q,1}[n, m] = 4A_q g_q[M] \sum_{k=1}^{N-1} h'[k] s[n+M+k] s^*[n+M-k] e^{-2j\pi km/N} \quad (3.34)$$

$$\delta_{q,2}[n, m] = 4A_q g_q[-M+2] \sum_{k=1}^{N-1} h'[k] s[n-M+2+k] s^*[n-M+2-k] e^{-2j\pi km/N} \quad (3.35)$$

$$\mu_{q,1}[n] = 2A_q g_q[M] s[n+M] s^*[n+M] \quad (3.36)$$

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$$\mu_{q,2}[n] = 2A_q g_q[-M+2]s[n-M+2]s^*[n-M+2] \quad (3.37)$$

Then, (3.25), (3.30) and (3.31) can be rewritten as

$$\xi_q[n, m] = \text{Re}(\eta_{q,1}[n, m] - \eta_{q,2}[n, m]) + h[0](\mu_{q,1}[n] - \mu_{q,2}[n]) \quad (3.38)$$

$$\psi_q[n, m] = (M-1)[\text{Re}(\eta_{q,1}[n, m] + \eta_{q,2}[n, m]) + h[0](\mu_{q,1}[n] + \mu_{q,2}[n])] \quad (3.39)$$

$$\zeta_q[n, m] = \text{Im}(\delta_{q,1}[n, m] - \delta_{q,2}[n, m]) + h'[0](\mu_{q,1}[n] + \mu_{q,2}[n]) \quad (3.40)$$

Thus,

$$\begin{aligned} SPWV_q^{gh}[n+1, m] &= A_q SPWV_q^{gh}[n, m] \\ &\quad + \text{Re}(\eta_{q,1}[n, m] - \eta_{q,2}[n, m]) \\ &\quad + h[0](\mu_{q,1}[n] - \mu_{q,2}[n]) \end{aligned} \quad (3.41)$$

$$\begin{aligned} SPWV_q^{Tgh}[n+1, m] &= A_q (SPWV_q^{Tgh}[n, m] - SPWV_q^{gh}[n, m]) \\ &\quad + (M-1)[\text{Re}(\eta_{q,1}[n, m] + \eta_{q,2}[n, m]) \\ &\quad + h(0)(\mu_{q,1}(n) + \mu_{q,2}(n))] \end{aligned} \quad (3.42)$$

$$\begin{aligned} SPWV_q^{gDh}[n+1, m] &= A_q SPWV_q^{gDh}[n, m] \\ &\quad + \text{Im}(\delta_{q,1}[n, m] - \delta_{q,2}[n, m]) \\ &\quad + h'[0](\mu_{q,1}[n] + \mu_{q,2}[n]) \end{aligned} \quad (3.43)$$

For further efficiency of computation, $\eta_{q,1}[n, m]$, $\eta_{q,2}[n, m]$, $\delta_{q,1}[n, m]$ and $\delta_{q,2}[n, m]$ can be computed using the fast Fourier transform (FFT) algorithm (Cunningham and Williams 1992). For $\eta_{q,1}[n, m]$, $\eta_{q,2}[n, m]$, $\delta_{q,1}[n, m]$ and $\delta_{q,2}[n, m]$, by changing only A_q and $g_q[p]$ at the last steps, only four FFT computations are needed for all the sub-windows. As shown in (3.41), (3.42) and (3.43), the $SPWV_q^{gh}[n, m]$, $SPWV_q^{Tgh}[n, m]$ and $SPWV_q^{gDh}[n, m]$ can be obtained more efficiently.. Finally, the time and frequency reassignment values, $\hat{n}_x[n, m]$ and $\hat{m}_x[n, m]$, can be computed using (2.20) and (2.21).

3.3.3 The Direct RSPWVD Algorithm

Auger and Flandrin (1995) proposed the direct RSPWVD algorithm given as Algorithm 3.2. The algorithm mainly requires three different SPWVDs and the RSPWVD values are stored in a matrix initially set to zero, where the SPWVD values are added as they are computed. At the end of the SPWVD computation, this matrix finally contains the RSPWVD.

Obviously, comparing with the SPWVD algorithm, the direct RSPWVD algorithm has to compute two more SPWVDs, reassign time and frequency values and add different SPWVDs with different frequency bins. It makes the computational cost of the direct RSPWVD algorithm about 10 times higher than that of the SPWVD algorithm.

3.3.4 The Simplified RSPWVD Algorithm

The direct RSPWVD algorithm can be modified to be more efficient. The new simplified RSPWVD algorithm is represented in Algorithm 3.3.

In Algorithm 3.3, (2.19), (2.22) and (2.23) of Algorithm 3.2 are substituted by (3.10), (3.11) and (3.12). So that about half of the original computation will be executed to obtain the $SPWV^{gh}[n,m]$, $SPWV^{Tgh}[n,m]$ and $SPWV^{gDh}[n,m]$. The position of the computation of $SPWV^{Tgh}[n,m]$ and $SPWV^{gDh}[n,m]$ in Algorithm 3.3 is also changed, they will be computed only when $SPWV^{gh}[n,m] > \varepsilon$, so that some unnecessary computation will be neglected.

Chapter 3: Improved Algorithms for Time-Frequency Analysis

Algorithm 3.2: The Direct RSPWVD algorithm proposed by Auger and Flandrin (1995)

For each k ($0 \leq k \leq N-1$)

Step (0): For a real signal $s_R[k]$, compute first its analytic signal $s[k]$.

Step (1): For a TFR on N_T analysis instants and N_{TFR} frequency bins, create a matrix RSPWVD of $N_T \times N_{TFR}$ elements, initialized to zero.

Step (2): For every analysis time n ($n = n_0 + i\Delta, i = 0, \dots, N_T - 1$) do

{

For every frequency bin m ($m = 0, \dots, N_{TFR} - 1$) do

{

Compute $SPWV^{gh}[n, m]$, $SPWV^{Tgh}[n, m]$ and $SPWV^{gDh}[n, m]$ based on (2.19), (2.22) and (2.23)

If ($SPWV^{gh}[n, m] > \varepsilon$) then

{

$$\hat{n}[n, m] = n - \text{Round} \left(\frac{1}{\Delta} \frac{SPWV^{Tgh}[n, m]}{SPWV^{gh}[n, m]} \right) \Delta$$

$$\hat{m}[n, m] = m + \text{Round} \left[\frac{N_{TFR}}{2\pi} \text{Im} \left(\frac{SPWV^{gDh}[n, m]}{SPWV^{gh}[n, m]} \right) \right]$$

$$RSPWV[\hat{n}, \hat{m}] = RSPWV[\hat{n}, \hat{m}] + SPWV^{gh}[n, m]$$

}

}

}

where $s_R[k]$ is real signal with N_z points, n_0 is the initial time, $s[k]$ is the analytic signal of $s_R[k]$, Δ is the time increment, $g[k]$ is the smoothing window of $2M-1$

points, $\frac{\pi}{N_{TFR}}$ is the frequency increment, $h[k]$ is the smoothing window of $2N-1$

points, and $h'[k] = Te \frac{dh}{dt}(kTe)$.

Chapter 3: Improved Algorithms for Time-Frequency Analysis

Algorithm 3.3: The Simplified RSPWVD algorithm

For each k ($0 \leq k \leq N-1$)

Step (0): For a real signal $s_R[k]$, compute first its analytic signal $s[k]$.

Step (1): For a TFR on N_T analysis instants and N_{TFR} frequency bins, create a matrix RSPWVD of $N_T \times N_{TFR}$ elements, initialized to zero.

Step (2): For every analysis time n ($n = n_0 + i\Delta, i = 0, \dots, N_T - 1$) do

{

For every frequency bin m ($m = 0, \dots, N_{TFR} - 1$) do

{

Compute $SPWV^{gh}[n, m]$ based on (3.10)

If ($SPWV^{gh}[n, m] > \varepsilon$) then

{

Compute $SPWV^{Tgh}[n, m]$ and $SPWV^{gDh}[n, m]$ based on (3.11) and (3.12),

$$\hat{n}[n, m] = n - \text{Round}\left(\frac{1}{\Delta} \frac{SPWV^{Tgh}[n, m]}{SPWV^{gh}[n, m]}\right)\Delta$$

$$\hat{m}[n, m] = m + \text{Round}\left[\frac{N_{TFR}}{2\pi} \text{Im}\left(\frac{SPWV^{gDh}[n, m]}{SPWV^{gh}[n, m]}\right)\right]$$

$$RSPWV[\hat{n}, \hat{m}] = RSPWV[\hat{n}, \hat{m}] + SPWV^{gh}[n, m]$$

}

}

}

where $s_R[k]$ is real signal with N_z points, n_0 is the initial time, $s[k]$ is the analytic

signal of $s_R[k]$, Δ is the time increment, $g[k]$ is the smoothing window of $2M-1$

points, $\frac{\pi}{N_{TFR}}$ is the frequency increment, $h[k]$ is the smoothing window of $2N-1$

points, and $h'[k] = Te \frac{dh}{dt}(kTe)$.

3.3. 5 The Time-Recursive RSPWVD Algorithm

The recursive approach may be used to compute the RSPWVD using Algorithm 3.4.

At first, the $SPWV^{gh}[n,m]$, $SPWV^{Tgh}[n,m]$ and $SPWV^{gDh}[n,m]$ are decomposed into the summations of $SPWV_q^{gh}[n,m]$, $SPWV_q^{Tgh}[n,m]$ and $SPWV_q^{gDh}[n,m]$ based on the properties of certain smoothing windows $g[p]$. Then $SPWV_q^{gh}[n,m]$, $SPWV_q^{Tgh}[n,m]$ and $SPWV_q^{gDh}[n,m]$ will be computed recursively. For every frequency bin m , three initial SPWVDs along time sequence, $SPWV_q^{gh}[n_0,m]$, $SPWV_q^{Tgh}[n_0,m]$ and $SPWV_q^{gDh}[n_0,m]$ are computed using the same idea of the simplified RSPWVD algorithm. For all $SPWV_q^{gh}[n,m]$, $SPWV_q^{Tgh}[n,m]$ and $SPWV_q^{gDh}[n,m]$, $n = n_0 + i\Delta, i = 1, \dots, N_T - 1$, these SPWVDs can be computed from (3.24), (3.28) and (3.29). Finally, $SPWV^{gh}[n,m]$, $SPWV^{Tgh}[n,m]$ and $SPWV^{gDh}[n,m]$ are calculated based on (3.19), (3.20) and (3.21) and the time and frequency reassignment values, $\hat{n}[n,m]$ and $\hat{m}[n,m]$, can be computed with little cost using (2.20) and (2.21). The condition of $SPWD_s^{gh}[n,m] > \varepsilon$ is also used to avoid unnecessary computation.

Algorithm 3.4: The Time-Recursive RSPWVD algorithm

For each k ($0 \leq k \leq N - 1$)

Step (0): For a real signal $s_R[k]$, compute first its analytic signal $s[k]$.

Step (1): For a TFR on N_T analysis instants and N_{TFR} frequency bins, create a matrix RSPWVD of $N_T \times N_{TFR}$ elements, initialized to zero.

Step (2): For every frequency bin m ($m = 0, \dots, N_{TFR} - 1$) do

{

 Compute of $SPWV^{gh}[n_0,m]$, $SPWV^{Tgh}[n_0,m]$ and

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$SPWD^{gDh}[n_0, m]$, based on (3.10), (3.11) and (3.12).

If ($SPWV^{gh}[n_0, m] > \varepsilon$) then

{

$$\hat{n}[n_0, m] = n_0 - \text{Round}\left(\frac{1}{\Delta} \frac{SPWV^{Tgh}[n_0, m]}{SPWV^{gh}[n_0, m]}\right)\Delta$$

$$\hat{m}[n_0, m] = m + \text{Round}\left[\frac{N_{TFR}}{2\pi} \text{Im}\left(\frac{SPWV^{gDh}[n_0, m]}{SPWV^{gh}[n_0, m]}\right)\right]$$

$$RSPWV[\hat{n}, \hat{m}] = RSPWV[n_0, m] + SPWV^{gh}[n_0, m]$$

}

For every analysis time n ($n = n_0 + i\Delta, i = 1, \dots, N_T - 1$) do

{

Compute $\eta_{q,1}[n-1, m]$, $\eta_{q,2}[n-1, m]$, $\mu_{q,1}[n-1, m]$ and $\mu_{q,2}[n-1, m]$ based on (3.32), (3.33), (3.36) and (3.37),

Compute $SPWV_q^{gh}[n, m]$ and $SPWV^{gh}[n, m]$ based on (3.41) and (3.19).

If ($SPWV^{gh}[n, m] > \varepsilon$) then

{

Compute $\delta_{q,1}[n, m]$ and $\delta_{q,2}[n, m]$ based on (3.34) and (3.35),

Compute $SPWV_q^{Tgh}[n, m]$ and $SPWV_q^{gDh}[n, m]$ based on (3.42) and (3.43),

Compute $SPWV^{Tgh}[n, m]$ and $SPWV^{gDh}[n, m]$ based on (3.20) and (3.21),

$$\hat{n}[n, m] = n - \text{Round}\left(\frac{1}{\Delta} \frac{SPWV^{Tgh}[n, m]}{SPWV^{gh}[n, m]}\right)\Delta$$

$$\hat{m}[n, m] = m + \text{Round}\left[\frac{N_{TFR}}{2\pi} \text{Im}\left(\frac{SPWV^{gDh}[n, m]}{SPWV^{gh}[n, m]}\right)\right]$$

$$RSPWV[\hat{n}, \hat{m}] = RSPWV[n, m] + SPWV^{gh}[n, m]$$

$$\left. \begin{array}{l} \{ \\ \{ \\ \{ \end{array} \right\}$$

where $s_R[k]$ is real signal with N_z points, n_o is the initial time, $s[k]$ is the analytic signal of $s_R[k]$, Δ is the time increment, $g[k]$ is the smoothing window of $2M-1$ points, $\frac{\pi}{N_{TFR}}$ is the frequency increment, $h[k]$ is the smoothing window of $2N-1$ points, and $h'[k] = Te \frac{dh}{dt}(kTe)$.

3.4 Evaluation of RSPWVD Algorithms

To evaluate the computational cost of the above three algorithms, we focus on the computation process of a single column of the $N_T \times N_{TFR}$ RSPWVD matrices. This column consists of N_{TFR} elements. Each element is determined by three corresponding elements in three $N_T \times N_{TFR}$ SPWVD matrixes, $SPWV^{gh}[n, m]$, $SPWV^{Tgh}[n, m]$ and $SPWV^{gDh}[n, m]$. However, for the direct RSPWVD and simplified RSPWVD algorithms, these SPWVD elements are computed separately. Comparing (2.19) with (3.10), (2.22) with (3.11) and (2.23) with (3.12), the simplified RSPWVD algorithm only needs about half of the computational cost to obtain these three SPWVDs. Surely, it will reduce the computation cost greatly. Furthermore, the condition of $SPWV^{gh}[n, m] > \varepsilon$ (ε is a threshold) in the simplified RSPWVD algorithm can avoid unnecessary computation which will be executed by the direct RSPWVD algorithm. Because in the simplified RSPWVD algorithm, the computation in the complicated reassignment stage will only be done where the distribution is enough to display.

The direct and simplified RSPWVD algorithms compute the elements of three

Chapter 3: Improved Algorithms for Time-Frequency Analysis

$N_T \times N_{TFR}$ SPWVD matrixes, $SPWV^{gh}[n,m]$, $SPWV^{Tgh}[n,m]$ and $SPWV^{gDh}[n,m]$ one by one. However, the time-recursive RSPWVD algorithm computes these elements column by column. But the time-recursive RSPWVD algorithm can be used only when $g[p]$ belongs to the window family G of (3.15). It can't be used if the smoothing windows don't belong to the window family G of (3.15). So only the simplified RSPWVD algorithm can be used to reduce the computational time of the RSPWVD.

For each column of these three $N_T \times N_{TFR}$ SPWVD matrices, only the first element is computed directly. The following elements are computed in different sub-window $g_q(p)$, ($1 \leq q \leq Q$). In the time-recursive RSPWVD algorithm, $SPWV_q^{gh}[n,m]$, $SPWV_q^{Tgh}[n,m]$ and $SPWV_q^{gDh}[n,m]$, $n = n_0 + i\Delta, i = 1, \dots, N_T - 1$ are computed based on (3.41), (3.42) and (3.43), and $SPWV^{gh}[n,m]$, $SPWV^{Tgh}[n,m]$ and $SPWV^{gDh}[n,m]$ are calculated based on (3.19), (3.20) and (3.21). $\mu_{q,1}[n]$ and $\mu_{q,2}[n]$, ($1 \leq q \leq Q$) are computed directly using (3.36) and (3.37). However, only $\eta_{1,1}[n,m]$, $\eta_{1,2}[n,m]$, $\delta_{1,1}[n,m]$ and $\delta_{1,2}[n,m]$ are computed using (3.32)-(3.35), and $\eta_{q,1}[n,m]$, $\eta_{q,2}[n,m]$, $\delta_{q,1}[n,m]$, $\delta_{q,2}[n,m]$, ($2 \leq q \leq Q$) can be computed by only changing the values of $A_q g_q[M]$ and $A_q g_q[-M+2]$. So during the whole process to compute three corresponding columns of SPWVD matrices using the time-recursive RSPWVD algorithm, only four FFTs of $M-1$ -element sequences are needed. As $Q \ll N_{TFR}$, the computation cost may be less than the other two algorithms. Both simplified and time-recursive RSPWVD algorithms have good performance in the reduction of computational cost.

In order to evaluate the performance of the existing and improved algorithms in practical applications, comparisons of their visual effects are made. The example is for

comparison is 120-point computer-generated signal made up of two chirps. Fig. 3.1 shows the WVD representation. The signal components are well localized but the various high amplitude oscillating cross-terms make it hardly readable. Fig. 3.2 shows the SPWVD representation with a 40-point Hamming window as the time-direction smoothing and the frequency-direction smoothing, respectively. There are very few cross-terms but concentration of the signal components is very weak. Most cross-terms are merely substantially attenuated by smoothing windows and the signal components are strongly localized by the reassignment method. From their mathematical definitions, the direct, simplified and time-recursive RSPWVD algorithms produce the same output; so only one figure is used. Fig. 3.3 shows the RSPWVD representation of the signal with the same smoothing windows in time and frequency directions as the SPWVD representation.

Fig. 3.1 the WVD representation of the signal.

Fig. 3.2 the SPWVD representation of the signal.

Fig. 3.3 The RSPWVD representation of the signal based on original RSPWVD algorithm.

Chapter 3: Improved Algorithms for Time-Frequency Analysis

For the whole $N_T \times N_{TFR}$ time-frequency domain, let

$$TD = \text{the total computation time using the direct RSPWVD algorithm} \quad (3.44)$$

$$TS = \text{the total computation time using the simplified RSPWVD algorithm} \quad (3.45)$$

$$TR = \text{the total computation time using the time-recursive RSPWVD algorithm} \quad (3.46)$$

$$ERSD = \frac{TS}{TD} \quad (3.47)$$

$$ERRD = \frac{TR}{TD} \quad (3.48)$$

Where $ERSD$ and $ERRD$ are efficiency ratios. They can show the improvement on computational time reduction by the simplified and time-recursive RSPWVD algorithms.

To evaluate the differences in computational time of the three different RSPWVD algorithms, a real speech signal sequence with a sampling rate of 8 kHz and 30,000 samples is used. Different smoothing windows with different lengths are employed. All five programs are executed on the same PC, and the computational times are recorded in Tables 3.5-3.8. The computational times of all the three programs using the direct, simplified and time-recursive RSPWVD algorithms are always much higher than those of the WVD and SPWVD programs. However, the computational time of RSPWVD can be reduced by using the new simplified and time-recursive RSPWVD algorithms. The program based on simplified algorithm only takes about 60% computational time of the program based on the direct RSPWVD algorithm. The computational time of the program based on the time-recursive RSPWVD algorithm is much less, only about 20% computational time of the direct RSPWVD algorithm.

Table 3.9 and 3.10 shows the efficiency ratios of the new simplified and

Chapter 3: Improved Algorithms for Time-Frequency Analysis

time-recursive RSPWVD algorithms. The $ERRD$ and $ERSD$ are affected by the type of smoothing window $g[p]$ and the length of time and frequency smoothing windows $g[p]$ and $h[k]$. The $ERRD$ also changes with the change of the Q value of $g[p]$. When $g[p]$ has to be decomposed into more sub-windows $g_q[p]$, the $ERRD$ increases, because more computation steps are needed for more sub-windows. However, the $ERSD$ remains almost the same.

Table 3.5 Computational time of 5 algorithms with different window lengths using Rectangular window. (N, L is the length of time and frequency smoothing windows, respectively)

Algorithm	Computational time (s)			
	$N=M=60$	$N=M=80$	$N=M=120$	$N=M=200$
WVD algorithm	12.340	14.133	16.891	26.123
SPWVD algorithm	11.234	13.212	15.652	24.431
Direct RSPWVD algorithm	89.291	143.312	180.661	307.621
Simplified RSPWVD algorithm.	58.921	88.901	110.431	187.231
Time-recursive RSPWVD algorithm	19.341	25.498	30.459	46.721

Table 3.6 Computational time of 5 algorithms with different window lengths using Hamming window. (N, M is the length of time and frequency smoothing windows, respectively)

Algorithm	Computation time(s)			
	$N=M=60$	$N=M=80$	$N=M=120$	$N=M=200$
WVD algorithm	17.015	19.516	22.502	33.597
SPWVD algorithm	12.859	17.094	19.515	29.596
Direct RSPWVD algorithm	104.811	165.972	205.994	337.472
Simplified RSPWVD algorithm.	67.092	101.821	129.287	211.902
Time-recursive RSPWVD algorithm	25.221	29.982	37.198	55.621

Chapter 3: Improved Algorithms for Time-Frequency Analysis

Table 3.7 Computational time of 5 algorithms with different window lengths using Blackman window. (N , M is the length of time and frequency smoothing windows, respectively)

Algorithm	Computational time (s)			
	$N=M=60$	$N=M=80$	$N=M=120$	$N=M=200$
WVD algorithm	19.431	21.872	25.942	38.052
SPWVD algorithm	14.892	19.924	23.015	34.024
Direct RSPWVD algorithm	120.721	181.201	228.104	385.903
Simplified RSPWVD algorithm.	74.110	113.031	140.093	243.021
Time-recursive RSPWVD algorithm	30.312	38.987	46.009	71.982

Table 3.8 Computational time of 5 algorithms with different window lengths using Half-Sine window. (N , M is the length of time and frequency smoothing windows, respectively)

Algorithm	Computational time (s)			
	$N=M=60$	$N=M=80$	$N=M=120$	$N=M=200$
WVD algorithm	15.402	18.035	20.987	29.431
SPWVD algorithm	10.991	16.990	18.604	26.082
Direct RSPWVD algorithm	98.439	149.621	189.320	316.362
Simplified RSPWVD algorithm.	62.387	92.765	119.082	196.821
Time-recursive RSPWVD algorithm	22.908	29.176	32.097	47.590

Table 3.9 *ERSD*, efficiency ratio of the simplified RSPWVD algorithm against the direct RSPWVD algorithm with different window lengths of different windows. (N , M is the length of time and frequency smoothing windows, respectively)

Window	Computational time (s)			
	$N=M=60$	$N=M=80$	$N=M=120$	$N=M=200$
Rectangular	0.65	0.62	0.61	0.61
Hamming	0.64	0.61	0.62	0.63
Blackman	0.61	0.62	0.61	0.63
Half-Sine	0.63	0.61	0.63	0.62

Chapter 3: Improved Algorithms for Time-Frequency Analysis

Table 3.10 *ERRD*, efficiency ratio of the time-recursive RSPWVD algorithm against the direct RSPWVD algorithm with different window lengths of different windows. (N, M is the length of time and frequency smoothing windows, respectively)

Window	Computational time (s)			
	$N=M=60$	$N=M=80$	$N=M=120$	$N=M=200$
Rectangular	0.22	0.18	0.17	0.15
Hamming	0.24	0.20	0.18	0.16
Blackman	0.25	0.22	0.20	0.19
Half-Sine	0.23	0.20	0.17	0.15

3.5 Summary

Two improved RSPWVD algorithms, the simplified and time-recursive RSPWVD algorithms, are developed. The simplified RSPWVD algorithm changes the computation of the SPWVDs in the direct RSPWVD algorithm. The time-recursive RSPWVD algorithm uses time-recursive approach in the computation of the SPWVDs. Both of these new algorithms are able to reduce the computational time of the RSPWVD greatly.

In most situations, when the time-recursive RSPWVD algorithm can reduce the computational time of the RSPWVD computation much greater than the simplified RSPWVD algorithm. However, the time-recursive RSPWVD algorithm is not suitable for those smoothing windows don't belong to the window family G of (3.15). The simplified RSPWVD algorithm can be used to reduce the computational time of the RSPWVD in those situations.

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

4.1 Introduction

The human genome research has released a wealth of information about the genetic code of humans. However much of the information is yet to be interpreted (Collins *et al.* 1998). The code contains instructions that make up three-dimensional proteins that constitute and maintain a human body. In order to affect a function on a cell, it is not uncommon for a protein to have to dock a specially shaped section of its three-dimensional structure into a specifically shaped receptor on a target cell. In short, many proteins have a site on them that initiate, mediate or terminate a particular biological action (Reece and Campbell 2005).

The genetic code is a sequence. One might wonder if the genetic code could be made amenable to time series analysis techniques. White *et al.* (1994) used signal-processing techniques to predict the protein tertiary structure from amino acid sequences. Peng *et al.* (1992) used random walks to study correlations in nucleotide sequences. Cosic *et al.* (1991) established the Resonant Recognition Model (RRM) which is a physicomathematical model for protein analysis.

The basis of the RRM is that there exists a significant correlation between the spectra of numerical representations of amino acids and their biological activity (Pirogova and Cosic 2001). More specifically, the biological function of a protein is characterized by

certain frequencies of its signal representation (de Trad C *et al.* 2000). The RRM analysis first involves converting amino acids that constitute a protein into a ‘discrete time series.’ The position of an amino acid in the sequence can be thought of as time. The datum associated with each time is hydrophobicity which is a measure of the tendency of an amino acid to avoid water (Pirogova and Cosic 2001). After the conversion of the amino acid sequence is interpreted into the protein time (space) series signal (called as ‘protein signal’) the signal is analyzed to locate the dominant frequencies. It has been shown that a particular function in a protein is represented by one RRM characteristic frequency that can be determined by Fourier analysis (de Trad C *et al.* 2000, Pirogova and Cosic 2001).

In this chapter the RRM framework and STFT and RSPWVD are used to analyze proteins. These TFDs are expected to add advantage of presenting information about space (time) that in this case is associated with corresponds to a particular amino acids location in a protein. Consequently, the active amino acids contributing to the characteristic frequencies of the proteins are able to be identified. With this information, some of the most functionally important amino acids in the protein are able to be determined (Essien *et al.* 2002).

4.2 Biological Background

4.2.1 From DNA to Gene

The human genetic code is found in deoxyribonucleic acid (DNA) which is organized into cellular structures called chromosomes. DNA is a double helix of nucleic acids (see Fig. 4.1). Each nucleic acid consists of units called nucleotides each of which is composed

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

of a nitrogenous base, a 5-carbon sugar and a phosphate group. There are four bases present in DNA – adenine (A), cytosine(C), thymine (T) and guanine (G). The deoxyribose sugar and the phosphate group of each nucleotide form the sugar-phosphate backbone of each strand of DNA. Within the backbone, the nucleotides are joined by covalent bonds between the phosphate group of one nucleotide and the deoxyribose sugar of another nucleotide (Raven and Johnson 2001).

Across the helix, nucleotides are linked by hydrogen bonds between the nitrogenous bases. In DNA, hydrogen bonding of bases produces only two combinations of amino acid pairs across the helix. Adenine must always be paired with thymine. Cytosine must always be paired with guanine. This specificity of the amino acid pairing is termed complementary base pairing (Reece and Campbell 2005).

The bases and the order in which they appear in a strand of DNA are of particular interest as they comprise the DNA sequence of a human. Within the DNA sequence are genes, which are specific sequences of bases that specify the information necessary for creating a particular protein. The human DNA sequence contains between 30000-40000 genes. Despite the high number of genes, only about 10% of genome consists of meaningful code or is expressed as proteins. These expressed regions are known as exons. The remaining 90% of the intervening regions of the genome (introns) are not useless but are believed to have regulatory roles in cells (Gilbert 1987).

Fig. 4.1 The structure of DNA, DNA is a double helix formed by base pairs attached to a sugar-phosphate backbone. (U.S. National Library of Medicine, <http://ghr.nlm.nih.gov/handbook/illustrations/dnastructure>)

4.2.2 From Gene to Protein

Ribonucleic acid (RNA) is another important nucleic acid found in living organisms. It is similar to deoxyribonucleic acid except that instead of a deoxyribose sugar, RNA contains a ribose sugar. There is also a difference in nitrogenous bases between RNA and DNA. As mentioned earlier, DNA has the bases adenine, thymine, guanine and cytosine. In DNA, adenine and thymine pair together in complementary fashion and so do cytosine and guanine. RNA on the other hand has the bases adenine (A), uracil (U), guanine (G) and cytosine (C). Uracil and adenine complement each other as do cytosine and guanine. There are three types of RNA involved in protein synthesis, namely, messenger RNA (mRNA) which is the template of constituents of a protein, ribosomal RNA (rRNA) which makes up the ribosome - the site of protein synthesis, and the transfer RNA (tRNA) which brings

amino acids to the ribosome (Reece and Campbell 2005).

Transcription is the first major part of protein synthesis. During transcription, a molecule called RNA polymerase parts of the DNA double helix in regions with genes. RNA polymerase then moves along one particular strand of the helix, known as the coding strand, and creates a strand of messenger RNA (mRNA) in complementary base pair fashion (an *A* in the DNA coding strand results in a *U* in mRNA, a *T* results in an *A* in mRNA, a *C* results in a *G* in mRNA, a *G* results in a *C* in mRNA). Fig. 4.2 shows a pictorial view of this process. As RNA polymerase moves along the coding strand, the transcribed portions of the coding strand become associated once more with their complementary strand of DNA.

After transcription, mRNA leaves the nucleus of the cell and begins sliding through an organelle known as the ribosome. Each triplet of bases on the strand of mRNA is called a codon as it codes for a particular amino acid. The second major step of protein synthesis is called translation and during this step, codons are translated into amino acids, which are the building blocks of amino acids. Fig. 4.3 shows the generic structure of an amino acid. There are 20 amino acids that make up proteins. Structurally each amino acid consists of a centrally located alpha carbon, an amino end, a carboxyl end and a side chain. The side chain (also called the R-group) is the variable part of the amino acid that distinguishes one amino acid from another.

Transfer-RNA (tRNA) transports specific amino acids from the cytoplasm to the ribosome. Each tRNA molecule has an anticodon that is complementary to a particular codon in mRNA. As the mRNA slides through the ribosome, a molecule of tRNA docks

into the ribosome with an anticodon complementary to the mRNA's codon. Once it happens the amino acid attached to the tRNA molecule is appended to the string of amino acids that are needed to synthesize the protein specified by the strand of mRNA. Such a string of amino acids is called a polypeptide chain and proteins consist of polypeptides in a particular conformation.

Fig. 4.2 Protein Synthesis. (<http://www.emc.maricopa.edu/faculty/farabee/BIO-BK/BioBookglossR.html>)

Fig. 4.3 A generic α -amino acid, where the 'R' group is variable, and is the only difference between the 20 common amino acids. This form is called a zwitterion, because it has both positive and negatively charged atoms. The zwitterionic state results from the amine group (NH_2) gaining a hydrogen atom from solution, and the acidic group (COO) losing one. (<http://cnx.org/content/m11614/latest/>)

Proteins like many other things in nature are three-dimensional. Owing to the structural complexity of proteins their structure is studied at four different levels known as primary, secondary, tertiary and quaternary structures. The primary structure of a protein is its linear arrangement of amino acids. In many proteins there are repeating structural motifs resulting from hydrogen bonding between amino acids in a polypeptide chain. These motifs form the secondary structure of the protein. Two common motifs are alpha helices and beta pleated sheets. The level of organization right above the secondary structure is tertiary structure. Tertiary structure is the consequence of proteins contorting themselves to form a more compact three dimensional structure. Quaternary structure on the other hand refers to the structure of the protein formed when two or more proteins with tertiary structure, called subunits, come together to form a protein.

4.2.3 Determining Three-Dimensional Structure of Proteins

Two major techniques for determining the three-dimensional structure of proteins are

X-ray crystallography and nuclear magnetic Resonance spectroscopy (Komoroski 2002).

In X-ray crystallography, the protein to be studied must first be crystallized. X-rays are then passed through the resulting crystal. The X-rays diffract and form an image known as a diffraction pattern on photographic film or on a radiation counter. From the diffraction pattern, a map showing the electron density of the protein can be constructed. Through a combination of electron density map and protein primary structure information, the location of each atom in the protein can be determined. Unfortunately, this technique only works for proteins that can be crystallized. In solution magnetic resonance, an aqueous solution of the protein is made. The spins of the protons in the protein are then aligned by placing the solution in a magnetic field. The application of radiofrequency pulses excites the protons and results in them emitting signals with frequencies depending on the molecular environment of the protons. The signals emitted can be used to come up with estimates of the distances between protons in the protein being studied. Using the distances and information such as bond angles, three-dimensional ensembles of models of the protein can be computed. A collection of possible structures is obtained as opposed to a one unique structure (Komoroski 2002).

4.2.4 Hemoglobin

Respiration in living cells requires oxygen. Oxygen enters the human body through the pulmonary system and has to be carried to cells all over the body. Hemoglobin is the protein found in red blood that carries oxygen to most of the cells the body (Hardison 1996).

In human adults hemoglobin is a protein with a quaternary structure composed of two sets of two identical subunits. The two alpha subunits are each made up of 141 amino acids and the beta subunits are made up of 146 subunits each. The most important part of each subunit is its heme group. The heme group is a cofactor (a non-protein compound required for the proper functioning of certain proteins) with a central iron atom. An oxygen molecule binds reversibly to each subunit via its heme group; consequently, each molecule of hemoglobin can carry four molecules of oxygen. Upon reaching its destination, hemoglobin unloads the oxygen molecules. In Fig. 4.4 the heme group is the tiny structure within the polypeptide chains (Hardison 1996).

4.2.5 Myoglobin

Myoglobin is similar in function to hemoglobin except that its function is to store oxygen in muscle cells (George *et al.* 2004). Diving animals like seals and whales often spend extended periods of time underwater and as such are not able to take in oxygen through their blow holes and into their pulmonary systems as they normally do on land. In times like these diving animals rely heavily on myoglobin for oxygen. Myoglobin is also found in the skeletal and cardiac muscle of non-diving animals. Human myoglobin consists of one peptide chain consisting of 154 amino acids. It only has one heme group, and consequently can only store one molecule of oxygen. Fig. 4.5 shows the structure of myoglobin.

Fig. 4.4 Hemoglobin
(<http://fig.cox.miami.edu/~cmallery/150/protein/proteinsb.htm>).

Fig. 4.5 Myoglobin
(<http://courses.cm.utexas.edu/jrobertus/ch339k/overheads-1.htm>).

4.2.6 Prolactin

Prolactin is a hormone produced in the anterior pituitary gland and has a diversity of effects in humans and in other animals. In humans, its primary role is the stimulation of milk production in the mammary gland. Structurally, human prolactin is a polypeptide consisting of 227 amino acids. The most important site on most hormones is the binding site. Hormones like prolactin attach themselves via their binding sites to a receptor on their target cells and effect their action (Speroff *et al.* 1999). Fig. 4.6 shows the structure of prolactin.

between certain proteins in the inner mitochondrial membrane. Cytochrome C carries out part of the electron transfer by shuttling electrons from a complex known as the cytochrome b-c1 complex to another complex known the cytochrome oxidase complex. The lone heme group within the cytochrome c is the carrier of the electron (Reece and Campbell 2005). Fig. 4.8 shows the structure of cytochrome C.

4.2.9 Lysozyme

Enzymes are substances that speed up reactions without taking part in the reactions themselves. Lysozyme is an enzyme whose primary role in humans is protection against bacteria. It effects this action by making possible a hydrolysis reaction in which a water molecule is use to break polysaccharide molecules in bacterial cell walls. The most important site of the lysozyme molecule is the active site in which parts of bacterial cell walls lodge before they are hydrolyzed. In humans, lysozyme is found in several secretions including tears where it aids in keeping the eyes bacteria free and mucus for protecting the inner lining of the nose. Rat and human lysozyme are examined in this study (Reece and Campbell 2005). Fig. 4.9 shows the structure of human lysozyme.

4.2.10 Epidermal Growth Factor

Epidermal growth factor is protein that unlike most proteins is not confined to a particular site of synthesis or a specific target cell type. Its varied effects include accelerating incisor eruption and eye-lid opening in new-born mice, the proliferation of epidermal cells and the inhibition of gastric juice secretion (Carpenter and Cohen 1990).

Fig. 4.9 shows the structure of epidermal growth factor.

Fig. 4.6 Prolactin (http://en.wikipedia.org/wiki/Image:PRL_structure.png).

Fig. 4.7 Glucagon (<http://en.wikipedia.org/wiki/Image:Glucagon.png>).

Fig. 4.8 Cytochrome C (<http://bai.nci.nih.gov/>).

Fig. 4.9 Human Lysozyme
(<http://commons.wikimedia.org/wiki/Image:Lysozyme.png>).

Fig. 4.10 Epidermal Growth Factor
(<http://www-nmr.cabm.rutgers.edu/photogallery/proteins/htm/page3.htm>).

4.3 Identification of Active Sites in Protein Sequences Using Time-Frequency Analysis

As explained in the previous sections, the amino acid sequence of a protein is typically represented as a string of 20 different letters indicating the 20 naturally occurring amino acids. So an amino acid mapping index is a set of 20 numerical values representing specific

physicochemical or biochemical properties of the amino acids. Each of the 20 amino acids has various properties that are responsible for the specificity and diversity of protein structure and function, experimental and theoretical research has characterized different kinds of properties of individual amino acids and represented them in terms of a numerical index (Kawashima and Kanehisa 2000). The amino acid index database currently contains more than 500 indices (<http://www.genome.ad.jp/dbget/aaindex.html>). Among these indices, EIIP is the electron-ion interaction potential which is the energy of delocalized electrons of each amino acid (Cosic 1997). This mapping represents the distribution of the free electron energies along the protein sequence. Mapping the symbolic representation of an amino acid sequence into the EIIP representation allows signal analysis methods to be used to study the structural or functional patterns within protein sequences (Pirogova and Cosic 2001).

In this application of signal processing techniques to the sequences, the sampling rate can be assumed to be one amino acid since the distance between amino acids is about $3.8\text{\AA}$. Protein sequences of the test set are transformed into numerical sequences of the RRM by assigning the EIIP value in Ry to each amino acid of the macromolecule. The values of EIIP for 20 amino acids are given in Table 4.1 (Cosic 1997). Fig. 4.11 shows the protein sequence of the alpha and beta chains of human hemoglobin transformed into a numerical sequence by assigning an EIIP value to each amino acid.

The FFT is proved to be a useful tool for protein signal sequence analysis (Katoh *et al.* 2002). However, the FFT gives no time information and so nothing is known about which parts of the signal contribute to which frequencies. Consequently, if there are two signals composed of similar frequencies and the case that these frequencies occur at very different

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

locations in time, then their FFT may appear highly alike. Also the FFT assumes that the signals being analyzed are stationary.

Table 4.1 Electron-ion interaction pseudopotential (EIIP) of amino acids.

Amino acid	EIIP/Ry	Amino acid	EIIP/Ry
D (Aspartic acid)	0.1263	A (Alanine)	0.0373
R (Arginine)	0.0959	K (Lysine)	0.0371
F (Phenylalanine)	0.0946	H (Histidine)	0.0242
T (Threonine)	0.0941	P (Proline)	0.0198
C (Cysteine)	0.0829	E (Glutamic acid)	0.0058
S (Serine)	0.0829	V (Valine)	0.0057
M (Methionine)	0.0823	G (Glycine)	0.0050
Q (Glutamine)	0.0761	N (Asparagine)	0.0036
W (Tryptophan)	0.0548	I (Isoleucine)	0.0000
Y (Tyrosine)	0.0516	L (Leucine)	0.0000

The time-frequency analysis techniques provide one way to alleviate the problem. One of the simplest time-frequency analysis techniques is the STFT. Windows of different sections of the signal are analyzed with the FFT. As the windows have specific sizes, time information is not lost. A window length of 12 points is chosen as this is close to the average length of an alpha helix, one of the common secondary structure conformations in proteins. Overlap of 11 points is added between windows. Overlap is important as it dictates the amount of frequency information lost due to the splitting of the signal into windows.

There are various types of TFDs have been discussed in Chapters 2 and 3. Among them the WVD, SPWVD and RSPWVD are powerful signal processing techniques that can be used in multi-resolution analysis and feature extraction of signals.

Fig. 4.11 Protein sequence of human hemoglobin chains is transformed into a numerical sequence by assigning an EIIP value to each amino acid: (a) alpha chain, (b) beta chain.

All the TFDs provide time-frequency representations of a signal. They can easily capture the evolution of frequencies within a signal over time. Meanwhile, the TFDs are able to capture time information and other than the frequency information. In this study, the

time series consist of hydrophobicities of consecutive amino acids in a protein. Consequently, the ‘time’ actually represents the location of an amino acid within a protein sequence. The reassignment method of TFD, such as the RSPWVD, can greatly improve the sharpness of the location of the signal components by relocating its TFD in the time-frequency space. Although the computational cost of the direct RSPWVD algorithm is high, new algorithms such as the simplified and time-recursive RSPWVD algorithms (Chapter 3) have a much better efficiency. So the time-recursive RSPWVD algorithm is used here to analyze the sequences for site identification. Meanwhile, the spectrogram (the magnitude squared of STFT) is also used for comparison. In applying identification algorithms to critical parts of the proteins, one has to look for the local-energy maxima in the space-frequency representation of the protein signal because the amino acids being searched are those that contribute the most energy.

4.4 Results and Analysis

4.4.1 Human Hemoglobin

The oxygen-carrying heme prosthetic group has an iron atom that is critical to its function. This atom is attached to the 87th residue in the alpha chain. Lehmann (1966) and Huntsman also point out that there exists a gap between the iron atom in the heme group and the alpha chain’s 58th residue. This gap accommodates an oxygen molecule. Several important residues that have been implicated in the functioning of hemoglobin are located between the 40th and 90th residues. The 64th, 70th, 79th and 83rd amino acid residues are of particular importance (Cosic *et al.* 2000). An examination of the spectrogram of the

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

alpha chain of hemoglobin (Fig. 4.12) reveals that most of the high-energy areas are located between the 40th and 90th residues, with the most significant region occurring in the vicinity of the 87th amino acid residue. There is also a hot spot at the 63rd amino acid. Residues 18, 22, 36, 43 and 59 are also within hot spot regions. These amino acids are significant as Cosic *et al.* (2000) report that even though they are not part of the active site they 'group in a domelike fashion over the heme group.' In the beta chain, the heme group is linked and the 92nd residue and the oxygenaccomodating-gap is located at the 63rd residue. Most of the important residues are found between the 35th to the 90th residues. The 94th and 100th residues are also thought to be important. As expected, the spectrogram of the beta chain of hemoglobin (Fig. 4.14) shows that most of the high-energy regions occur between the 35th and 90th amino acids. There are two regions of high energy at the 63rd residues and a region of extremely high energy at the 92nd residue capturing both the oxygen-accomodating space and the heme-linked residue. Cosic *et al.* (2000) note that residues 42, 45, 64 and 69 form a dome over the heme group and all these residues fall within the high-energy regions. There are unaccounted regions for energy below the 35th residue and above the 92nd residue which warrant experimental investigation. The RSPWVD representation of the beta chain (Fig. 4.15) presents basically the same information as the spectrogram. It too has regions of high energy whose role in the function of hemoglobin was not found in the literature. However, Fig. 4.13 is more concentrated and there is less 'noise'.

Fig. 4.12 Spectrogram of human hemoglobin alpha chain, where n is the index of amino acids.

Fig. 4.13 RSPWVD of human hemoglobin alpha chain, where n is the index of amino acids.

Fig. 4.14 Spectrogram of the beta chain of hemoglobin, where n is the index of amino acids.

Fig. 4.15 RSPWVD of the beta chain of hemoglobin, where n is the index of amino acids.

4.4.2 Myoglobin

Residues 64 and 93 together with the heme group form active site of myoglobin. Other important residues and residue ranges are the 138th amino acid and 29th to 45th, 64th to 72nd and 89 to 107th residue ranges (Lehmann and Huntsman 1966).

The regions referred to earlier are covered by the high-energy regions in the spectrogram (Fig. 4.16) and the RSPWVD (Fig. 4.17). There are high-energy regions falling outside the aforementioned ranges. Cosic *et al.* (2000) point out that the 25th, 33rd, 46th and 65th residues, which are all captured by the spectrogram and the RSPWVD, are part of the dome-shaped region covering the heme group.

Fig. 4.16 Spectrogram of myoglobin, where n is the index of amino acids.

Fig. 4.17 RSPWVD of myoglobin, where n is the index of amino acids.

4.4.3 Cytochrome C

The lone heme group of cytochrome C is bonded to the cysteine residues at positions 14 and 17. Tzagaloff reported that amino acids important in the association of cytochrome c to its heme group can be found in the ranges 10-17, 30-35, 64-71, 82-85 and 94-98 (Tzagaloff 1982).

Both the spectrogram (Fig. 4.18) and RSPWVD (Fig. 4.19) capture the ‘hotspots’ of cytochrome c. However, the intensity of the protein critical parts produced by the spectrogram is significantly less than that of those produced by the RSPWVD. In Fig. 4.19, there are lots of excess high-energy spots whose possible role in the function of cytochrome C merites experimental exploration.

Fig. 4.18 Spectrogram of human cytochrome C, where n is the index of amino acids.

Fig. 4.19 RSPWVD of human cytochrome C, where n is the index of amino acids.

4.4.4 Prolactin

Prolactin has two binding sites. Binding Site I consists of amino acids in the ranges 19-37 and 169-180. Site II consists of residues between the 21st to the 32nd amino acid residues and residues between the 110th to the 122nd amino acids (Norman and Litwak 1997).

The experimentally determined hot spot ranges fall within the ranges of high energy in both the spectrogram and RSPWVD of prolactin. The brightest regions in the spectrogram (Fig. 4.20) and RSPWVD (Fig. 4.21) occur around amino acids 10-28 and 28-90, 100-210. The 10-28 region is near prolactin's N-terminus which Clapp and Weiner (1992) claim is of biological importance. In the 28-90 range the section of highest energy is located in the vicinity of the 37th amino acid which de Trad *et al.* (2000) cite as a binding determinant for human prolactin. The 58th to 74th amino acids are also known to be bioreactive. The 129th amino acid residue is believed to be important in maintaining the geometry of the second binding site of prolactin. The last region of high energy indicated is the 100-210 amino acids. Within this range the zone of highest energy radiates from the 129th amino acid. Residues 176 and 177 in the 100-200 are also binding determinants.

Fig. 4.20 Spectrogram of prolactin, where n is the index of amino acids.

Fig. 4.21 RSPWVD of prolactin, where n is the index of amino acids.

4.4.5 Lysozyme

Owing to its similarity with chicken / hen egg white lysozyme, human lysozyme is member of the chicken-type (c-type) lysozyme family. The c-type lysozymes have an active site divided into six subsites. The subsites are spread over the 35th to the 63rd and the 98th to 114th amino acid residues. The residues Glu-35 and Asp-52 while not part of any subsite play a catalytic role (Strynadka 1996).

Fig. 4.22 and 4.23 show that both the RSPWVD and spectrogram capture the experimentally predicted critical parts. In both figures, the region of highest-energy is the 98th to 114th amino acid zone. Owing to the excess energy appearing in the spectrogram image, it appears that this is a case where the RSPWVD is better suited than the spectrogram for protein analysis.

Fig. 4.22 Spectrogram of lysozyme, where n is the index of amino acids.

Fig. 4.23 RSWVD of lysozyme, where n is the index of amino acids.

4.4.6 Epidermal Growth Factor

In their structure analysis of epidermal growth factor, Groenen *et al.* (1994) report binding affinity studies of fragments of epidermal growth factor and suggest that amino acid residues 4 to 48 are required for strong binding to the epidermal growth factor receptor. The spectrogram of human epidermal growth factor (Fig. 4.24) does not follow the Groenen *et al.*'s (1994) experimental results. However there are high-energy areas located in the vicinities of residues 37, 41, 43 and 47 which Lu *et al.* (2001) report to be among the most important residues for the function of human epidermal growth factor. Fig. 4.25 reveals that high-energy regions span the region from the 10th amino acid residue to slightly beyond the 48th amino acid residue. The lack of energy between the 4th to 10th

amino acids might be due to the thresholding incorporated in the calculation of the scalogram. The 37th, 41st, 43rd and 47th amino acids appear in the regions with the highest energy.

4.4.7 Glucagon

The hormone glucagon is an interesting molecule as all its 29 amino acid residues are critical to its role in inducing the breakdown of fatty acids and glycogen into glucose (Norman and Litwak 1997). The spectrogram (Fig. 4.26) fails drastically in this case as it signifies that only residues 15 to 25 are of importance. The RSPWVD (Fig. 4.27) gives a much better result suggesting that the 7th right through to the 29th amino acids are important.

4.4.8 Comparison of Computational Time

In order to compare the computational cost of the STFT and RSPWVD analysis tools existing, their computational times are recorded. The STFT analysis uses a 40-point Hamming window, and the RSPWVD analysis uses the time-recursive RSPWVD algorithm with the same window as the time-direction smoothing and the frequency-direction smoothing. Table 4.2 shows the results. However, the RSPWVD analysis gives better results, its computational time is about three times longer than the STFT analysis.

Fig. 4.24 Spectrogram of human epidermal growth factor, where n is the index of amino acids.

Fig. 4.25 RSPWVD of human epidermal growth factor, where n is the index of amino acids.

Fig. 4.26 Spectrogram of glucagons, where n is the index of amino acids. Fig. 4.27 RSPWVD of glucagons, where n is the index of amino acids.

Table 4.2 Computational time of the protein analysis (using a 40-point Hamming window).

Protein Analyzed	Computational Time of Different Analysis Method (s)	
	STFT Analysis	RSPWVD Analysis
Hemoglobin	12.54	29.78
Myoglobin	11.56	26.79
Prolactin	17.29	46.61
Glucagon	13.89	31.21
Cytochrome C	15.12	41.98
Lysozyme	12.77	33.76
Epidermal Growth Factor	14.98	41.22

4.5 Quantitative Comparison of Results

Three measures can be used to compare the performance of STFT and RSPWVD quantitatively, the two-dimensional correlation, the IF correlation, and the time-frequency resolution (Guo *et al.* 1994, Forsberg and Oung 1995).

The central two-dimensional correlation is defined as

$$\rho = \frac{\sum_l \sum_k E_d(l, k) E_e(l, k)}{\sqrt{\sum_l \sum_k E_d(l, k)^2 \sum_l \sum_k E_e(l, k)^2}} \quad (4.1)$$

where, $E_d(l, k)$ is the desired TFD and $E_e(l, k)$ is the estimated TFD and l and k are the discrete time and frequency indexes respectively. This measure is used to assess the difference in terms of energy distribution, between the estimated TFD and the ideal version.

The IF correlation is defined as

$$IF = \frac{\sum_l \sum_k E_{dn}(l, k) E_{en}(l, k)}{\sqrt{\sum_l \sum_k E_{dn}(l, k)^2 \sum_l \sum_k E_{en}(l, k)^2}} \quad (4.2)$$

where the n subscript denotes a conditional normalization of $E_d(l, k)$ and $E_e(l, k)$ such that

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

$$E_{*n}(l, k) = \begin{cases} 1 & \text{if } E_{*n}(l, k) > 2E \\ 0 & \text{if } E_{*n}(l, k) \leq 2E \end{cases} \quad (4.3)$$

where,

$$E = \frac{\sum_l \sum_k E_e(l, k)}{N^2} \quad (4.4)$$

and N is the number of time and frequency samples in $E(l, k)$. This is a measure of the general shape of the TFD in terms of its IF laws.

The time-frequency resolution measure is defined as

$$res = \frac{\sum_l B(l)}{N} \cdot \frac{\sum_k T(k)}{N} \quad (4.5)$$

where,

$$B(l) = \frac{\sum_k E_e(l, k)}{\max_k E_e(l, k)} \quad (4.6)$$

$$T(k) = \frac{\sum_l E_e(l, k)}{\max_l E_e(l, k)} \quad (4.7)$$

The resolution measure is used to measure the energy concentration of the TFD.

Here the performance measures have been normalized to assess the relative performance between STFT and RSPWVD. The normalization procedure is defined as

$$\rho_n = \frac{\rho}{\max\{\rho\}}, \quad IF_n = \frac{IF}{\max\{IF\}}, \quad res_n = \frac{res}{\max\{res\}} \quad (4.8)$$

where the normalization occurs within each pair of results from Figure 12-27. For two-dimensional correlation and IF correlation, the value 1.00 infers the best performance. For time frequency resolution, the value 0.0 relates to the best performance. Table 4.3 shows the value of performance measures over the eight pairs of results from Figure 12-17.

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

Table 4.3 The value of performance measures over the eight pairs of results from Figure 12-27.

	STFT			RSPWVD		
	ρ	IF	res	ρ	IF	res
Figure 4.12 and 4.13	0.76	0.89	0.22	1.00	1.00	0.06
Figure 4.14 and 4.15	0.81	0.83	0.09	1.00	1.00	0.03
Figure 4.16 and 4.17	0.79	0.88	0.12	1.00	1.00	0.05
Figure 4.18 and 4.19	1.00	1.00	0.21	0.87	0.76	0.16
Figure 4.20 and 4.21	0.77	0.79	0.17	1.00	1.00	0.08
Figure 4.22 and 4.23	0.57	0.61	0.75	1.00	1.00	0.06
Figure 4.24 and 4.25	0.63	0.71	0.28	1.00	1.00	0.10
Figure 4.26 and 4.27	0.56	0.62	0.67	1.00	1.00	0.21

Both of the STFT and RSPWVD perform variedly, depending on the time frequency characteristics of the input signal. In most cases, the RSPWVD outperforms the STFT. Only in the pair of results in Figure 4.18 and 4.19, the STFT has better performance in the energy distribution and IF laws. In Figure 19 there are lots of excess high-energy spots which may be related to the function of cytochrome C. However, during the computation of ρ and IF these spots are treated as noise. It is a very special case, and so conclusion can be made that the RSPWVD method has better performance over the STFT method in time frequency representation of protein signal sequence and localizing critical parts.

4.6 Summary

Chapter 4: Identification of Critical Parts of Proteins Using Time Frequency Analysis

Protein signals are constructed based on amino acid hydrophobicity and analyzed with the STFT and the RSPWVD. The objective is to locate the amino acids critical to the function of these proteins in the resulting spectrograms. They show that both the spectrogram (the magnitude squared of STFT) and the RSPWVD are able to locate a protein's critical parts given a protein signal based on amino acid hydrophobicities. This proves that hydrophobicity is a viable datum to be used in protein critical part localization via time-frequency analysis. Furthermore, quantitative comparison between the STFT and RSPWVD methods are made. Three quantitative measures, the two-dimensional correlation, the IF correlation, and the time-frequency resolution are used for comparison.

Both the qualitative and quantitative results reveal that the RSPWVD method is better than the STFT method in localizing critical parts. However, the computational cost of the RSPWVD method is higher than the STFT method.

Chapter 5: Computational Gene-Finding Using Time-Frequency Analysis

5.1 Introduction

In this era of intensive genomic sequencing, when millions of bases of genomic DNA are sequenced daily in genome centers worldwide and the list of completely sequenced genomes from different organisms is growing rapidly, tools for interpreting the content of these genomes are more important than ever. The first step in deciphering the DNA sequence information is finding all the genes contained in a sequence and elucidating their structure. Although many gene-finding programs have been developed in the past 10 years and their prediction accuracy is constantly improving, the completely automatic gene discovery with 100% accuracy is still not achieved. Current programs, although very good in discovering the majority of coding nucleotides (more than 90% predicted correctly) and moderately good in discovering exact exon boundaries (80-85% of exons predicted correctly) are still weak when it comes to predicting complete gene structures. Less than 50% of predicted genes correspond exactly to the actual genes. Consequently, predictions given by these programs need to be verified by other evidence such as similarity to a cDNA sequence, or similarity to a known protein or expressed sequence tag (EST) sequence (Collins *et al.* 2003).

There are several methods for experimental discovery of genes, but they are time-consuming and costly. Accordingly, for the last 15 years researchers have been

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

developing computational methods for gene-finding that could automate, or facilitate, identification of genes. Two basic approaches have been established for computational gene-finding: (i) the sequence similarity search or lookup method, and (ii) the integrated compositional and signal search or template method (Fickett 1996).

5.1.1 Sequence Similarity Search Method

Sequence similarity search is a well established computational method for gene discovery, which has been used extensively with considerable success. It is based on sequence conservation due to the functional constraints and is used to search for regions of similarity between uncharacterized sequence of interest and already characterized sequences in a sequence database. A query sequence can be compared with DNA, protein or EST sequences or it can be searched for known sequence motifs (Dunham *et al.* 1999). If a query sequence is found to be significantly similar to an already annotated sequence (DNA or protein), the information from the annotated sequence can be used to maybe infer gene structure or function of the query sequence. Comparison with an EST database can provide information if the sequence of interest is transcribed, but will only give incomplete clues about the structure of the whole gene or its function.

Although sequence similarity search has been proven useful in many cases, it has been shown that only a fraction of newly discovered sequences have identifiable homologs in the current databases (Dunham *et al.* 1999). Furthermore, currently known proteins may already include representatives of most ancient conserved regions and that new sequences not similar to any database sequence are unlikely to contain ancient conserved regions. This

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

is supported by recent analysis of human chromosome 22 (Dunham *et al.* 1999) where only 50% of the proteins are found to be similar to previously known proteins. These results suggest that, even today, only one half of all new vertebrate genes may be discovered by sequence similarity search across phyla. So it is obvious that sequence similarity search within vertebrates is currently limited.

5.1.2 The Integrated Compositional and Signal Search Method

The second computational approach for prediction of genes structures in the genomic DNA sequences, termed the template approach, integrates coding statistics with signal detection into one framework. Coding statistics are measures indicative of protein coding function since they behave differently on coding and non-coding regions. A number of these measures have been evaluated in (Fickett and Tung 1992) and it has been concluded that the in-phase hexamer measure, which measures the frequency of occurrence of oligonucleotides of length six in a specific reading frame, is the most effective. Indeed, this measure was used successfully in many recently developed programs such as GeneMark.hmm (Lukashin and Borodovski 1998), Genscan (Burge 1997) and HMMgene (Krogh 1997). This coding statistic is usually implemented as a 5th order hidden Markov model (HMM). Signal sensors attempt to mimic closely processes occurring within the cell. They are intended to identify sequence signals, usually just several-nucleotides-long subsequences, which are recognized by cell machinery and are initiators of certain processes. The signals that are usually modeled by gene-finding programs are: promoter elements, start and stop codons, splice sites, and polyA sites.

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

Many different pattern recognition methods have been used as signal detectors, including simple consensus sequences, weight matrices, weight arrays, neural network, and decision trees. DNA sequence signals have low information content; they are usually degenerate and highly unspecific since it is almost impossible to distinguish the signals truly processed by the cell from those that are apparently non-functional. Therefore, signal sensors are not sufficient to elucidate gene structure and it is necessary to combine them with coding statistics methods in order to obtain satisfactory predictive power. Both codon statistics and signal models are “learned” from a training set: frequencies of oligonucleotide occurrence in different regions of the genes are calculated from sequences in the training set and the signal models are constructed using the multiple alignment of the signal sequences from the training set.

5.1.3 Gene-Finding Programs

Existing gene-finding programs have been designed to identify gene structure simpler than the intricate structure described above. Most of the programs, especially the older ones, are trained to identify just one gene in a sequence, rarely predicting any promoter elements. (Burge 1997) Some progress has been made with recently developed programs, which are capable of identifying more complex genomic structure: any number of genes with either complete or partial structure. This is the case with GeneMark.hmm, Genscan, and HMMgene. Nevertheless, the prediction of the coding sequence of typical genes is an important first step in deciphering the content of any genome and gene-finding programs are used extensively for this task with considerable success.

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

In some previous work the DNA sequences can be viewed as character strings, and be digitalized into numerical sets. Then digital-signal-processing techniques can be applied for searching useful information buried in this sequence (Anastassiou 2002). What to be done in this project is to use “DNA spectra” for the identification of genes. In order to use time-frequency analysis tools, some mapping algorithms from characters to numbers are to be developed first.

5.2 Testing and Training Dataset for Gene-Finding Programs

5.2.1 Testing Dataset for All Programs

Here a dataset is constructed for the evaluation of gene structure prediction programs. All the sequences in the dataset should not be the training sequences of these programs. Furthermore, not only the human sequences but also the mouse sequences are selected. Both these two kinds of genomes are well studied, and presented in GenBank. Also human and mouse genomes are phylogenetically close enough to be analyzed with the same parameter files used in the gene-finding programs. DNA sequences are extracted from NCBI GenBank (<http://www.nlm.nih.gov>). The basic requirements in sequence selection were:

- (i) Only genomic sequences that contained exactly one gene considered
- (ii) The source organism is *Homo sapiens* or *Mus musculus*
- (iii) mRNA sequences and sequences containing pseudo genes or alternatively spliced genes are excluded.

Sequences collected according to those principles are further filtered to meet the

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

following requirements:

- (i) All annotated coding sequences are started with the *ATG* initiation codon and ended with one of the stop codons: *TAA*, *TAG* and *TGA*
- (ii) All exons have dinucleotide *AG* at their acceptor site and dinucleotide *GT* at their donor sites
- (iii) Sequences that don't contain any nucleotides in their 5' or 3' *UTR* are discarded;
- (iv) Sequences longer than 200,000 bp are discarded because some of the programs analyzed can only accept sequences up to that length; sequences whose coding region contains in-frame stop codons are discarded.

Sequences that passed these filtering steps are further subjected to non-redundancy testing. All-against-all neighbour search with the Entrez Browser (command line Entrez-Nentrcmd from the NCBI tool kit, <http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi>) is performed and if two sequences are linked as neighbors only one of them is selected to enter the final dataset. Neighbour linkage in Entrez represents high similarity between two sequences.

The final restriction of the dataset is imposed to confirm exon locations annotated in the GenBank records. For each sequence in the dataset the BLAST algorithm (Altschul *et al.* 1997) is used to find a corresponding mRNA sequence that has been independently sequenced and not derived from the genomic sequence. If such an mRNA sequence exists, the sim4 program (Florea *et al.* 1998) is used to align the genomic sequence and the mRNA sequence. The result of the sim4 alignment is the list of exon locations, which are then

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

compared with the annotation in the GenBank record. Only those sequences whose exon annotation perfectly matches the sim4 results are selected for the final version of the dataset. Unfortunately, this analysis can not confirm the start location of the initial exon and the end location of the terminal exon. Since mRNAs also contain 5' and 3' untranslated regions (UTR) that also align to the genomic sequence, so these annotations remains unconfirmed.

The resulting dataset contains 132 sequences with exactly one complete gene, either single-exon or multi-exon, and it is named HMD132 (Appendix A), which has the following characteristics:

- (i) The ratio of human: mouse sequences is 66:66
- (ii) The mean length of the sequences in the set is 7,241 bp
- (iii) The number of single-exon genes is 43 and of multi-exon genes is 52
- (iv) The average number of exons per gene is 4.86
- (v) The mean exon length is 208 bp, the mean intron length is 678 bp, and the mean coding length of a gene is 1,015 bp (~340 amino acids)
- (vi) The proportion of coding sequence in this dataset is 14%, the proportion of the intronic sequence in this dataset is 46%, and the proportion of the intergenic DNA in this dataset is 40%.

The dataset is not a typical subset of sequences from human and mouse genomes. The fraction of coding sequence in the dataset is 14%. It is much higher than the fraction of coding sequence for estimated human and mouse genomes, which is 3%. The mean coding length of 340 amino acids is shorter than the calculated mean of 450 amino acids for *C.*

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

elegans and *S. cerevisiae* (Zhang, 2000). It is realistic to expect that the average protein in human and mouse will be at least this long, since the analysis of Zhang (2000) shows that protein length seems to increase with the complexity of organism. Also, the average number of exons in a gene in HMD132, which is 4.86, is lower than the calculated 5.4 for human chromosome 22 (Dunham *et al.* 1999). The discrepancies are a direct product of biases in GenBank and other public sequence databases. With the current limited amount of data it is not yet feasible to generate a dataset which would perfectly model human and mouse genomes. Also, the proportion of single-exon genes in HMD132 substantially exceeds that in real genomes. Overrepresentation of these sequences in GenBank is another source of the discrepancies. Since there is no other biological reason to eliminate single-exon genes, it is chosen to keep them within the dataset.

5.2.2 Training Dataset for Gene-Finding Using Time-Frequency Analysis

Just like the testing dataset HMD132, a training dataset is constructed to obtain the optimized parameters for gene-finding programs using time-frequency analysis. The training dataset contains 64 sequences with exactly one complete gene, either single-exon or multi-exon, and it is named THMD32 (Appendix B). Optimized parameters are used in finding genes in the sequences of HMD132. The sequences in THMD32 is very similar to those in HMD132. The THMD32 is constructed with the following characteristics:

- (i) None of the sequences in THMD32 is in HMD132
- (ii) The ratio of human: mouse sequences is 16:16
- (iii) The mean length of the sequences in the set is 7,145 bp

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

- (iv) In both the human sequences and the mouse sequences, the number of single-exon genes is 7 and the number of multi-exon genes is 25
- (v) The average number of exons per gene is 4.46
- (vi) The mean exon length is 197 bp, the mean intron length is 624 bp, and the mean coding length of a gene is 1,101 bp (~367 amino acids)
- (vii) The proportion of coding sequence in this dataset is 15%, the proportion of the intronic sequence in this dataset is 42%, the proportion of the intergenic DNA in this dataset is 43%.

5.2.3 Tests of Gene-Finding Programs

Traditional gene-finding programs under test are described here. For each program a short description of the methods used by the program is given.

(A). *GeneMark.hmm*

It (<http://exon.gatech.edu/GeneMark/hmmchoice.html>) uses an explicit state duration HMM. The optimal gene candidates selected by the HMM and dynamic programming are further processed by a ribosomal binding site recognition algorithm. The dataset used for training is not described. The type (first, internal, last, single) and location of each exon is specified in the output of the program. In this analysis GeneMark.hmm is run with the human.mtx matrix for every sequence in HMD132, and it predicts genes in every sequence.

(B). *Genescan*

In this program (<http://genes.mit.edu/GENSCAN.html>) the structure of the genomic

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

sequence is modeled by an explicit state duration of the HMM. The states of the HMM are probabilistic models themselves. Signals are modeled by weight matrices, weight arrays and maximal dependence decomposition. The model of Genscan can predict the absence of genes or the presence of a single gene or multiple genes, which can be either complete or partial. It also has the option to predict suboptimal exons, which are defined as potential exons with a probability higher than a certain threshold but which are not contained in the optimal parse of the sequence. This type of exon can potentially represent alternatively spliced exons. Genscan is trained on Kulp *et al.*'s (1996) dataset of human genomic sequences and an additional set of 1999 human cDNA sequences are used for training the coding region HMM. The maximal length of the input sequence for this version of Genscan is 200 kb. The output of Genscan is similar to the output of GeneMark.hmm, giving information about exon locations and their probabilistic scores, but scores for other sequence features such as splicing sites are also given. Genscan is run with parameter file HumanIso.smat for all the sequences in HMD132. It predicts genes in 129 out of 132 sequences.

(C.) HMMgene

The program (<http://www.cbs.dtu.dk/services/HMMgene/>) is based on HMMs and it is trained using a criterion called conditional maximum likelihood, which maximizes the probability of correct prediction. If the sequence analyzed already has some subregions identified, those regions can be locked as coding or non-coding and then submitted to HMMgene. The underlying gene structure model can predict both partial and complete

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

genes in sequence and any number of them. The program has the option to give more than one prediction, which can indicate alternative splicing of the gene in the sequence. The dataset of human single- and multi-exon genes collected by Kulp *et al.* (1996) is used for the training of this program. The output does not give the location of the splicing sites, but only of the exons, whose type is also specified. Every sequence from the testing dataset of HMD132 is submitted to the program, which predicts genes in 128 out of 132 sequences.

5.3 Gene-Finding Programs Using Time-Frequency Analysis Techniques

Now novel gene-finding programs using the ‘DNA Spectra’ for the identification of genes are developed. The time-frequency analysis is used to obtain the DNA spectra. The programs are in two parts: the optimized parameters computation using THMD32 and the gene sites identification for HMD132.

5.3.1 Optimized Parameters Computation

The optimized parameters computation algorithm is given in Algorithm 4.1.

(A). The Sequence Mapping and TFD Computation

The algorithms to compute TFDs are discussed in Chapter 3. Here, three different TFDs, STFT, WVD and RSPWVD are used for the novel gene-finding programs. For computation efficiency, the time-recursive RSPWVD algorithm is chosen for RSPWVD. All these three TFDs describe the energy density of a signal simultaneously in the time and frequency domains. The frequency energy of a certain TFD is defined as (Cohen 1995)

$$E = \int_{-\infty}^{+\infty} |TFD(t, \omega)| d\omega \quad (5.1)$$

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

Algorithm 4.1: The optimized parameters computation algorithm

- Step (0): The sequences of THMD32, in the form of character sequence, are input from external files.
- Step (1): For each sequence of THMD32, a random sequence of the same length is generated.
- Step (2): Four indicator sequences are constructed to represent all the sequences; each indicator sequence represents a kind of nucleotide (A , T , C , or G)
- Step (3): The TFD and frequency energy of each indicator sequence are computed.
- Step (4): The TFD and frequency energy of each indicator sequence are computed.
- Step (5): The optimization function provided by Matlab is used to get the optimization parameters for each training sequence of THMD32, which can maximize the predictive power between protein coding regions (with respective random variables A , T , C and G) and regions of random sequence.
- Step (6): The mean optimized parameters for human and mouse sequences are calculated in the human sequences set and mouse sequences set of THMD32, respectively.

Assume that the numbers a , t , c , g are assigned to the characters A , T , C , G respectively. A DNA sequence of length N can be represented as

$$s[n] = as_A[n] + ts_T[n] + cs_C[n] + gs_G[n], \quad n = 0, 1, 2, \dots, N-1 \quad (5.2)$$

In which $s_A[n]$, $s_T[n]$, $s_C[n]$ and $s_G[n]$ are called the binary indicator sequences

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

which take the value of either one or zero at location n , depending on whether or not the corresponding character exists at location n . For example, in the sequence of *CATTGCCAGT*, $s_C[0] = s_C[5] = s_C[6] = 1$.

Because only one nucleotide is possible at any index, $s_A[n]$, $s_T[n]$, $s_C[n]$ and $s_G[n]$ add to 1 for all n . Therefore any three of these four binary indicator functions are sufficient for characterizing DNA sequence. The frequency energy of (5.2) can be expressed as

$$E[k] = aE_A[k] + tE_T[k] + cE_C[k] + gE_G[k], \quad k = 0, 1, 2, \dots, N-1 \quad (5.3)$$

where $E_A[k]$, $E_T[k]$, $E_C[k]$ and $E_G[k]$, are the frequency energy of $s_A[n]$, $s_T[n]$, $s_C[n]$ and $s_G[n]$, respectively. Any three of those four binary indicator sequences are sufficient to determine the DNA character string because only one of them equals 1 at one time. This implies that the corresponding four sequences, $E_A[k]$, $E_T[k]$, $E_C[k]$ and $E_G[k]$, are also a redundant set. That is

$$E_A[k] + E_T[k] + E_C[k] + E_G[k] = \begin{cases} 0, & k \neq 0 \\ N, & k = 0 \end{cases}, \quad k = 0, 1, 2, \dots, N-1 \quad (5.4)$$

For pure character strings, the sequence $E_A[k]$, $E_T[k]$, $E_C[k]$ and $E_G[k]$ represents the frequency content of each nucleotide. Combining all four contributions we have

$$SC[k] = |E_A[k]|^2 + |E_T[k]|^2 + |E_C[k]|^2 + |E_G[k]|^2 \quad (5.5)$$

SC is the “DNA energy”, which can be used as a measure of the total spectral content of the sequences at frequency k . In particular the frequency $k = N/3$ corresponds to a period of three samples (the length of each codon) and it has been shown that a protein region in a sequence typically has a peak at that frequency (Anastassiou 2002). If normalized DFT coefficients at frequency $k = N/3$ are defined as

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

$$A = \frac{1}{N} E_A \left[\frac{N}{3} \right], \quad T = \frac{1}{N} E_T \left[\frac{N}{3} \right], \quad C = \frac{1}{N} E_C \left[\frac{N}{3} \right], \quad G = \frac{1}{N} E_G \left[\frac{N}{3} \right] \quad (5.6)$$

then for each segment of length N (multiple of three) of a sequence, there corresponds a complex number

$$W = aA + tT + cC + gG \quad (5.7)$$

W can be considered as a random variable. For properly chosen values of a, t, c and g , it is proved to be a superior predictor of whether or not a given segment belongs to a coding region. The statistical properties of A, T, C and G (their mean and standard deviation) can be obtained by collecting a large number of samples from protein coding regions of a sequence (Anastassiou 2002).

(B). The Optimization Strategy

The discriminatory capability between protein coding regions (with respective random variables A, T, C and G) and random DNA regions need to be maximized. Using a random number generator a sequence of the same length of the statistical sample can be synthesized. So that the corresponding random variables A_R, T_R, C_R, G_R can be obtained. Because of (5.4), $A + T + C + G = 0$, and therefore if any constant value is added to a, t, c and g the value of W remains the same. So the sequences A, T, C and G represent a redundant set, and one of the four coefficients can be set to zero. For example

$$W = aA + tT + gG, \quad c = 0 \quad (5.8)$$

Therefore the optimization problem can be expressed as: Find the complex numbers a, t, c and g that maximize the quantity

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

$$f(a, t, g) = \frac{Ecp\{|aA + tT + gG|\} - Ecp\{|aA_R + tT_R + gG_R|\}}{std\{|aA + tT + gG|\} + std\{|aA_R + tT_R + gG_R|\}} \quad (5.9)$$

where std is the standard deviation and Ecp is expectation operator under the constraints

$$Ecp\{\arg(aA + tT + gG)\} = 0, \quad |a| + |t| + |g| = 1 \quad (5.10)$$

This optimization problem can be solved by using the optimization toolbox available in Matlab, which uses a technique based on iterated random perturbations starting with an initial guess. A sequence of AF092047 from THMD32 (Appendix B) is used as an example.

The resulting optimized parameters for the STFT of the sequence are

$$\begin{aligned} a &= 0.319 - j0.010 \\ t &= -2.803 - j0.652 \\ c &= 0 \\ g &= 0.242 - j0.102 \end{aligned} \quad (5.11)$$

and thus $f(a, t, g) = 2.462$.

After the optimized parameters for all the sequences are obtained, the human- and the mouse-sequences groups are established. The mean optimized parameters for human and mouse sequences are calculated as the mean of these parameters in their own groups, respectively. Table 5.1 shows the final results.

Table 5.1 Mean optimized parameters for different TFDs of human and mouse sequences.

Mean optimized parameters	Human			Mouse		
	STFT	WVD	RSPWVD	STFT	WVD	RSPWVD
a	0.219-j0.021	0.234-j0.018	0.253-j0.014	0.321-j0.034	0.394-j0.041	0.388-j0.059
t	-2.231-j0.321	-2.461-j0.303	-2.577-j0.292	-2.866-j0.273	-2.647-j0.211	-2.906-j0.321
c	0	0	0	0	0	0
g	0.243-j0.082	0.215-j0.091	0.319-j0.074	0.239-j0.126	0.287-j0.102	0.304-j0.121

5.3.2 Identification of Gene Sites

Gene sites are identified by analysis of each segment of frequency energy. Here three parameters are used to define the sample-position of each sites: τ_k^s , τ_k^c and τ_k^e , which means the start, the center and the end sample-position of the gene sites respectively. In this application, the position where the frequency energy is the highest is defined as the ‘center’ sample-position rather than the median point. So the k -th gene site, $p(k)$, between marks τ_k^s and τ_k^e , is given by

$$p(k) = r[\tau_k^s : \tau_k^e] \quad (5.11)$$

where $r[\tau_k^s : \tau_k^e]$ denotes a continuous segment beginning at τ_k^s and ending at τ_k^e , which includes the sample τ_k^s but excludes the sample τ_k^e . So the exon length, $L_{p(k)}$, is given by

$$L_{p(k)} = \tau_k^e - \tau_k^s \quad (5.12)$$

The maximum instantaneous frequency of the frequency energy is used to detect the possible position of a gene site. It is accepted as a candidate of gene site. So an algorithm is designed to search the frequency peaks. The search is be processed frame by frame and the details of the algorithm are described in Algorithm 4.2.

Although the peak detection method is acceptable for identifying the position whose energy is higher than that of adjacent positions, the result is not usually satisfying. Extra processing is used. Because of the segmentation of sequences and the frame based TFD computation, it is unavoidable that some gene sites are split into two adjacent frames and this will lead to inaccurate identification of gene site. Algorithm 4.2 cannot detect the first gene site. So frame-merging algorithm is used to judge whether the problem happens and

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

correct it if so. The frame-merging algorithm is applied frame by frame and is given in Algorithm 4.3.

Algorithm 4.2: The Peak-Searching Algorithm

Step (0): Initialize $m = 2$ (m is the sample point), $k = 0$, $L = \text{length of the frame}$.

Step (1): If m meets the end of the frame, then stop.

Step (2): If $E_x(m) > E_x(m-1)$ and $E_x(m) > E_x(m+1)$, then $k = k + 1$, $\tau_k^c = m$ and go to Step (3); else $m = m + 1$, go to Step (1).

Step (3): Set i from 1 to $(m-2)$ and search for whether

$$E_x(m-i-1) > E_x(m-i).$$

If found, $\tau_k^s = m-i$; if not found and i reaches $m-2$, then $\tau_k^s = 1$.

Step (4): Set j from 1 to $(L-m-1)$ and search for whether

$$E_x(m-i-1) > E_x(m-i).$$

If found, $\tau_k^s = m+j$; if not found and j reaches $L-m-1$, then

$\tau_k^s = L$. Set $m = \tau_k^s + 1$ and go to Step (1).

Since not all periodic components belong to gene sites, all the pseudo gene sites (pulses which are wrongly judged as gene sites) must be removed from gene-site candidates. In this application the Maximum amplitude difference (MAD) method is used to remove the pseudo gene sites (Ross *et al.* 1974). The MAD calculates the range of amplitude change in each frame. The assumption is that amplitude of gene sites should change much more than noise buried in the signals. The MAD, denoted by M_{ad} , is given

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

by

$$M_{ad} = |S_{\max} - S_{\min}| \quad (5.13)$$

where $S_{\max}$ and $S_{\min}$ are the maximum and minimum amplitudes of the signal at time t .

In this case, a threshold $\theta (0 < \theta < 1)$ is applied here. If $M_s < \theta M_{ad}$, where M_s is the amplitude of a pulse in a period, the pulse is then decided as a pseudo gene site and is removed. Different values of θ can be set to meet the properties of different signals. However, in some situations one gene site is much weak than the others. The MAD will ignore it as a pseudo one. In this experiment, $\theta = 0.8$ gives the most satisfying result.

Algorithm 4.3: The Frame-Merging Algorithm

- Step (0): Calculate the position of the pitch-pulse candidate: τ_k^s , τ_k^c and τ_k^e .
- Step (1): If τ_k^c reaches the beginning of the frame, copy the previous frame and the current frame into memory. Re-calculate the TFD with the sum of the two frame lengths. Identify the pitch-pulse candidates by the gene site detection method.
- Step (2): If τ_k^c reaches the end of the frame, copy the current frame and the following frame into memory. Re-calculate the TFD with the sum of the two frame lengths. Identify the pitch-pulse candidates by the gene site detection method.

5.3.3 Result Explanations

To explain how the gene-finding programs using time-frequency analysis work, the

DNA sequence of AF099730 (Appendix A) is used. Using the optimization coefficients in Table 5.1, the value of $|W|^2$ in (5.7) is used to evaluate the magnitude for a DNA segment containing 4000 nucleotides. Fig. 5.1-5.3 are the results. From former research, the DNA stretch is found containing a gene with five exons. So Fig. 5.1 can identify these exons by peaks. Fig. 5.2 is the result using the frequency energy of WVD. Comparing with Fig. 5.1, the result of WVD can display more concentrated peaks. Fig. 5.3 shows the results of RSPWVD, the improvement of in removing noise can easily be noted, and the peaks are as concentrated as those in Fig. 5.2.

Fig. 5.1 $|W|^2$ of the DNA sequence of AF099730 using STFT

Fig. 5.2 $|W|^2$ of the DNA sequence of AF099730 using WVD

Fig.5.3 $|W|^2$ of the DNA sequence of AF099730 using RSPWVD

5.4 Measuring Predictive Accuracy

In order to evaluate the predictive accuracy of a gene-finding program, the exons predicted by the program need to be compared with the actual coding exons, as annotated in the GenBank record under the 'CDS' feature. Annotated non-coding exons are not considered, since the programs under analysis do not predict them (Kulp *et al.* 1996). From this comparison, nucleotide and exon level accuracy measures are calculated.

5.4.1 Nucleotide Level Accuracy

The values TP (true positives), TN (true negatives), FP (false positives) and FN (false negatives) are defined as follows (Burset and Guigo 1996):

TP - the number of coding nucleotides predicted correctly as coding.

TN - the number of non-coding nucleotides predicted correctly as non-coding.

FP - the number of non-coding nucleotides predicted incorrectly as coding.

FN - the number of coding nucleotides predicted incorrectly as non-coding.

The sensitivity, Y_n , is defined as the proportion of coding nucleotides that have been correctly predicted as coding, i.e.,

$$Y_n = \frac{TP}{TP + FN} \quad (5.14)$$

The specificity, Y_p , is defined as the proportion of predicted coding nucleotides that are actually coding, i.e.,

$$Y_p = \frac{TP}{TP + FP} \quad (5.15)$$

These are widely used measurements of accuracy for gene prediction programs.

Both Y_n and Y_p range independently over $[0,1]$, with perfect prediction occurring

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

only when both measures are equal to 1. Each of these measures is not sufficient by itself because perfect sensitivity can be obtained if all the nucleotides were predicted as coding and perfect specificity if all nucleotides were predicted as non-coding.

A single measure that captures both specificity and sensitivity called the correlation coefficient (CC) is defined as

$$CC = \frac{(TP * TN) - (FN * FP)}{\sqrt{(TP + FN) * (TN + FP) * (TP + FP) * (TN + FN)}} \quad (5.16)$$

This measure has been extensively used for evaluating gene structure prediction programs, but it has the undesirable property that it is not defined for some sequences (e.g. if there is not any coding region in an input sequence or an input sequence has been predicted to be entirely non-coding). A measure with similar characteristics, but defined under any circumstance, is the approximate correlation (AC), introduced in (Burset and Guigo 1996), defined as:

$$AC = (ACP - 0.5) * 2 \quad (5.17)$$

where ACP is the average conditional probability defined as:

$$ACP = \frac{1}{4} \left(\frac{TP}{TP + FN} + \frac{TP}{TP + FN} + \frac{TN}{TN + FN} + \frac{TN}{TN + FP} \right) \quad (5.18)$$

CC and AC range over $[-1, 1]$ and usually are close to each other whenever CC is defined.

5.4.2 Exon Level Accuracy

Exon level prediction is also estimated by sensitivity and specificity, but in this case true positives are exactly predicted exons (identical to an annotated exon). The formulas for

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

exon level sensitivity (ES_n) and specificity (ES_p) are:

$$ES_n = \frac{TE}{AE} \quad (5.19)$$

$$ES_p = \frac{TE}{PE} \quad (5.20)$$

$$ES = (ES_n + ES_p)/2 \quad (5.21)$$

where TE (true exons) is the number of exactly predicted exons, and AE and PE are the numbers of annotated and predicted exons, respectively. Similar to nucleotide level accuracy, these measures cannot be used alone and usually their average is used as a reliable measure of exon level accuracy (Burset and Guigo 1996).

Sometimes, knowing just the proportion of the exactly predicted exons may underestimate the performance of the program. To get a better estimate of the prediction accuracy of the analyzed programs considered other categories of predicted and annotated exons can be considered. Predicted exons can be divided into four categories: exactly predicted, partially predicted (only one exon boundary is correctly predicted), overlapped (neither exon boundary is correct, but it overlaps an actual exon) and wrongly predicted (does not overlap any actual exon). Analogously, annotated exons can be divided in those that are exactly predicted, partially predicted, overlapped and missed (not overlapped with any predicted exon).

Proportions on the exon level sensitivity are defined as

CR_a - proportion of annotated exons that are correctly predicted.

CR_p - proportion of predicted exons that are exactly correct.

PC_a - proportion of partially predicted annotated exons.

PC_p - proportion of predicted exons that are partially correct.

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

OL - proportion of predicted exons that overlap actual exons.

ME - proportion of missed exons.

WE - proportion of wrong exons.

It shows that this measure is not defined when a program does not predict any exons in a sequence. In this case, 0 is assigned to both ES_n and ES_p , and this sequence will not be considered when calculating the average values for the whole dataset. Even though ES_n is defined for every sequence containing actual exons it is not averaged over the sequences for which ES_p is not defined in order to obtain a more realistic relationship between two measures (in real genomic sequences the sequences without exon prediction would be less common than the sequences without actual exons).

On the sequence level CR_a and CR_p are identical to ES_n and ES_p , except for $PE = 0$ when $CR_p = 0$, but they are averaged over all sequences in the dataset. For this reason, CR_a and CR_p are used as the more credible measures when programs are run on the subsets of the dataset and when sequences without predictions can strongly influence the results for ES_n and ES_p .

5.5 Results and Evaluation

Sequences from the HMD132 dataset are analyzed using the gene prediction programs. For each sequence, the exons predicted on the forward strand are compared to the actual coding exons, as annotated in the GenBank ‘CDS’ feature (predictions for the reverse strand were ignored). For each gene-finding program, a Perl script is written that parses the prediction from the program and compares predicted exons to the annotated exons obtained

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

from the annotation file for HMD132. The separate scripts are necessary since the outputs of the programs tested are different for each program. For each sequence in HMD132, all the accuracy measures are calculated and written to the output file. The final output from each Perl script is the averaged nucleotide and exon level prediction accuracy measures on the HMD132 dataset.

Averaging by sequence is chosen, where measures are first calculated for each gene then averaged over all genes, as opposed to averaging by base, where measures are summed for all sequences and then averaged by nucleotide or by exon, depending on the measure type. The measures are averaged only over sequences for which they are defined. The accuracy measures for all of the programs analyzed, averaged for the entire HMD132 dataset are presented in Table 5.2 and 5.3.

The next step is to examine accuracy as a function of various sequence or prediction features. For each characteristics the dataset is divided into the subsets exhibiting different value ranges or types of the characteristic examined and the accuracy measures are calculated and averaged over all the sequences belonging to a particular subset. For each of the programs and each of the sequence or prediction features a separate Perl script is written. The results are presented in Tables 5.3-5.7.

It is apparent that all the three traditional gene-finding programs (GeneMark.hmm, Genscan and HMMgene) have substantially higher prediction accuracy than the newly developed gene-finding programs based on STFT, WVD and RSPWVD. The best approximate correlations are Genscan and HMMgene, with $AC = 0.91$. However, for the RSPWVD-based gene-finding program, the result is also satisfying with $AC = 0.88$. On

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

the exon level, ES has increased from 0.41 for WVD-based gene-finding program to 0.76 for HMMgene. It shows that all the three traditional gene-finding programs and the RSPWVD-based gene-finding program have high sensitivity and specificity. However, the results of STFT and WVD-based gene-finding program are not acceptable.

Table 5.2 Nucleotide level accuracy.

Programs	Number of Sequences analyzed	Nucleotide accuracy				Exon accuracy	
		Y_n	Y_p	AC	CC	ES_n	ES_p
GeneMark.hmm	132 (0)	0.87	0.89	0.84 ± 0.18	0.83	0.53	0.54
Genscan	132 (3)	0.95	0.90	0.91 ± 0.12	0.91	0.70	0.70
HMMgene	132 (5)	0.93	0.93	0.91 ± 0.13	0.91	0.76	0.77
STFT	132(10)	0.62	0.57	0.61 ± 0.15	0.60	0.47	0.47
WVD	132(16)	0.55	0.53	0.68 ± 0.12	0.67	0.42	0.45
RSPWVD	132(4)	0.91	0.92	0.88 ± 0.09	0.88	0.67	0.68

Table 5.3 Exon level accuracy.

Programs	Exon accuracy							
	ES_n	ES_p	$(ES_n + ES_p)/2$	ME	WE	PC_a	PC_p	OL
GeneMark.hmm	0.53	0.54	0.54 ± 0.36	0.13	0.11	0.29	0.27	0.09
Genscan	0.70	0.70	0.70 ± 0.32	0.08	0.09	0.21	0.19	0.02
HMMgene	0.76	0.77	0.76 ± 0.30	0.12	0.07	0.14	0.14	0.02
STFT	0.47	0.47	0.43 ± 0.38	0.18	0.11	0.35	0.39	0.05
WVD	0.42	0.45	0.41 ± 0.27	0.09	0.21	0.49	0.46	0.08
RSPWVD	0.67	0.68	0.66 ± 0.25	0.07	0.06	0.15	0.12	0.03

The accuracy measures at the nucleotide level is as high as 0.91 for AC of Genscan and HMMgene. However, there is a gap between AC and ES . (5.19) and (5.20) show that in calculating ES_n and ES_p only exons with both boundaries predicted correctly will be considered. An almost perfectly predicted exon, covering the whole sequence of an actual exon, but exceeding the splicing site by just few nucleotides will not be counted in TE . In order to predict the exact boundaries of an exon, the frame-merging algorithm is

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

used in all the three TFD-based gene finding programs for splice site detection, and extra processing is made. So the results are better than GeneMark.hmm, Genscan and HMMgene programs. Using very simple time-frequency analysis techniques the exon locations can be predicted in a stretch of DNA sequence. This method needs to set different optimization value in different problems. New parameters a , t , c and g should be obtained to maximizing the predictive power of a reading frame predictor with complicated calculations. In the algorithm the optimization parameters are computed directly from the difference of frequency energy between the training sequence and the random sequence. They are more suitable for displaying the information for the further detection.

In Table 5.2 and 5.3, for each sequence in the HMD132 dataset, the exons predicted on the forward strand are compared to the annotated exons. The standard measures of predictive accuracy on nucleotide and exon level are calculated for each sequence and averaged over all sequences. This is done separately for each of the programs tested. The value in parentheses of Table 5.2 is the number of sequences in which absence of gene is predicted.

5.5.1 G+C Content

The human genome and the genomes of other warm-blooded vertebrates are not a structurally homogenous sequence of nucleotides (Bernardi 1993). Instead, they have a mosaic of isochores, long DNA regions (>300 kb, on average) whose base composition is locally homogenous, but varies significantly between disjoint regions. The genome is usually divided into five different compositional categories: L1, L2 (A+T rich regions), H1,

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

H2 and H3 (G+C rich regions) in increasing order of G+C%. It has been observed that L1+L2 constitute approximately 60% of human genome, H1+H2 30% and H3 only 5%. These compositional regions widely vary in gene density: Zoubak *et al.* (1996) calculated that L1+L2 regions have a relative gene concentration of 4%, H1+H2 20% and H3 76% respectively. This means that the gene density in very G+C rich DNA segments is almost 20 times higher than in A+T rich regions.

Important structural properties of genes are found to be strongly correlated with G+C content (Duret *et al.* 1995). Genes from G+C poor isochores code for proteins that are on average longer than those from G+C rich isochores. Intronic DNA is on average three times longer in L1+L2 than in H3 and the number of introns per gene is higher in L1+L2 than in H3. Burset and Guigo (1996) have shown in their analyses that gene-finding programs usually perform worse when the G+C content is low. The proposed reasons for this anomaly are that G+C rich genes have stronger codon bias that makes them easier to identify and that they are more frequent than the genes in A+T rich isochores. In another study Guigo and Fickett (1995) have shown that coding statistics used by gene-finding programs (codon, dicodon and hexamer frequency) are strongly dependent on G+C content.

Table 5.4 shows the nucleotide and exon level accuracy versus G+C content. The HMD132 dataset was partitioned according to the G+C% content of the sequences. For each program, *AC* and *ES* are averaged over all sequences belonging to a particular partition for which it is defined. Among these programs tested, GeneMark.hmm, Genscan and HMMgene have the lowest accuracy measures averaged on the sequences with G+C content between 40 and 50 %. However, different G+C content has little impact on the

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

accuracy of the detection results of all the STFT, WVD and RSPWVD-based methods.

Actually the detection results of the TFD-based methods are independent of G+C content.

Table 5.4 Nucleotide and Exon Level Accuracy versus G+C content.

C+G content	< 40 %		40 –50 %		50 –60 %		> 60%	
	<i>AC</i>	<i>ES</i>	<i>AC</i>	<i>ES</i>	<i>AC</i>	<i>ES</i>	<i>AC</i>	<i>ES</i>
GeneMark.hmm	0.79	0.48	0.80	0.46	0.87	0.62	0.85	0.58
Genscan	0.94	0.80	0.85	0.66	0.91	0.74	0.88	0.70
HMMgene	0.91	0.76	0.90	0.73	0.92	0.79	0.91	0.77
STFT	0.58	0.44	0.59	0.44	0.60	0.44	0.60	0.45
WVD	0.59	0.40	0.61	0.39	0.60	0.41	0.65	0.41
RSPWVD	0.87	0.67	0.89	0.67	0.88	0.68	0.87	0.67

5.5.2 Exon Length

The length distributions of different gene elements differ considerably between each other. Exons have significant functional constraints. The exon length plays an important role in proper splicing and inclusion in the mature mRNA (Dominski and Kole 1991). These constraints have shaped the exon length distribution quite differently from a geometric distribution. The length distribution depends on the exon type. Internal exons have length distribution close to a Gaussian distribution with a broad peak between 100 and 170 bp. Length distributions for initial and terminal coding exons are not recognizable statistical distributions. They are still substantially peaked around 60 and 160 bp respectively (Burge 1997), but do not have a steep drop-off in density after 300 bp. Both types of exons are more variable in length than internal exons and their calculated means are 134 bp for initial exons and 198 bp for terminal exons. In the HMD132 dataset it is 207 bp for initial and 265 bp for terminal. Here exons have been grouped by both their annotated length and their predicted length. The accuracy measures are averaged in each

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

group. Since all the GeneMark.hmm, Genscan and HMMgene programs tested in this analysis use explicit duration HMMs which have length distribution associated with each state of the model, it is interesting to see how these distributions influence the accuracy of their exon prediction.

Table 5.5 shows the accuracy versus exon length. The HMD132 dataset is partitioned according to the length of the annotated exons. For each program, the proportion of real exons that are correctly predicted (the upper number), CR_a , and the proportion of predicted exons that are correct (the number in parentheses), CR_p , are averaged over all sequences belonging to a particular partition. From Table 5.5 it can be observed that all the six programs have a very low proportion of correctly predicted short exons. They then slowly but monotonically rises with the length of annotated exons. In nearly all the programs, exons are most accurately predicted if their length ranges between 75 and 200 nucleotides. It is easy to understand why the identification accuracy of TFD-based gene-finding programs rises with the length of annotated exons, from the definition of the optimization mapping, the longer the range of exons, the higher the energy.

Table 5.5 Exon Accuracy versus exon length

Programs	Length range of exons in bp				
	0 – 24	25 – 49	50 – 74	75 – 99	100 – 199
GeneMark.hmm	0.05 (0.12)	0.39 (0.51)	0.60 (0.58)	0.77 (0.72)	0.75 (0.73)
Genscan	0.18 (0.29)	0.45 (0.81)	0.68 (0.79)	0.89 (0.85)	0.84 (0.76)
HMMgene	0.23 (0.42)	0.59 (0.76)	0.64 (0.75)	0.79 (0.77)	0.80 (0.85)
STFT	0.06(0.38)	0.20(0.45)	0.44(0.50)	0.50(0.47)	0.58(0.54)
WVD	0.08(0.31)	0.18(0.36)	0.40(0.44)	0.47(0.43)	0.55(0.48)
RSPWVD	0.14(0.33)	0.46(0.66)	0.71(0.81)	0.83(0.86)	0.90(0.88)

5.5.3 Exon Type and Signal Prediction

Table 5.5 shows the exon accuracy versus exon type. The HMD132 dataset is partitioned according to the type of the annotated exons. For each program, CR_a (the upper number) and CR_p (the number in parentheses) are averaged over all sequences belonging to that particular partition. It summarizes accuracy measures for different exon types. There is a striking difference between the proportion of correctly predicted internal exons on one side and the proportion of correctly predicted initial and terminal exons on the other side. For single-exon genes, the difference is similar for GeneMark.hmm and Genscan programs in the sense that the CR is usually significantly lower than that for internal exons. However, they have very high values for PC . The difference is that single exon genes are very rarely missed and the proportion of missed exons of this type is the lowest among all exon types. All the three TFD-based programs perform well in identifying single exon gene, since it is easier to detect one single peak in a sequence. HMMgene and RSPWVD-based gene-finding program are the only two programs which are almost equally successful in predicting exons. They also have high proportion of correctly predicted exons (CR_p) for initial, internal, terminal and single exons among all other programs.

Table 5.6 Exon accuracy versus exon type

Programs	Exon type			
	Initial	Internal	Terminal	Single
GeneMark.hmm	0.40 (0.48)	0.78 (0.72)	0.52 (0.51)	0.30 (0.65)
Genscan	0.57 (0.71)	0.87 (0.76)	0.67 (0.73)	0.63 (0.83)
HMMgene	0.68 (0.72)	0.78 (0.83)	0.70 (0.73)	0.77 (0.79)
STFT	0.34(0.42)	0.38 (0.53)	0.35 (0.46)	0.47 (0.49)
WVD	0.28 (0.22)	0.31 (0.26)	0.20 (0.29)	0.51 (0.47)
RSPWV	0.67 (0.73)	0.69 (0.65)	0.68(0.73)	0.72 (0.70)

5.6 Combining Time-Frequency Analysis and Traditional Gene-Finding Programs

New programs are developed for gene finding in sequences using time-frequency analysis techniques. Although the TFD-based gene-finding programs can identify genes, only the RSPWVD-based program performs well compared to the traditional gene-finding programs. Instead of attempting to improve some components of those TFD-based gene-finding programs or designing a new one, a new program can be developed by combining the best features from two gene-finding programs.

5.6.1 Motivation

After extensive simulations, the following two important points are observed:

- (i) It is difficult for the traditional gene-finding programs, such as GeneMark.hmm and Genscan, to identify initial, internal, terminal and single exons.
- (ii) Table 5.2 and 5.3 show the proportion of partially predicted annotated exons and predicted exon from STFT and WVD based gene-finding programs is much higher than other programs. It implies that these two programs can detect the existence of exons but they are not able to detect the exact locations.

Thus if those two kinds of gene-finding programs are combined, improvement is expected at both the nucleotide and exon levels. Considering the results of gene-prediction-program evaluation in Section 5.5, it shows that HMMgene is the best candidate of the three traditional gene-finding programs tested. It has the best overall prediction accuracy and almost the highest number of correctly predicted exons. The newly

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

combined gene-finding programs are adapted from the algorithms in Section 5.3. The difference is that a TFD-based gene-finding program is only used to decide the approximative exon locations, while the HMMgene which performs the best among the three traditional gene-finding programs identifies the exact exon locations. The algorithm of optimized parameters computation is exactly the same as the algorithm in Section 5.3.1. The gene-site identification algorithm in Section 5.3.2 is used to decide the maximum instantaneous frequency of the frequency-energy and the frequency energy peak, $E[k]$ in k -th sample of sequence. Once $E[k]$ is decided, the search is processed using HMMgene. The algorithm is given as follow:

Algorithm 4.4: The combined gene-finding algorithm

- Step (0): Initialize k , $L = \text{length of the sequence } s[n]$.
- Step (1): If $200 \leq k \leq L - 200$, then do HMMgene searching in the segment $[s[k - 200], s[k + 200]]$.
- Step (2): If $k \leq 200$, then do HMMgene searching in the segment $[s[1], s[200]]$.
- Step (3): If $k \leq L - 200$, then do HMMgene searching in the segment $[s[L - 200], s[L]]$.
- Step (4): end.

Considering the characteristics of HMD132, almost every exon in HMD132 has a length of less than 400. So the value of 400 is enough for exon identification.

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

5.6.2 Results and Evaluation

The HMD132 dataset is again used for the evaluation. For each sequence, the exons predicted on the forward strand (predictions for the reverse strand are ignored) are compared to the actual coding exons, as annotated in the GenBank ‘CDS’ feature.

Table 5.8 shows the results of the combined gene-finding programs. For each sequence in the HMD132 test set, the forward strand exons in the default outputs of the programs tested are compared to the annotated exons. The standard measures of predictive accuracy on nucleotide and exon level are calculated for each sequence and averaged over all sequences for which they are defined. The second column gives the number of sequences where no prediction is made. The numbers in parenthesis in the last two columns are the actual numbers of missed and wrong exons, respectively.

It is observed that each of the combined methods outperforms individual Genscan and HMMgene in all categories except for nucleotide level sensitivity (Y_n) and proportion of missed exons (ME). The results in Table 5.6 suggest that each of three methods improve specificity more than sensitivity at both the nucleotide and exon levels.

Table 5.7 Results of the combined gene-finding programs.

Methods	Number of sequences of no prediction	Nucleotide accuracy			Exon accuracy				
		Y_n	Y_p	AC	ES_n	ES_p	$(ES_n + ES_p)/2$	ME	WE
HMMgene	5	0.93	0.93	0.91	0.76	0.77	0.76	0.12	0.07
With STFT	3	0.92	0.93	0.93	0.79	0.78	0.76	0.10	0.08
With WVD	15	0.91	0.92	0.92	0.78	0.76	0.75	0.09	0.09
With RSPWVD	3	0.96	0.95	0.93	0.85	0.86	0.82	0.04	0.03

5.7 Summary

Novel gene-finding programs based on simple time-frequency analysis techniques to predict the exons locations in DNA sequences are developed. Three TFDs, the STFT, the WVD, and the RSPWVD, are used to obtain the frequency energy of the sequences for identification. The results indicate that the TFD-based gene-finding programs can identify genes like the traditional gene-finding programs such as GeneMark.hmm, Genscan and HMMgene programs. Among all the three developed programs, the RSPWVD-based gene-finding program has almost the same or higher accuracy performance than the three traditional gene-finding programs, GeneMark.hmm, Genscan and HMMgene programs. The other two TFD-based gene-finding programs are not able to perform well for the identification job.

The G+C content of the sequence has little impact on the accuracy of the detection results from all the three TFD-based programs. However, the behavior of a traditional gene-finding program, such as GeneMark.hmm, Genscan and HMMgene programs, on the sequences with different G+C content is not consistent. Their accuracy appears to be slightly dependent on the G+C content.

The accuracy of exon prediction is dependent on the length of the exon. The general trend of the programs is to have a very low proportion of correctly predicted short exons. The proportion of correctly predicted exons then rises with the length of annotated exons.

All the three TFD-based programs perform well in identifying single exon gene. Among all the programs analyzed, HMMgene and RSPWVD-based gene-finding program are the only two programs that are almost equally successful in predicting exons, which

Chapter 5: Computational Gene-Finding Using Time-Frequency analysis Algorithms

also have high proportion of correctly predicted exons (CR_p) for initial, internal, terminal and single exon among all other programs.

Combining TFD-based with HMMgene programs, they successfully improve prediction accuracy. The improvements are obtained at both the nucleotide and exon levels. The major advantage of the method is the elimination of many false exon predictions. It directly improves the specificity at both levels.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

The time-frequency analysis is a powerful tool in signal processing. It reveals the properties of a signal in both time and frequency domains. Allowing one to see how its amplitude varies with time and also how its energy changes with frequency. There are two main factors affecting the performance of a TFD: the concentration of the TFD which decides the accuracy of the representation and the cross-terms of the TFD which decide the performance in multicomponent signals. Optimized TFDs, such as the RSPWVD, can perform well in time-frequency representation of signals. New algorithms, such as simplified and time-recursive RSPWVD algorithms, can compute the RSPWVD with less computational cost.

The amino acids critical to the function of the proteins can be located in the resulting spectrograms of their TFDs. This makes the time-frequency analysis techniques useful for the identification of the critical parts of the proteins. Both the STFT and the RSPWVD are able to locate protein critical parts given a protein signal based on amino acid hydrophobicities. It shows that hydrophobicity is a suitable to be used in protein critical parts localization via time-frequency analysis. The results also reveal that the RSPWVD is significantly better than the STFT in performing such a task.

Novel gene-finding programs based on simple time-frequency analysis can be used to predict the exons locations in DNA sequences like the traditional gene-finding programs

such as GeneMark.hmm, Genscan and HMMgene programs. The performance of the RSPWVD-based gene-finding program is almost the same as that of Genscan and HMMgene, while the other two TFD-based gene-finding programs behave badly for the identification job.

In this thesis, problems, advantages, and new method in the field of time frequency analysis of biomedical signal processing haven are investigated. In this research work, the following contributions have been made:

(A). New Efficient Time-Frequency Analysis Algorithm

New efficient time-frequency analysis algorithms are developed. They depend on the usage of time-recursive approach in computing TFDs. Based on the WVD algorithms and recursive methods, two algorithms, the simplified and time-recursive RSPWVD algorithms, are proposed to compute the RSPWVD of a given signal. For all the algorithms proposed, analysis on computational cost is made.

(B). Tools Based on Time-Frequency Analysis to Locate Critical Amino Acid Residues in Protein Sequences

Tools based on the newly proposed recursive time-frequency analysis algorithms are developed to locate critical amino acid residues in protein sequences. Protein signals are constructed based on amino acid hydrophobicity and the resulting spectrograms are analyzed to locate important amino acid domains. The amino acids critical to the function of these proteins are located in the resulting time-frequency representations.

(C). Tools Based on Time-Frequency Analysis for Gene-Finding

Tools based on the newly developed recursive time-frequency analysis algorithms are designed for gene-finding. Together with the new time-frequency analysis tools, three frequently used gene-finding programs such as GeneMark.hmm, Genscan and HMMgene are used for comparison. A small dataset named HMD132 is constructed to evaluate the performance of these algorithms. Further integration of prediction method by combining both the time-frequency analysis and the traditional gene-finding programs are made for higher accuracy.

6.2 Future Work

Although the time-frequency analysis techniques have been well studied and extensively applied in many fields like speech, the audio processing, radar systems and brain signal localizations in recent years, there still remain many possible areas for future research. The future work described here is mainly concentrated on the possible applications of time-frequency analysis techniques in biomedical research. Possible areas of future research are listed below:

(A). Appropriate TFDs for Corresponding Applications

Different TFDs have different characteristics. There is no single TFD suitable for all applications. Precise mathematical measures of TFDs need to be formulated, so that only appropriate TFDs are chosen for corresponding applications, especially in biomolecular signal processing.

(B). More Efficient TFD Algorithms

Although some efficient algorithms for TFD-computation have been developed, the computation time is still high for the RSPWVD algorithm. Further improvement is still possible, especially at algorithmic or hardware level. Improvement on parallelization of algorithm and hardware level architecture is also another possibility.

(C). Automatic TFD-Based Tools for Identification of Critical Parts of Proteins

Newly developed tools based recursive time-frequency analysis algorithms are able to locate critical amino acid residues in protein sequences. However, these tools are not complete. The amino acids critical to the function of the proteins have to be picked out from the resulting spectrograms manually. So tools based on pattern classification are needed to do the detection automatically.

(D). Intact Computational Gene-Finding Tools Using Time-Frequency Analysis

Although the proposed gene-finding programs can predict the exons locations in DNA sequences like the traditional gene-finding programs, such as GeneMark.hmm, Genscan and HMMgene. Only the RSPWVD-based program (among STFT, WVD and RSPWVD) has the same performance as the traditional gene-finding programs. It is believed that the concentration and clearness of the RSPWVD representation bring good identification results. However, the efficient time-recursive RSPWVD algorithm, the computational cost of RSPWVD are still higher than the other two TFDs. Future attempt can be to improve the identification algorithm of the program. This is likely to improve the results of STFT and

WVD-based gene-finding programs. This idea can also be applied to the combined programs of TFD-based gene-finding programs with HMMgene gene-finding program.

Further methods for modifying the methods presented to include more than two programs can also be made.

Finally, the newly assembled HMD132 dataset is currently the more suitable set of genomic sequences for the purposes of gene-finding evaluation and analysis. Nevertheless, the sequences in this dataset are not typical of sequences submitted to gene prediction algorithms for analysis. They are relatively short sequences containing only one complete gene which is usually less complex than a typical mammalian gene (shorter, less exons), and flanked by unrealistically short stretches of intergenic regions. More realistic sequences that are well annotated (which is currently not the case) will become available and thus should be used to reevaluate gene-finding programs. A more realistic dataset could also be obtained at the moment by including sequences containing partial and multiple genes. However, the problems associated with the selection of these sequences are doubtful annotation (many partial genes are not in the final finished form) and there is increased difficulty in the verification of the annotation.

Appendix A: The HMD132 Dataset

Accession	Definition line
AB016625	Homo sapiens OCTN2 gene
AF008216	Homo sapiens candidate tumor suppressor pp32r1 (PP32R1) gene
AF092047	Homo sapiens homeobox protein Six3 (SIX3) gene
AF096303	Homo sapiens putative sterol reductase SR-1 (TM7SF2) gene
AF019563	Homo sapiens DAP12 gene
AB012922	Homo sapiens MTA1-L1 gene
U25134	Human carbonic anhydrase V (CA-V) gene
U17081	Human fatty acid binding protein (FABP3) gene
AB021866	Homo sapiens KIP gene
AF039704	Homo sapiens lysosomal pepstatin insensitive protease (CLN2) gene
AF082802	Homo sapiens telencephalin (ICAM5) gene
AF039954	Homo sapiens CC chemokine LCC-1 precursor gene
AB018249	Homo sapiens gene for CC chemokine LEC
AF099731	Homo sapiens connexin 31.1 (GJB5) gene
AF099730	Homo sapiens connexin 31 (GJB3) gene
AF039401	Homo sapiens calcium-dependent chloride channel-1 (hCLCA1) gene
AF084941	Homo sapiens pre-T cell receptor alpha chain 1 precursor gene
AF059675	Homo sapiens putative RNA helicase Ski2w (SKI2W) gene
AF007189	Homo sapiens claudin 3 (CLDN3) gene
AF016898	Homo sapiens B-ATF gene

Appendix A: The HMD132 Dataset

AF076214	Homo sapiens prophet of Pit1 (PROP1) gene
AB012113	Homo sapiens gene for CC chemokine PARC precursor
AB019534	Homo sapiens gene for cathepsin L2
AF080237	Homo sapiens Rho GDP-dissociation inhibitor gamma (ARHGDIG) gene
AF071596	Homo sapiens apoptosis inhibitor (IEX-1L) gene
U43842	Homo sapiens bone morphogenetic protein-4 (hBMP-4) gene
AF053455	Homo sapiens tetraspan TM4SF (TSPAN-5) gene
AF071216	Homo sapiens beta defensin 2 (HBD2) gene
AF058761	Homo sapiens ribosomal protein S12 gene
Y16791	Homo sapiens hHa5 gene
AB016492	Homo sapiens hJTB gene
AF001689	Homo sapiens ribosomal protein L23A (RPL23A) gene
AF029081	Homo sapiens 14-3-3 sigma protein promoter and gene
U96846	Human natural killer protein group 2-F (NKG2-F) gene
AF027152	Homo sapiens P450 25-hydroxyvitamin D-1 alpha hydroxylase
U55058	Human uroguanylin gene
AF016190	Mus musculus connexin-36 (Cx36) gene
U40494	Mus musculus augments of liver regeneration (Alr) gene
AF057156	Mus musculus small proline-rich protein 1A (Sprr1a) gene
AF006668	Mus musculus uroguanylin gene
AF002719	Mus musculus secretory leukoprotease inhibitor gene
AF019045	Mus musculus vesicular acetylcholine transporter gene
AF074912	Mus musculus chemokine receptor CX3CR1 gene
U83462	Mus musculus serotonin N-acetyltransferase (AANAT) gene
AF077860	Mus musculus helix-loop-helix protein Id2 gene

Appendix A: The HMD132 Dataset

AF043289	Mus musculus muscle-specific serine kinase 1 gene
AF037313	Mus musculus potassium inward rectifier 6.2 (Kir6.2) gene
AF035672	Mus musculus MHC class I related protein 1 (MR1) gene
AF001797	Mus musculus glucosidase I gene
U44436	Mus musculus bradykinin B1 receptor gene
AB017361	Mus musculus Kip/Cip gene
AF059211	Mus musculus cholesterol 25-hydroxylase gene
AF069778	Mus musculus A3 adenosine receptor gene
AF035684	Mus musculus beta chemokine TCA4 gene
AF024513	Mus musculus menin (Men1) gene
AF045662	Mus musculus cell cycle checkpoint control protein Mrad9 gene
U96809	Mus musculus chromatin structural protein homolog (Supt4h) gene
AF093853	Mus musculus 1-Cys peroxiredoxin protein 2 gene
AF092536	Mus musculus heat shock protein hsp40-3 gene
AF074856	Mus musculus C1q/MBL/SPA receptor C1qRp (C1qrp) gene
AF039602	Mus musculus extracellular superoxide dismutase (SOD3) gene
AF057301	Mus musculus keratocan (Ktcn) gene
AF029791	Mus musculus UDP-Gal:betaGlcNAc beta 1,3-galactosyltransferase-II
AF026469	Mus musculus ubiquitin-specific protease (Unp) gene
AF031426	Mus musculus small unique nuclear receptor co-repressor (SUN-CoR)
U84903	Mus musculus L23 mitochondrial-related protein (L23mrp) gene
AF016697	Mus musculus chemokine receptor gene
AF022651	Mus musculus macrosialin gene
AF031826	Mus musculus leukocystatin gene
AF053757	Mus musculus complement C3a anaphylatoxin receptor (C3ar) gene

Appendix A: The HMD132 Dataset

AF015881	Mus musculus nuclear factor erythroid-related factor 1 (Nrf1) gene
AF046000	Mus musculus rod cGMP phosphodiesterase delta subunit (Pde6d) gene
AF024524	Mus musculus LIM domain binding protein 1 (Ldb1) gene

Appendix B: The THMD32 Dataset

AF068624	Homo sapiens 5-aminolevulinate synthase 2 (ALAS2) gene
AF053069	Homo sapiens NADH:ubiquinone dehydrogenase 51 kDa subunit (NDUFV1)
AF032437	Homo sapiens mitogen activated protein kinase activated protein
AF007876	Homo sapiens Na,K-ATPase beta 2 subunit gene
AF051160	Homo sapiens tyrosine phosphatase (PRL-1) gene
AF022382	Homo sapiens UDP-galactose 4' epimerase (GALE) gene
AF045999	Homo sapiens rod cGMP phosphodiesterase delta subunit (PDEd) gene
U53447	Homo sapiens PAPS synthase gene
AF009356	Homo sapiens regulator of G-protein signaling-16 (RGS16) gene
AF019409	Homo sapiens uncoupling protein 2 (UCP2) gene
AF015224	Homo sapiens mammaglobin gene
AF042782	Homo sapiens galanin receptor type 2 (GALR2) gene
AF037207	Homo sapiens persyn gene
AB016243	Homo sapiens gene for regulatory factor 2 of sodium/hydrogen
AF049259	Homo sapiens keratin 13 gene
AB012668	Homo sapiens hFuc-T VII gene for selectin-ligand synthase
U96626	Mus musculus chondroadherin gene
AF042784	Mus musculus galanin receptor type 2 (GalR2) gene
AF042783	Mus musculus galanin receptor type 3 (GalR3) gene
AF078705	Mus musculus vascular adhesion protein-1 gene
AF034569	Mus musculus anticoagulant protein C gene
AF064081	Mus musculus alpha-sarcoglycan gene
AB012159	Mus musculus gene for uncoupling protein-2
AF068199	Mus musculus D-dopachrome tautomerase gene
AF079877	Mus musculus cyclin G2 (Ccng2) gene
AF030522	Mus musculus stannin gene
AF079528	Mus musculus IER5 (Ier5) gene
AF003255	Mus musculus alpha-N-acetylglucosaminidase gene

Appendix B: The THMD32 Dataset

AF033620	Mus musculus platelet endothelial tetraspan antigen-3 (Peta3) gene
AB015652	Mus musculus gene for DJ-1
U82792	Mus musculus allograft inflammatory factor-1 gene
AF009614	Mus musculus homeobox containing nuclear transcriptional factor
AB007139	Mus musculus Psme3 gene for PA28 gamma subunit

List of Author's Publications

1. Yu, R. J. and Tan, E. C., 15-18 December 2003, "Comparison of Different Time-Frequency Distributions in Pitch Detection", *Fourth International Conference on Information, Communications & Signal Processing, Fourth IEEE Pacific-Rim Conference On Multimedia (ICICS-PCM 2003)*, Singapore.
2. Yu, R. J. and Tan, E. C., 8-11 Jan 2004, "Cross-Term Control of Time Frequency Analysis in Electrophysiology Signal Analysis", *Second International Conference on Information Technology and Applications (ICITA2004)*, Harbin, China.
3. Yu, R. J. and Tan, E. C., 25-30 July, 2004, "Application of Time-Frequency analysis in exon classification", *The Fourth International Conference on Bioinformatics of Genome Regulation and Structure (BGRG2004)*, Novosibirsk, Russia.
4. Yu, R. J. and Tan, E. C., 8-10 September 2004, "Application of Time-Frequency analysis in exon classification", *Tenth International Symposium on Integrated Circuits Devices & Systems, Integrated Systems on Silicon (ISIC2004)*, Singapore.
5. Yu, R. J. and Tan, E. C., 11-12 August 2005, "Time-Frequency Analysis of Genome Small Patterns", *Seventh Annual NTU-SGH Symposium*, Singapore.
6. Yu, R. J. and Tan, E. C., 6-9 December 2005, "Spectrogram Analysis of Genome Small Patterns Using Pseudo Smoothed Wigner-Ville Distribution", *Fifth International Conference on Information, Communications & Signal Processing (ICICS2005)*, Bangkok, Thailand.
7. Yu, R. J. and Tan, E. C., Jan 2008 (in press), "Fast Algorithm for Time-Recursive Reassigned Wigner-Ville Analysis", *Signal Processing* (accepted).
8. Yu, R. J. and Tan, E. C., (in press) "Time-Frequency Analysis: A Good Tool for Identification of Critical Parts of Proteins", *Bioinformatics* (accepted).
9. Yu, R. J. and Tan, E. C., "Computational Gene-Finding Using Time-Frequency Analysis", *Journal of Computational Biology* (Submitted, second revision).

References

- Andria, G. and Savino, M., 1994, "Application of Wigner-Ville distribution to measurements on transient signals", *IEEE Transactions on Instrumentation and Measurement*, vol. 43, No. 2, pp. 187-193.
- Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M. and Roberts, K., 1998, *Essential Cell Biology*, New York: Garland Publishing.
- Allen, J. B. and Rabiner, L. R., 1977, "A unified approach to short-time Fourier analysis and synthesis", *Proc. of IEEE*, vol. 65, pp. 1558-1564.
- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J, Zhang, Z., Miller, W., and Lipman, D. J., 1997, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", *Nucleic Acids Res.*, vol. 25, No. 17, pp. 3389-3402.
- Amin, M.G., 1995, "The running time-frequency distributions", *Circuits Systems Signal Process*, vol. 14, No. 3, pp. 401-414.
- Amin, M.G., 1996, "Recursive kernels for time-frequency signal representations", *IEEE Signal Process. Lett.*, vol. 3, No. 1, pp. 16-18.
- Anastassiou, D., 2002, "Frequency-domain analysis of Biomolecular sequences", *BioInformatics*, vol. 16, pp. 1073-1081.
- Anthony, G. J. F., Miller J. H., Suzuki, D. T., Lewontin, R. C. and Gelbart, W. M., 1999, *Introduction to Genetic Analysis*, New York: W. H. Freeman & Co.
- Ashburner, M., Mishra, S., Roote, J., Lewis, S. E., Blazej, R., Davis, T., Doyle, C., Galle, R., George, R., Harris, N., *et al.* 1999, "An exploration of the sequence of a 2.9 Mb region of the genome of drosophila melanogaster: the Adh region", *Genetics*, vol. 153, pp. 179-219.
- Ashburner, M., 2000, "A biologist's view of the Drosophila genome annotation assessment", *Genome Res.*, vol. 10, pp. 391-393.
- Auger, F. and Flandrin, P., 1995, "Improving the readability of time-frequency and time-scale representations by the reassignment method", *IEEE Transaction on Signal Processing*, vol. 43, No. 5, pp. 1068-1089.

-
-
- Baraniuk, R. G. and Jones, D. L., 1996, "Wigner based formulation of the chirplet transform", *IEEE Trans. on Signal Processing*, vol. 44, pp. 3129-3535.
- Bernasconi, V., Bollea, L., Breda, A., Daponte, P., Maroncelli, G. and Rapuano, S., 2004, "A TFR-based method for the quality assessment of UMTS signals: an application on the first Italian experimental network", *IEEE Trans. on Instrumentation and measurement*, vol. 53, No. 2, pp. 485-492.
- Bernardi, G., 1993, "The isochore organization of the human genome and its evolutionary history - a review", *Gene*, vol. 135, pp. 57-66.
- Bertrand, P., Izrar, B., Nguyen, V. T. and Feix, M. R., 1983, "Obtaining non-negative quantum distribution function", *Phys. Lett.*, vol. 94, pp. 415-417.
- Bikdash, U. M. and Yu, K. B., 1993, "Analysis and filtering using optimally smoothed Wigner distribution", *IEEE Trans. on Signal Processing*, vol. 41, pp. 1603-1617.
- Bitmead, R. R., 1982, "On Recursive Discrete Fourier Transform," *IEEE Trans. on ASSP*, vol. 30, no. 2, pp. 319-322.
- Bultan, A., 1999, "A four-parameter atomic decomposition of chirplets", *IEEE Trans. on Signal Processing*, vol. 41, pp. 731-745.
- Bultan, A. and Akansu, A. N., 1998, "A novel time-frequency exciser in spread spectrum communications for chirp-like interference", *Proc. of ICASP*, pp. 3265-3268, Seattle, USA.
- Burge, C. and Karlin, S., 1997, "Prediction of complete gene structure in human genomic DNA", *J. Mol. Biol.*, vol. 268, pp. 78-94.
- Burset, M. and Guigo, R., 1996., "Evaluation of gene structure prediction programs", *Genomics*, vol. 34, pp. 353-367.
- Carpenter, G. and Cohen, S., 1990, "Epidermal growth factor", *J Biol Chem*, vol. 265, No.14, pp. 7709-7712.
- Chen, W., Kehtarnavaz, N. and T. W. Spencer, 1993, "An Efficient Recursive Algorithm for Time-Varying Fourier Transform," *IEEE Trans. on Signal Processing*, vol. 41, no. 7, pp. 2488-2490.
- Chen, V. C. and Hao, L., 1999, "Joint time-frequency analysis for radar signal and image processing", *IEEE Signal Processing Magazine*, vol. 16, pp. 81-93.

-
-
- Choi, H. I. and Williams, W. J., 1989, "Improved time-frequency representation of multicomponent signals using exponential kernels", *IEEE Trans. on Acoust., Speech and Signal Processing*, vol. 37, pp. 861-871.
- Claasen, T. A. C. M. and Mecklenbrauker, W. F. G., 1980, "The Wigner distribution-a tool for time-frequency signal analysis, Part I: continuous-time signals", *Philips J. Res.*, vol. 35, pp. 217-250.
- Claasen, T. A. C. M. and Mecklenbrauker, W. F. G., 1980, "The Wigner distribution-a tool for time-frequency signal analysis, Part II: discrete time signals", *Philips J. Res.*, vol. 35, pp. 276-300.
- Claasen, T. A. C. M. and Mecklenbrauker, W. F. G., 1980, "The Wigner distribution-a tool for time-frequency signal analysis, Part III: relations with other time-frequency signal transformations", *Philips J. Res.*, vol. 35, pp. 372-389.
- Clapp C. and Weiner, R.I., 1992, "A specific, high affinity, saturable binding site for the 16-kilodalton fragment of prolactin on capillary endothelial cells", *Endocrinology*, vol. 130, pp. 1380-1386
- Claverie, J. M., 1997, "Computational methods for the identification of genes in vertebrate", *Human Molecular Genetics*, vol. 6, No. 10, pp. 1735-1744.
- Cohen, L., 1989, "Time-frequency distribution—a review", *Proceeding of IEEE*, vol. 77, pp. 941-981.
- Cohen, L., 1995, *Time-Frequency Analysis*, Prentice Hall.
- Collins, F. S., Patrinos, A., Jordan, E., Chakravarti, A., Gesteland, R., Walters, L., and the members of the DOE and NIH planning groups, 1998, "New Goals for the U.S. Human Genome Project: 1998-2003", *Science*, vol. 282, No. 5389, pp. 682-689.
- Collins, F. S., Morgan, M. and Patrinos, A., 2003, "The human genome Project: lessons from large-scale biology", *Science*, vol. 300, no. 5617, pp. 286 – 290.
- Cooper, P.R., Smilnich, N. J., Day, C. D., Nowak, N. J., Reid, L. H., Pearsall, R. S., Reece, M., Prawitt, D., Landers, J., Housman, D. E., Winterpacht, A., Zabel, B. U., Pelletier, J., Weissman, B. E., Shows, T. B., and Higgins, M. J., 1998, "Divergently transcribed overlapping genes expressed in liver and kidney and located in the 11p15.5 imprinted domain", *Genomics*, vol. 49, pp. 38-51.
- Cosic, I., 1997, *The Resonant Recognition Model of Macromolecular Bioactivity*, Birkhouser.

-
- Cosic, I., Hodder, A. N., Aguilar, M. I., and Hearn, M. T., 1991, "Resonant Recognition Model and Protein Topography," *European Journal of Biochemistry*, vol. 198, pp. 113-119.
- Cosic, I., de Trad, C., Fang, Q. and Akay, M., 2000, "Protein sequencing the RRM model and WT methods, a comparative study", *Protein Engineering*, Vol. 15, No. 3, pp. 193-203.
- Cunningham, G. S. and Williams, W. J., 1992, "Fast computation of the Wigner Distribution for finite-length signals", *ICASSP-92*, pp. 23-26, San Francisco, California.
- de Trad C., Fang Q. and Cosic I., 2000, "The Resonant Recognition Model (RRM) Predicts Amino Acids in Highly Conserved Regions of the Hormone Prolactin (PRL)," *Biophysical Chemistry*, vol. 84, pp. 149-157.
- de Trad, C., Fang, Q. and Cosic, I., 2001, "An overview of protein sequence comparisons using wavelets," *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference*, Victoria. Australia.
- Debruijn, N. G., 1967, "Uncertainty principles in Fourier analysis", *In Inequalities*, O. Shisha, Ed. York, pp. 57-71, NY: Academic Press.
- Dominski, Z. and Kole, R., 1991, "Selection of splice sites in pre-mRNA with short internal exons", *Mol. Cell Biol.*, vol. 11, No.12, pp. 6075-6083.
- Dunham, I., Shimizu, N., Roe, B. A., Chisoe, S., Hunt, A. R., Collins, J. E., Bruskewich, R., Beare, D. M., Clamp, M., Smink, L. J., Ainscough, R., Almeida, J. P., Babbage, A., Bagguley, C., Bailey, J., Barlow, K., Bates, K. N., Beasley, O., Bird, C. P., Blakey, S., Bridgeman, A. M., Buck, D., Burgess, J., Burrill, W. D., O'Brien, K. P., *et al.*, 1999, "The DNA sequence of human chromosome 22", *Nature*, vol. 402, pp. 489-495.
- Duret, L., Mouchiroud, D. and Gautier, C., 1995, "Statistical analysis of vertebrate sequences reveals that long genes are scarce in GC-rich isochores", *J. Mol. E.*, vol. 40, pp. 308-317.
- Essien, K., Akay, M. and Sekine, M., 2002, "Time Frequency Analysis Based on Hydrophobicity," *IEEE EMBS Asia/Pacific Conference 2002*.
- Fickett, W. F. and Tung, C. S., 1992, "Assessment of protein coding measures", *Nucleic Acids Res.*, vol. 20, No.24, pp. 6441-6450.

-
- Fickett, J. W., 1996, "The gene identification problem: an overview for developers", *Computer Chem.*, vol. 20, No.1, pp. 103-118.
- Flandrin, P. and Martin, W., 1985, "Wigner-Ville spectral analysis of nonstationary process", *IEEE Trans. on Acoust., Speech and Signal Processing*, vol. 33, pp. 1461-1470.
- Florea, L., Hartzell, G., Zhang, Z., Rubin, G. M., and Miller, W., 1998, "A computer program for aligning a cDNA sequence with a genomic DNA sequence", *Genome Res.* vol. 8, No.9, pp. 967-974.
- Forsberg, F., and Oung, H., 1995, "Comparison of time-frequency distributions for Doppler spectral estimation", *IEEE Ultrasonics Symposium Proceedings*, vol. 2, pp. 1519-1522.
- Gabor, D., 1946, "Theory of communication", *Journal of IEE*, vol. 93, pp. 429-457.
- George, A., Ordway and Daniel J. Garry, 2004, "Myoglobin: an essential hemoprotein in striated muscle", *Journal of Experimental Biology*, vol. 207, pp. 3441-3446.
- Gilbert, W., 1987, "The exon theory of genes", *Cold Spring Harbor Symposia on Quantitative Biology*, vol. LII: *Evolution of Catalytic Function*, pp. 907-913.
- Giridhar, J., 1998, *Time-Frequency Distributions and Their Applications in Radar Signal Processing. M. S. Thesis.*, Indian Institute of Technology.
- Goertzel, G., 1958, "An algorithm for the evaluation of finite trigonometric series". *Am. Math. Monthly*, vol 65, pp. 34-35.
- Groenen, L. C., Nice, E. C. and Burgess, A. W., 1994, "structure-function relationships for the EGF /TGF-alpha family of mitogens" *Growth Factors*, vol. 11, pp. 235-257.
- Guigo, R. and Fickett, J. W., 1995, "Distinctive sequence features in protein coding, genic non-coding, and intergenic human DNA", *J. Mol. Biol.*, vol. 253, pp. 51-60.
- Guo, Z. Y., Durand, L. G, and Lee, H. C, 1994, "Comparison of time-frequency distribution techniques for analysis of simulated Doppler ultrasound signals of the femoral artery", *Biomedical Engineering, IEEE Transactions on*, vol. 41, Issue 4, pp.332-342.
- Hardison, R. C., 1996, "A brief history of hemoglobins: plant, animal, protist, and bacteria", *Proc Natl Acad Sci USA*, pp. 5675-5679.

-
-
- Hlawatsch, F. and Boudreaux-Bartles, G. F., 1992, "Linear and quadratic time-frequency signal representations", *IEEE Signal Processing Magazine*, Apr., pp. 21-67.
- Jeong, J. and Williams, W. J., 1992, "Kernel design for reduced interference distributions", *IEEE Trans. on Signal Processing*, vol. 40, pp. 402-412.
- John, G. P. and Dimitris, G. M., 1996, *Digital Signal Processing*, Prentice Hall.
- Jones, D. L. and Baraniuk, R. G., 1993, "A signal-dependent time-frequency representation: optimal kernel design", *IEEE Trans. on Signal Processing*, vol. 41, pp. 1589-1602.
- Jones, D. L. and Baraniuk, R. G., 1994, "A signal-dependent time-frequency representation: fast algorithm for optimal kernel design", *IEEE Trans. on Signal Processing*, vol. 42, pp. 134-146.
- Jones, D. L. and Parks, T. W., 1992, "A resolution comparison of several time-frequency representations", *IEEE Trans. on Signal Processing*, vol. 40, pp. 413-420.
- Katoh, K., Misawa, K., Kuma, K., and Miyata, T., 2002, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform", *Nucleic Acids Res.*; vol. 30, No. 14, pp. 3059-3066.
- Kawashima, S. and Kanehisa M., 2000, "Aaindex: Amino acid index database", *Nucleic Acids research*, vol. 28, No. 1, pp. 374-375.
- Kodera, K., Gendrin, R. and de Villedary, C., 1978, "Analysis of time-varying signals with small BT values", *IEEE Trans. ASSP*, vol. 26, issue. 1, pp. 64-76.
- Komoroski, E. M., Komoroski R. A., Valentine, J. L., Pearce, J. M. and Kearns, G. L., 2000, "The use of nuclear magnetic resonance spectroscopy in the detection of drug intoxication", *Journal of Analytical Toxicology*, vol. 24, No. 3, pp.180-187.
- Krogh, A., 1997, "Two methods for improving performance of an HMM and their application for gene-finding", in *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pp. 179-186.
- Kulp, D., Haussler, D., Reese, M.G., and Eeckman, F.H., 1996, "A generalized hidden Markov model for the recognition of human genes in DNA", In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pp. 134-142, AAAI Press, Menlo Park, CA.
- Kyte, J. and Doolittle, R. F., 1982, "A simple method for displaying the hydropathic character of a protein", *Journal of Molecular Biology*, vol. 157, pp. 105-132.

Reference

-
-
- Lehmann, H. and Huntsman, R.G., 1966, *Man's Haemoglobins Including the Haemoglobinopathies and Their Investigation*, Philadelphia Lippincott.
- Levin, M. J., 1964, "Instantaneous spectra and ambiguity functions", *IEEE Trans. Inform. Theory*, vol. 10, pp. 95-97.
- Liu, K. J. R. and Chiu, C. T., 1993, "Unified Parallel Lattice Structures for Time-Recursive Discrete Cosine/Sine/Hartley Transforms," *IEEE Trans. on Signal Processing*, vol. 41, no. 3, pp. 1357-1377.
- Liu, K. J. R., 1993, "Novel parallel architectures for Short-Time Fourier Transform", *IEEE Trans. on Circuits and Systems*, vol. 40, pp. 786-789.
- Liu, K. J. R., Chui, C. T., Kolagata, R. K. and Jaja, J. F., 1994, "Optimal unified architectures for real-time computation of time-recursive discrete sinusoidal transforms", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, pp. 168-180.
- Lu, H. S., Chai, J. J., Li, M., Hiang, B. R., He, C. H. and Bi, R. C., 2001, "Crystal structure of human epidermal growth factor and its dimerization.", *Journal of Biochemistry*, vol. 276, issue. 37, pp. 34913-34917.
- Lukashin, A.V. and Borodovski, M., 1998, "GeneMark.hmm: new solutions for gene-finding", *Nucleic Acids Res.*, vol. 26, pp. 1107-1115.
- Mallat, S. G. and Zhang, Z., 1993, "Matching pursuits with time-frequency dictionaries", *IEEE Trans. on Signal Processing*, vol. 41, pp. 3397-3415.
- Mann, S. and Haykin, S., 1995, "The chirplet transform: physical considerations", *IEEE Trans. on Signal Processing*, vol. 44, pp. 2745-2761.
- Mironov, A. A., Ficket, J. W., and Gelfand, M. S., 1999, "Frequent alternative splicing of human genome", *Genome Res*, vol. 9, pp. 1288-1293.
- Nawab, S. H. and Quatieri, T. F., 1988, "Short-time Fourier Transform, In Lim, J. S. and Opeenheim, A. V. (eds.)", *Advanced Topics in Signal Processing*, Printice-Hall, Englewood cliffs, New Jersey.
- Norman, A. and Litwak, G., 1997, *Hormones*, San Diego: Academic Press.
- O'Neill, J. C., 1997, *Shift Covariant Time-Frequency Distributions of Discrete Signals*, Ph. D. Thesis. University of Michigan.
- Page, C. H., 1952, "Instantaneous power spectra", *Jour. Appl. Phys*, vol.23, pp. 103-106.

-
-
- Papandreou, A., Hlawatsch, F. and Boudreaux-Bartles, G. F., 1993, "The hyperbolic class of QTFRs- Part I: constant-Q warping, hyperbolic paradigm, properties, and members", *IEEE Trans. on Signal Processing*, vol. 41, pp. 3425-3444.
- Peng, C. K., Buldyrev, S. V., Goldberger, A. L., Havlin, S., Sciortino, F., Simons, M., Stanley, H. E., 1992, "Long-range Correlations in Nucleotide Sequences," *Nature*, vol. 356, pp. 168-170.
- Pirogova E. and Cosic I, 2001, "Examination of Amino Acid Indexes Within the Resonant Recognition Model," *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference: Biomedical Research in 2001*.
- Qian, S., and Chen, D., 1996, *Joint Time-Frequency Analysis: Method and Application*, Prentice Hall.
- Qian, S. E. and Morris, J. M., 1990, "Fast algorithm for real joint time-frequency transformations of time-varying signals", *Electronics Letters*, vol. 26, pp. 537-539.
- Rao, R. M. and Bopardikar, A. S., 1998, *Wavelet Transforms, Introduction to Theory and Applications*, Addison Wesley Longman, Inc.
- Raven, P., H. and Johnson, G., B., 2001, *Biology*, McGraw-Hill Science/Engineering/Math
- Reece J. and Campbell N., 2005, *Biology, 7th Edition*, Benjamin Cummings.
- Rihaczek, A., 1968, "Signal energy distribution in time and frequencies", *IEEE Trans. on Information Theory*, vol. 42, pp. 3241-3244.
- Ross, M. J., Shaffer, H. L., Cohen, A., Freudberg, R. and Manley, H. J., 1974, "Average magnitude difference function pitch extractor", *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. ASSP-22, pp. 353-362.
- Schulz, R.A. and Butler, B.A., 1989, "Overlapping genes of *Drosophila melanogaster*: organization of the z600-gonadal-Eip28/29 gene cluster", *Genes Dev.*, vol. 3, No.2, pp. 232 –242.
- Speroff, L., Glass, R. H. and Kase, N., G., 1999, *Clinical Gynecologic Endocrinology and Infertility. Sixth edition*. Lippincott Williams & Wilkins, Baltimore, MD.
- Stankovic, L., 1994, "A multitime definition of the Wigner higher order distribution: L-Wigner distribution", *IEEE Letters in Signal Processing*, vol. 1, pp.106-109.

Reference

-
- Strynadka, N. and James, M., 1996, "Lysozyme: a model enzyme in protein crystallography" in *Lysozymes- Model Enzymes in Biochemistry and Biology*, Birkhauser.
- Huisman, T. H. J., Carver, M. F. H. and Efremov, G. D., 1996, "A Syllabus of Human Hemoglobin Variants", *The Sickle Cell Anemia Foundation*, Augusta, GA, USA.
- Tzagaloff, A., 1982, *Mitochondria*, New York: Plenum Press.
- Unser, M., 1983, "Recursion in Short-time Signal Analysis," *Signal Processing*, vol. 5, no. 3, pp. 229–240.
- Ville, J., 1948, "Theorie et applications de la notion de signal analytique", *Cable et transmissions*, vol. 2A, pp. 61-74. Translated from French by Selin, I., 1958, "Theory and applications of the notion of complex signal", *RAND Corporation Technical al Report T-92*, Santa Monica, CA.
- White, C. M., 1999, "A review of potential cardiovascular uses of intravenous glucagon administration", *J Clin Pharmacol*; vol. 39:442-447
- White, J. V., Stultz, C. M., Smith T. F., 1994, "Protein Classification by Stochastic modeling and Optimal Filtering of Amino Acid Sequences," *Mathematical Biosciences* vol. 119, pp.35-75.
- Wigner, E. P., 1932, "On the quantum correction for thermodynamic equilibrium", *Phy. Rev.*, vol. 40, pp. 749-759.
- Zaveri, H. P., Williams, W. J., Iasemidis, L. D. and Sackellares, J. C., 1992, "Time-frequency representation of electrocorticograms in temporal lobe epilepsy", *IEEE Transactions on Biomedical Engineering*, vol. 39, No. 5, pp. 502-509.
- Zhang, I., 2000, "Protein-length distributions for the three domains of life", *Trends Genet.*, vol. 16, No. 3, pp. 107-109.
- Zhang, Y. Z., Yao, Y. F., 1989, "Fast implementation of recursive DFTs", *IEEE Proc. ICASSP*, pp. 1071–1074.
- Zoubak, S., Clay, O., and Bernardi, G, 1996, "The gene distribution of the human genome", *Gene*, vol. 174, pp. 95-102.