2006

Alignment shift analysis using hybridization energy rules

Jaya Nair
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Nair, Jaya, "Alignment shift analysis using hybridization energy rules" (2006). *Master's Theses*. 2913.
DOI: <https://doi.org/10.31979/etd.u4c2-wmmt>
https://scholarworks.sjsu.edu/etd_theses/2913

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

ALIGNMENT SHIFT ANALYSIS USING HYBRIDIZATION ENERGY RULES

A Thesis

Presented to

The Faculty of the Department of Chemical Engineering

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Jaya Nair

May 2006

UMI Number: 1436939

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 1436939

Copyright 2006 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© 2006

Jaya Nair

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF CHEMICAL ENGINEERING

12/13/05

Dr. Brooke Lustig

12/13/2005

Dr. Melanie McNeil

12/13/05

Dr. Elaine Collins

APPROVED FOR THE UNIVERSITY

04/26/06

ABSTRACT

ALIGNMENT SHIFT ANALYSIS USING HYBRIDIZATION ENERGY RULES

By Jaya Nair

This study seeks to develop a new tool for sequence alignment search using a possible relationship between hybridization energy values, BLASTN scores, and the BLAST alignment lengths. Additionally, this research explores the validity of alternate alignments by introducing shifts in the original alignments. Directly translated DNA sequences from 96 proteins and 11 non-coding RNA are used to perform a BLASTN search in GenBank. Six shifts, three in the forward direction and three in the backward direction, are introduced to the resulting 29,954 alignments and hybridization energy values are computed for each of the shifted and original alignments. Results indicate a statistically significant linear correlation ($R = \pm 0.967$) between the BLAST score and the corresponding hybridization energy values for the original alignments. However, the shifted alignments did not exhibit any important correlation for the same parameters.

ACKNOWLEDGEMENT

I would like to thank all my reading committee members for their suggestions and guidance throughout this research. I am deeply appreciative of Dr. Brooke Lustig for making this study possible. In addition, I wish to thank Dr. Melanie McNeil and Dr. Elaine Collins for their invaluable advice and guidance that made this thesis possible. Lastly, I am thankful to my husband Nirmal, my parents, and my daughters Ammu and Nikki for their steadfast support and unwavering encouragement throughout this period.

TABLE OF CONTENTS

CHAPTER ONE	INTRODUCTION	1
CHAPTER TWO	BACKGROUND	5
CHAPTER THREE	LITERATURE REVIEW	14
3.1	ALIGNMENT SEARCH ALGORITHMS	14
3.1.1	DYNAMIC PROGRAMMING ALGORITHMS	15
3.1.2	HEURISTIC ALGORITHMS	16
3.2	SCORING MATRICES	16
3.3	SEQUENCE SEARCH ALGORITHMS USING RNA ENERGY RULES	18
3.4	FRAMESHIFT AND ALIGNMENT SHIFT ALGORITHMS	19
CHAPTER FOUR	HYPOTHESIS AND OBJECTIVES	25
4.1	HYPOTHESIS	25
4.2	OBJECTIVES	25
CHAPTER FIVE	MATERIALS AND METHODS	26
5.1	MATERIALS	26
5.2	METHOD	26
5.3	PROPOSED ANALYSIS	29
5.3.1	HYBRIDIZATION ENERGY VERSUS BLAST BIT SCORE	29
5.3.2	HYBRIDIZATION ENERGY VERSUS BLAST ALIGNMENT LENGTH	30
CHAPTER SIX	RESULTS AND DISCUSSION	32
6.1	FREQUENCY DISTRIBUTION OF NUMBER OF BASE PAIRS IN THE QUERY SEQUENCE	35

6.2	FREQUENCY DISTRIBUTION OF NUMBER OF ALIGNMENTS PER QUERY SEQUENCE	39
6.3	FREQUENCY DISTRIBUTION OF BIT SCORES	42
6.4	CORRELATION PLOTS	44
6.4.1	CORRELATION PLOTS FOR HYBRIDIZATION ENERGY VERSUS BLASTN SCORE	44
6.4.2	CORRELATION PLOTS FOR HYBRIDIZATION ENERGY VERSUS ALIGNMENT LENGTH	50
6.5	DISCUSSION	57
	CHAPTER SEVEN CONCLUSIONS	62
	REFERENCES	64
	APPENDICES	
A.	Table showing the ncRNA sequences	69
B.	Table showing the 96 proteins and 11 ncRNA with their corresponding number of alignments, number of query base pairs, slope, and R-values	73
C.	Hybridization Energy (HE) versus BLASTN Score	76
D.	Hybridization Energy (HE) versus Alignment Length	112
E.	Hybridization Energy (HE) versus BLASTN score with $E > 0.01$	148
F.	Equations to determine uncertainties	182
G.	Perl script used "ExtractBlastnData.pl"	183
H.	Perl script used "CalculateBlastnData.pl"	191
I.	Curve Fit	200

LIST OF FIGURES

Figure 1. Schematic of the DNA molecule	5
Figure 2. Schematic showing Central Dogma	6
Figure 3. Schematic of frameshift mutation	8
Figure 4. Schematic showing mismatches and gaps in alignments.	10
Figure 5. Schematic of the BLAST algorithm	12
Figure 6. Schematic of the Smith-Waterman algorithm	16
Figure 7. Flow diagram of the method to calculate HE with shifts in alignments	26
Figure 8. Frequency distribution of the number of base pairs in the 96 protein-based query sequences	36
Figure 9. Frequency distribution of the number of base pairs in the 11 ncRNA query sequences	38
Figure 10. Frequency distribution of the number of alignments per query sequence for the 96 protein-based DNA sequences	40
Figure 11. Frequency distribution of the number of alignments per query sequence for the 11 ncRNA sequences	41
Figure 12. Frequency distribution of the bit scores from the BLASTN searches for the 96 directly determined DNA sequences	42
Figure 13. Frequency distribution of the bit scores from the BLASTN searches for the 11 ncRNA sequences	43
Figure 14. Correlation plot of alignments for directly determined Trypsin (1sgt)	45
Figure 15. Aggregate correlation plot of Hybridization Energy versus BLASTN score for the 26,702 alignments of the 96 directly determined DNA sequences	46
Figure 16. Aggregate correlation plot of Hybridization Energy versus BLASTN score for the 3252 alignments of the 11 ncRNA sequences	47

Figure 17. Correlation plot of Hybridization Energy versus BLASTN scores for the subset of protein-based alignments of Trypsin (1sgt)	48
Figure 18. Aggregate correlation plot of 6,142 DNA alignments that form a subset of the 26,702 alignments that have the Expect values, $E > 0.01$	49
Figure 19. Correlation plot of Hybridization Energy versus corresponding Alignment Length for the protein-based alignments of Aminopeptidase (1amp)	51
Figure 20. Aggregate correlation plot of Hybridization Energy (ordinate) versus Alignment Length (abscissa) for BLASTN alignments of 96 directly translated protein sequences	53
Figure 21. Aggregate correlation plot of Hybridization Energy (ordinate) versus Alignment Length (abscissa) for BLASTN alignments of 11 directly translated ncRNA sequences	54
Figure 22. Aggregate correlation plot of the averaged shifted energy values for the 26,702 DNA alignments for the 96 directly translated protein sequences	55
Figure 23. Aggregate correlation plot of the averaged shifted energy values for the 3252 alignments of the 11 directly translated ncRNA sequences	56

LIST OF TABLES

Table 1. Definition of insertion, deletion, and inversion frameshifts	20
Table 2. “Proteins_Accession Number” used in this study	33
Table 3. Proteins not used in the study	34
Table 4. ncRNA used.	35
Table 5. ncRNA not used	35
Table 6. Sample sequences derived from back-translation versus direct method	37

CHAPTER ONE

INTRODUCTION

Biology, by definition, is the study of living things. It includes the interaction of species and the functions of cells and tissues within an individual organism [1].

Biologists and other related scientists focus on gathering and analyzing data. Now, unlike a few decades ago, data in the form of genetic information, DNA, and protein sequences are collected faster than can be analyzed because of sophisticated programming techniques [1]. As a result, there are very large repositories of DNA and protein sequence data in various databases.

Bioinformatics is the science of using the available genetic information to understand biology. Bioinformatics can be considered a branch of the larger field of computational biology, the application of quantitative analytical techniques in modeling biological systems [1]. One of the goals of bioinformatics is to create tools to aid researchers find homologous sequences of known function to the sequence of interest of unknown function. This comparison of biological sequences is the starting point of most of the common bioinformatics methods and is called an alignment. There are two major types of comparisons: pair wise and multiple sequence alignments. Pair wise alignments address a pair of sequences at a time whereas multiple alignments compare more than two sequences at a given time. The interest in search and comparison of the alignments resulted in the development of large databases like Genbank [2]. Early research in

bioinformatics focused on algorithms to accelerate the search for alignments in large databases.

In the early 1970's, dynamic programming algorithms revolutionized the field of sequence alignment searches [2]. Eventually programs like FASTA and BLAST were developed to optimize searches for sequence alignments. FASTA, an abbreviation for Fast-All, was one of the first popular tools developed for sequence alignment in the late 1980's [3]. It is used for a quick DNA or protein comparison. This program uses a substitution matrix to optimize searches for local alignments. However, there was a significant trade-off between sensitivity and speed when using FASTA. In addition, due to overwhelming increase in the number of genomic sequences in the different databases there emerged the dilemma of having increased sensitivity without compromising the speed of search [4]. Due to these various drawbacks of FASTA, users have been switching to Basic Local Alignment Search Tool or BLAST, which was developed in the early 1990's [5].

BLAST, like FASTA, employs a heuristic method to find the optimal alignment between the sequence of interest, query, and the database sequences. It does a word similarity search where if it finds similar letters on the two sequences that are being aligned, then it extends the search to the adjacent letters to find similarity without allowing gaps [6]. BLAST is described in detail in the background section of the thesis.

One of the major issues for BLAST is addressing gaps in alignments. As of 2004, there were 40,604,319 sequences in the Genbank database [7]. Hence, interpreting and

analyzing data needs to be accelerated. This is a reason new methods such as the hybridization energy method, developed by the Lustig group [8], are being actively researched.

The hybridization energy (HE) was initially developed for prediction of secondary and the three-dimensional structure of RNA by Turner et al. [9]. There is a significant correlation between free energies associated with DNA-DNA hybridization and those associated with RNA-RNA hybridization [10]. This similarity in energy properties, along with appropriate representation of mismatched regions in nucleotide sequence alignment by the loop structures in RNA, form the basis of application of RNA hybridization rules to the searches of nucleotide sequence alignments [8].

The approach employs hybridization of the query and the complementary of the subject sequences in an alignment from the BLASTN output. The numerical values of energies for base pairing and loop penalties were obtained from Turner et al. [9]. Here the gaps and mismatches in alignments are represented by bulges and internal loops that are assigned a numerical value. Now along with the Expectation (E) value and bit score from the BLASTN output, there will be an energy value associated with every alignment. Here bit score is the normalized alignment score that can be used to compare scores from different searches and the E value shows the number of alignments with scores equal to or better than the bit score that are expected to occur in a database search by chance [6].

Taking this a little further, this study applies the rules of hybridization to the resulting sequences after doing six shifts to the original sequence alignment. As a first

attempt, this method could also prove to be an alternate way to generate randomized sequences and their energies. Previously, random sequences were generated using shuffling to compute P-value from FASTA [11]. In addition, this alignment-shifting method increases the number of alignments for comparison to the original alignment generated by BLAST. The energy values that are calculated for the different shifts will be compared to that of the original alignment from the BLAST output to assert the correlation between BLAST scores and alignment length to their hybridization energy values

CHAPTER TWO

BACKGROUND

Deoxyribonucleic Acid (DNA), present inside the nucleus of the cell, is a double-stranded molecule twisted into a helix as shown in Figure 1. Each spiraling strand, comprised of a deoxy sugar-phosphate backbone and attached bases, is connected to a complementary strand by non-covalent hydrogen bonding between paired bases. The bases are adenine (A), thymine (T), cytosine (C), and guanine (G). A and T in adjoining strands are connected by two hydrogen bonds; G and C are connected by three hydrogen bonds [12].

Figure 1. Schematic of the DNA molecule [12].

Ribonucleic Acid (RNA) is similar to a single strand of DNA. In RNA, uracil (U) replaces thymine (T) of the DNA. Here the nucleotide molecule contains ribose rings in the backbone. RNA delivers DNA's genetic message to the cytoplasm of a cell where proteins are made. There are RNA molecules that function without being translated into proteins called non-coding RNA (ncRNA) e.g., transfer RNA (tRNA) and ribosomal RNA (rRNA) [12].

The Central Dogma describes the flow of information in a cell. This information pathway can be broken into three steps: replication of DNA, transcription of DNA to produce messenger RNA (mRNA), and translation of mRNA into proteins as shown in Figure 2 [13].

Figure 2. Schematic showing Central Dogma [13].

Codons: The three bases in a DNA or RNA sequence which specify a single amino acid [12]. There are 20 common amino acids, which can be coded from the four bases present in the DNA or RNA. There are 64 (4^3) possible combinations for the

formation of an amino acid. Due to this high redundancy, many amino acids can be coded by more than one triplet [12].

Reading Frames: In translation, codons of three nucleotides determine the amino acid that is added next in the growing protein chain. It is imperative then to decide the codon that will start the translation process and the codon that will terminate it. This is called the open reading frame (ORF). Once a gene has been sequenced, it is important to determine the correct ORF. Every region of DNA has six possible reading frames, three in each direction. The reading frame that is used determines which amino acids will be encoded by a gene. Once the open reading frame is known, the DNA sequence can be translated into its corresponding amino acid sequence. An open reading frame starts with an ATG in most species and ends with a stop codon (TAA, TGA, or TAG) [12].

Frameshift Mutation: Frameshift mutation as shown in Figure 3 is defined as the addition or deletion of one or two bases in DNA sequence which in turn can disrupt the reading frame, or the grouping of the codons, resulting in a completely different translation from the original [12]. The earlier in the gene the deletion or insertion occurs, the more altered is the gene product.

Figure 3. Schematic of frameshift mutation [14].

Sequence Alignment: Sequence alignment is the process of lining up two or more sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) in order to evaluate the degree of similarity and the possibility of homology [6]. There are two different ways of aligning sequences:

- **Global versus Local** - When two or more sequences (nucleic acid or protein) are compared in their entirety, it is called global alignment. It is used to find closely related sequences. Local alignment is the alignment of some parts of nucleic acid or protein sequences. This method finds related regions within sequences. It is a more flexible technique when compared to global alignment in that the related regions that appear in a

different order in the two proteins (known as domain shuffling) can be identified as being related. This is not possible with global alignment methods [6].

- **Pair wise versus Multiple** - The method of aligning a protein/amino acid or DNA/nucleic acid sequences is called pair wise alignment. Typically, the purpose of pair wise alignments is to identify homologues (relatives) of a gene or gene-product by aligning unknown sequences with a database of known sequences. Another important use of this technique is in the studies of molecular evolution. BLAST is an example of pair wise methods. Multiple alignment is an extension of pair wise alignment to incorporate three or more sequences. Several methods for determining multiple alignments exist, one of the most widely used being CLUSTAL-W. Instead of searching a database, multiple alignment methods take a few sequences and find common regions between them all [6].

Gap and Mismatch: A gap is a space introduced into an alignment to compensate for insertions and deletions in one sequence relative to another [6]. Introduction of a gap is penalized (the gap score) from the alignment score. This is called gap-open penalty (GOP). The extension of the gap to include additional nucleotides or amino acid is also penalized in the scoring of an alignment and is called gap-extension penalty (GEP). This results in an affine linear function in the gap length, commonly written as $g(k) = a + b \cdot k$. Here “a” is the cost of opening a gap, “b” is the cost of extending the gap, and “k” is the length of the gap [6]. Now if at any given spot in an alignment, there exists a difference

in base pairs, as shown below in Figure 4, it is called a mismatch. Mismatches are also penalized but not as heavily as gaps.

Figure 4. Schematic showing mismatches and gaps in alignments [6].

Basic Local Alignment Search Tool (BLAST): It uses the method of Altschul et al. [5] to search for similarities between a query sequence and all the sequences in the database. The query sequence and the database that needs to be searched can be protein or nucleic acids or any combination. There are five different programs in the BLAST family [15]:

- BLASTP: Protein query searching a protein database. Each database sequence is compared to the query in a separate protein-protein pair wise comparison.
- BLASTX: Nucleotide query searching a protein database. The query is translated, and each of the six products is compared to each database sequence in a separate protein-protein pair wise comparison.

- BLASTN: Nucleotide query searching a nucleotide database. Each database sequence is compared to the query in a separate nucleotide-nucleotide pair wise comparison.
- TBLASTN: Protein query searching a nucleotide database. Each nucleotide database sequence is translated, and each of the six products is compared to the query in a separate protein-protein pair wise comparison.
- TBLASTX: Nucleotide query searching a nucleotide database. The query and each database sequence are both translated in six frames, and each of the 12 products is compared in 36 different pair wise comparisons.

BLAST is a statistically driven search method that finds optimized regions of similarity between the sequence of interest, called the query sequence, and the sequences in the database. In a matrix, within these regions of similarity called segment pairs, the sum of scores of the subsequent pairs should be higher than some level that would be expected to occur by chance alone. This is called the Expectation Value (E value). This value can be set by the user for a particular search. However, by default E value is set at 10.0 [15].

Once all the words in the query sequence are finished testing, the highest scoring segment pair (HSP) alignments are generated for the database sequences called the subject sequences. The output gives the list of individual sequence alignments that are ranked according to the scores. These scores reflect the probability of procuring a

particular sequence with a similar or higher score from the database randomly [15]. The algorithm employed by BLAST is shown in Figure 5.

Figure 5. Schematic of the BLAST algorithm [6].

Nucleic Acid Hybridization: It is the process of creating a hybrid DNA/RNA strand [16]. The two strands of a DNA molecule are denatured by heating to about 100°C. At this temperature, the double helix strands that are held together by the complementary base pairs are disrupted and the helix rapidly dissociates into two single strands. This denaturing reaction is reversible. If the two single strands of DNA are kept together for a prolonged period at 65°C, they will pair together to form the helix again. This process is called DNA renaturation or hybridization. Similar hybridization reactions can occur between any two single stranded nucleic acid chains: DNA/DNA, RNA/RNA,

and DNA/RNA. If an RNA transcript is introduced during the renaturation process, the RNA competes with the coding DNA strand and forms double-stranded DNA/RNA hybrid molecule. These hybridization reactions can be used to detect and characterize nucleotide sequences using a particular nucleotide sequence as a probe. Techniques such as nucleic acid sequencing, polymerase chain reactions (PCR), and fluorescence *in situ* hybridization (FISH) are based on the binding of a nucleic acid fragment to its complement [16].

Hybridization Energy (HE): The energy required to bind a nucleotide sequence to its complement is called its energy of hybridization. In the scope of this thesis, it is the energy used to bind the query sequence to the complement of the subject sequence obtained from the BLASTN output thus modeling the energies of such sequence annealing [8].

CHAPTER THREE

LITERATURE REVIEW

Sequence alignment is the fundamental method to detect biologically significant information in sequences. Typically, the purpose of alignment is to find homologues (relatives) of a gene or gene-product in a database of known examples. Some of the common applications of alignment methods are to identify sequences of unknown structure/function and to study molecular evolution. There are no single best alignments. Alignments can be optimized by adjusting the parameters they employ to give the highest possible score. These optimized alignments give the most significant similarities with the least differences [1].

In an alignment, gaps are introduced to compensate for insertions and deletions in one sequence relative to another to achieve better correspondence between them. Biologically, this could correspond to a mutation event that eliminates a part of a gene (deletion), or introduces new DNA into a gene (insertion). This might cause shifts in the reading frames, which in turn results in mistranslations of DNA into proteins or introduce a stop codon in the gene [12].

3.1 Alignment Search Algorithms

The most widely used searching algorithms to find alignments can be broadly classified into Dynamic Programming algorithm and Heuristic algorithm [17].

3.1.1 Dynamic Programming Algorithms

Needleman and Wunsch [18] implemented the dynamic programming method for the first time to find the best global alignment. This method used a dot matrix and did not impose any restrictions on the penalty assigned to a gap of a certain length. Since then variations have been developed. One of them is the quasi-global alignment algorithm developed by Sellers [19] that aligns the entire length of a shorter sequence within the best matching region of the longer sequence. The Smith-Waterman algorithm, a recursive mathematical model searches for regions of local sequence similarity, referred to as high scoring pairs (HSP), and builds the sequences on those pairs. The Smith-Waterman algorithm can be expressed as [17]:

$$SW_{i,j} = \max\{SW_{i-1,j-1} + s(a_i, b_j); SW_{i-k,j} + gap_j; SW_{i,j-k} + gap_i; 0\} \quad \text{Equation 1}$$

Where, for an affine gap penalty:

$$gap_j = open_gap + (i-k+1)*extend_gap; gap_i = open_gap + (j-k+1)*extend_gap$$

Here $SW_{i,j}$ is the Smith Waterman score corresponding to the position (i,j) in the matrix aligning sequence A(A1A2A3.....Ai) with sequence B(B1B2B3.....Bj). 'k' is the gap length of the residues, as shown in Figure 6. This is essentially the same as the global alignment algorithm except that a zero is added to the recurrence equation, which does not allow the score to be a negative value [17].

Figure 6. Schematic of the Smith-Waterman algorithm [17].

The application of dynamic programming algorithms are computationally very intensive as the above mentioned Equation 1 needs to be applied to each of the cells in order to get the alignment score. This demands memory proportional to the length of the longest sequence being aligned which is beyond the scope of a personal computer [17].

3.1.2 Heuristic Algorithms

This includes the various flavors of FASTA and BLAST. As noted earlier in the Introduction and Background section of this paper, BLAST has been dominant in most applications of sequence search [3, 6].

3.2 Scoring Matrices

Scoring is a vital form of quantification of alignments when using search algorithms. The scores reflect whether the substitution of one residue for another still

conserves homology in structure and function. For pair wise alignments, score is the sum of substitution scores and gap penalties over all aligned areas, so that the best alignment corresponds to the alignment with the highest score. Initially, alignment programs for nucleic acid comparison used the 4x4 unitary matrix in which ones were assigned for all A-A, C-C, G-G, and T-T matches and zeros were assigned to all mismatches [20].

The detailed study by the Dayhoff group [21] gives the frequencies of amino acid substitutions for each other through the evolution process. In their study, they did a global alignment of all the proteins in several families and then constructed phylogenetic trees for each family. Here, phylogenetic tree is a tree constructed to show evolutionary relationships between species or entities which are presumed to have a common ancestor. This methodology catered to both the generation and selection of mutations. However, due to limited availability of resources, not all mutations were observed. The Percent Accepted Mutation (PAM) / Dayhoff family of matrices used counts derived from a tree-like branching evolutionary model where each matrix is developed for a specific PAM distance. However, none of the matrices covered the entire range of percent identity above 94% efficiency [17].

Henikoff and Henikoff [22] developed the BLOSUM set of matrices, which use counts derived from highly conserved blocks within multiple sequence alignment without gaps. The BLOCKS comprise of regions that would be most likely found in a database search. The counts are then summed up across all BLOCKS to calculate the relative

frequency of substituting one residue for another. BLOSUM 62 is the default matrix for many search programs [17,22].

The GONNET set of matrices [23], created like the PAM matrices, are better for computing evolutionary distance. The GONNET matrices are used by default in the ClustalW multiple alignment program (since version 1.8). Vogt et al., postulated that the GONNET set of matrices were best after optimization for gap penalties [24].

3.3 Sequence Search Algorithms using RNA Energy Rules

The Lustig group has developed a computational sequence using nucleic acid hybridization [8, 25, 26]. They selected representative set of protein and ncRNA sequences, carefully back-translated them to their DNA sequences and then aligned these sequences as queries using BLASTN. An aligned sequence strand, typically the one corresponding to the subject, was converted into its complementary form and annealed to the query as indicated by the BLASTN alignment. The resulting hybridization energy value was calculated for each alignment pair using RNA secondary structure energy values as estimates for DNA base-pairing energies and bulge penalties. These energy values showed a statistically significant linear correlation when plotted against BLAST scores suggesting the validity of the model developed by Lustig [8].

The RNA structure energy values used to calculate the hybridization values are obtained from studies done by Turner et al. [9] of experimentally determined calorimetric melting curves of oligonucleotides. The study contains data for match, mismatch, internal loop, and bulge loops. SantaLucia [27] consolidated data from various

researches and did a comparison study while taking into account the different conditions on the data. Allawi and SantaLucia [28, 29] and Peyret et al. [30] independently determined single base internal mismatch DNA hybridization data. Single base bulge loops effects were determined by Zhu and Wartell [31].

3.4 Frameshift and Alignment Shift Algorithms

Both the frequency and the type of errors (substitution, insertion, deletion of one or more bases) are important while considering the impact of errors on DNA sequences [32]. Each of these error types have different effects on the amino acids coded from the different genes. Substitution of one base for another reduces finding homologies in an alignment. Insertions and deletions can cause a shift in the open reading frame (ORF) or introduction of stop codons in putative genes, which has great impact at the translation level-either mistranslation or its premature termination [33]. This can again affect finding homologies when comparing proteins to sequences in the database. Estimates of frameshift errors are about 2.5% in prokaryotic open reading frames (ORF) [34].

Analogous to frameshift, is a novel concept called alignment shift in which one of the sequences in an alignment undergoes one base-pair shift at a time after which the new alignment is scored. This method can provide a better understanding to introducing gaps in sequences and give an insight to scoring with intentional errors introduced into an alignment. However, no literature is available on this approach. Nevertheless, researchers have developed several methods to address error detection due to frameshifts.

States and Botstein [35] performed a study using a two-stage Bayesian algorithm to see whether the errors in DNA sequences will affect the ability to align the predicted amino acid sequences with the related protein sequences. They found that in the case of amino acid alignments, this model can address sequences with up to 5% substitution errors and frameshift errors up to 1% provided there exists at least 33% amino acid identity in the protein sequences. The authors also noted that the Bayesian algorithms could not deal with introns.

Claverie [36] developed five series of new scoring matrices each, which can detect a particular kind of frameshift i.e., insertion and deletion of a single base and nucleotide sequence inversion (coding/non-coding strand exchange) in the three reading frames as shown in Table 1.

Table 1. Definition of insertion, deletion, and inversion frameshifts [36].

Non-coding strand:		$a_1a_2a_3$	$b_1b_2b_3$	$c_1c_2c_3$	$d_1d_2d_3$	$e_1e_2e_3$	$f_1f_2f_3$	
Coding strand:	5'	$A_1A_2A_3$	$B_1B_2B_3$	$C_1C_2C_3$	$D_1D_2D_3$	$E_1E_2E_3$	$F_1F_2F_3$	3'
Insertion (rf: 1->3)		A_1-A_3	$A_3B_1B_2$	$B_3C_1C_2$	$C_3D_1D_2$	$D_3E_1E_2$	$E_3F_1F_2$	
Deletion (rf: 1->2)		$A_1A_3B_1$	$B_2B_3C_1$	$C_2C_3D_1$	$D_2D_3E_1$	$E_2E_3F_1$	$F_2F_3G_1$	
In phase inversion:		$f_3f_2f_1$	$e_3e_2e_1$	$d_3d_2d_1$	$c_3c_2c_1$	$b_3b_2b_1$	$a_3a_2a_1$	
rf:-2 inversion		$f_2f_1e_3$	$e_2e_1d_3$	$d_2d_1c_3$	$c_2c_1b_3$	$b_2b_1a_3$	a_2a_1	
rf:-3 inversion		f_3f_2	$f_1e_3e_2$	$e_1d_3d_2$	$d_1c_3c_2$	$c_1b_3b_2$	$b_1a_3a_2$	

After validation, these matrices along with BLASTP were used to perform an exhaustive search for deletion, insertion and inversion frameshifts of 28,154 protein

sequence entries in Swissprot database (release 24.0, 9547,407 residues). The scoring matrices detected hundreds of frameshift-induced matches, a significant number of probable sequencing errors and some of the remaining matches showed the possible usage of frameshift mutations and sequence inversion to form new amino acids sequences from pre-existing genes [36].

However, Guan and Uberbacher [32] created a dynamic programming method similar to Smith-Waterman, which can detect (find the insertion/deletion locations) and correct frameshifts while comparing the translated sequences to the protein sequences in the database. In addition, it finds optimal alignment, stays sensitive and accurate, and can recognize homologous proteins with 30% identity and 7% frameshift error rate. This method does not use heuristics or specific codon usage information. The model is also very general and only uses information given in the DNA sequences. The original algorithm developed by the researchers reduced the speed by a factor of seven compared to the standard Smith-Waterman but the modified version took one-fourth the speed of Smith-Waterman's algorithm. However, this model like the States and Botstein cannot deal accurately with insertion/deletion gaps within a codon or presence of introns within the sequence [33].

Huang and Zhang [37] explain two sensitive and powerful tools to find sequence homology: Nucleotide-Amino acid Alignment Program (NAP) for optimal global alignment, a modified version of States and Botstein method of local alignment model and Local Alignment Program (LAP) for optimal local alignment between a nucleotide sequence and a protein sequence that follows the Smith-Waterman algorithm. Unlike the

Guan-Uberbacher algorithm, these programs can deal with sequencing errors that lead to frameshifts within a coding region. In addition, these methods require computer memory proportional to the sum of the sequence lengths unlike the Guan-Uberbacher algorithm which needs quadratic space i.e., space equal to the product of their lengths. As a result, the LAP can handle longer sequences compared to this methodology.

Birney [38] devised an algorithm to find an optimal alignment of a protein sequence or profile against all three DNA translation frames allowing frameshifts. The algorithm implemented in the Wise tool program, allows a sequence comparison between protein sequences, protein profiles and DNA sequences. Programs PairWise and SearchWise (of the Wise tool package) deal with frameshifts caused by errors or introns. The author postulated that 1 in 200 sequences have frameshift error problems. Therefore, for SWISS PROT database v.32, there should be about 2000 sequences with errors.

Zhang et al. [39] use a methodology to find optimal alignment of DNA- protein sequences using FASTA mode of database searching programs. The algorithm developed by this group finds a hypothetical coding region (HCR) within the intron-free DNA sequence and aligns the conceptual translation of the HCR and the given amino acid sequence. Here scores are assigned for the alignment of the translated HCR and the given protein sequence and penalized for the differences between the DNA sequence and the HCR. Protein alignments are scored with an amino-acid substitution score matrix together with gap-open and gap-extension penalties. Among other approaches mentioned, the three-frame alignment model determines fewer kinds of errors and hence gives a faster algorithm for finding an optimal alignment. This model addresses

nucleotide insertions between adjacent codons and deletions where the last nucleotide of the last codon equals the first nucleotide of the next codon. However, this algorithm does not address base miscalls.

Nigel Brown [33] provides a tool for rapid screening of coding regions for frameshifts and in-frame stop errors in intron-free DNA sequences. It scans a query nucleotide sequence against a protein database and looks for homology matching after the query sequence is translated in all six frames using the BLASTX tool. In this tool, the potential ORFs for each protein are assigned along the query sequence with sequencing errors indicated as a by-product. The tool cannot conclude the presence of an ORF if there are no similarities in the protein database sequences. However, the method can correctly predict the downstream codon but cannot infer the correct start.

All the methods mentioned above for sequence alignment searches, scoring, and frameshift require different parameter settings to give highest scores. Therefore, comparisons are difficult to make as different researchers use different algorithms and settings. The least complex method, developed by Lustig et al. [8], uses binding energies between molecules as their basis. These energy values are unique and accurate as they are experimentally determined. The approach employed is to determine scoring by alignment energy. Here the DNA hybridization energies are obtained using canonical RNA-RNA base pairing or stacking and RNA loop energies developed experimentally by Turner et al. [9]. In particular, the scope of this study is to determine whether the lowest value of energy will correspond to the most stable alignment, which in this case should be

the original alignment. The shifts in alignments will be reflected in the values of the energy if they exist.

CHAPTER FOUR

HYPOTHESIS AND OBJECTIVES

4.1 Hypothesis

The hypothesis of this research is that the lowest value of hybridization energy will correspond to the original frame of alignment.

4.2 Objectives

The ultimate goal of this project is to use hybridization energy rules to develop a new tool for sequence alignment search that is less computationally intensive. In addition, this project includes the following tasks:

- Correct the errors that were generated in the nucleotide sequences by using Entelechon for back-translation [20]. This will result in retrieving exclusive coding sequences for the proteins that can be used to do a BLASTN search.
- Determine whether there exists a relationship between hybridization energy values and BLAST alignment scores and BLAST alignment length values.
- Explore the validity of alternative sequence alignments by introducing shifts in the alignments. Determine whether the lowest value of hybridization energy corresponds to the original alignment compared to its shifted version.
- Explore the potential utility of these energy based scoring methods to alignment shift and frameshift analysis.

CHAPTER FIVE

MATERIALS AND METHODS

5.1 Materials

Items needed for research include:

- Protein based [25] and ncRNA based [40] BLASTN output.
- Sequence alignment scoring program to calculate hybridization energy [26].
- PERL script to do a 6-frame shift between the query and the subject sequences.
- Personal computer.

5.2 Method

Figure 7 gives an overview of the method that was followed to calculate the hybridization energy (HE) of the alignments that is obtained from the BLASTN output.

Figure 7. Flow diagram of the method to calculate HE with shifts in alignments.

Shifts were applied to the alignments, and then alignments were evaluated based on their HE values. This was done to determine whether the best alignment in the BLASTN output does correspond to the lowest value of HE.

The following steps were taken:

1. Wander's protein based BLASTN output was used as a guide [25]. Additionally, BLASTN outputs generated using 17 ncRNA sequences from another study conducted by Barman et al. are also analyzed in this study [40]. It should be noted that both the studies cumulatively covered a broad spectra of organisms to best generalize the data. However, only 96 out of the 143 protein sequences and 11 out of the 17 ncRNA were successfully analyzed in this study. The problems are enumerated in the Results section (Chapter 6). Further, it is worth mentioning that only proteins whose structures were determined by high-resolution X-ray diffraction were taken into consideration.
2. Better BLASTN searches were generated with Wander's initial BLAST output. This was necessary since the back-translation of some of the protein sequences using Entelechon [41] yielded nucleotide sequences that did not generate appropriate alignments when a BLASTN was done. Wander's output was used to retrieve the nucleotide coding sequence (CDS) for each of the proteins. Each of the CDS was then used to do further BLASTN search. This method has been paraphrased as the "direct method" versus Wanders' "back-translated" method for differentiation purposes.

3. Aforementioned modified BLASTN search results were used for further analysis using a PERL script i.e., ExtractBlastnData.pl (see Appendix G) originally developed by Yang [26]. Important data including GI numbers, description, score, e value, identity, gap, alignment and start position were extracted and output was generated in a text format. Gene identifiers (GI) numbers are NCBI generated numeric values of one or more digits used to identify the sequence data existing in the GenBank database. A new integer GI is assigned if the sequence changes.
4. To perform a shift on the alignments, a PERL script was used (see Appendix G). A six frame-shift was done on the query / subject alignment. This was achieved by sliding the subject with respect to the query sequence. The direction of the shift is fixed in the following way:
 - a. 5' to 3' direction as the forward direction
 - b. 3' to 5' direction as the reverse direction
5. For each alignment, the subject sequence was moved in one direction one base pair at a time three times. In the forward direction, F1, F2, and F3 were assigned for each of the positions; and in the reverse direction, R1, R2, and R3 were assigned for each of the positions. This resulted in six shifted alignments for each original alignment.
6. The subject sequences of the shifted alignments were then converted into their complementary sequences.

7. The free energy values of the query sequence with the complement of the subject sequences were calculated using another PERL script (see Appendix H) developed by Yang i.e., CalculateBlastData.pl. The matches represented by the annealed regions were calculated using their base pairing energy values. The gaps and mismatches were represented by bulges and internal loops which were then evaluated by loop values given by Turner et al.[9]
8. Then the value of energy was summed over the entire length of the sequence to get values for an alignment pair.
9. These energy values were then used to evaluate the alignment with respect to their corresponding BLASTN score.

5.3 Proposed Analysis

The output of the PERL script is converted to a format that can be analyzed on Microsoft Excel. This analysis will help answer the questions that form the objective of this project. Various graphs will be plotted to verify relationships. The different plots that are analyzed are as follows:

- Hybridization Energy (HE) versus BLAST Bit Score
- Hybridization Energy (HE) versus BLAST Alignment Length

5.3.1 Hybridization Energy versus BLAST Bit Score

Once forward or reverse shifts are applied to the original alignment, there are changes expected to be observed in the corresponding scores for that alignment. However, BLAST cannot calculate the change in score value for every alignment shift that is made. Hence, in this case, hybridization energy values for the original alignments

are plotted against their scores to observe if there exists a one-to-one correspondence between their values.

The BLAST bit scores are plotted on the X-axis and the hybridization energies are plotted on the Y-axis. Trend lines are plotted to determine the slopes and intercepts that define a quantitative relationship between score and hybridization energy value.

5.3.2 Hybridization Energy versus BLAST Alignment Length

Another objective of this project is to determine whether there exists any relationship between hybridization energy values and their BLAST alignment length obtained from the BLAST report. Once the shifts are introduced for each alignment, the resulting alignment lengths corresponding to each shift in position is plotted against its hybridization energy value. Here it is seen that for each original alignment that is obtained from the BLAST report, there are seven values for hybridization energies i.e., six shifted variants of the original alignment and one for the original alignment.

Here the BLAST alignment length is plotted on the X-axis and the hybridization energy values is plotted on the Y-axis. Trend lines are plotted for each of the shifted variants i.e., F1, F2, and F3 for the forward direction shifts and R1, R2, and R3 for the reverse direction shifts. Their slopes and intercepts are determined to define a relationship between the data plotted. Here the slope gives the average hybridization energy per nucleotide. If the lowest slope value corresponds to the trend line with the original alignment, then the hypothesis is met.

Once the lines are plotted, error bars are calculated that will give an estimate of standard deviation for the energy value [42]. In addition, the R-value or the correlation

coefficient can be determined for each of the lines. These values allow us to calculate the P value. A low P value for the test (less than 0.05) means there is a statistically significant linear relationship between the two variables. The P values calculated for both the plots i.e., HE versus BLAST Scores and HE versus Alignment Length, give an insight to the relationship between these parameters.

A more global analytical approach is required to judge the use of alignment shift methods to provide insights into the development of future heuristics for sequence alignment and related algorithms. In addition to correlation plots and their statistics, trend analysis is also required to develop a method for sequence alignment search. A possible methodology would be to compare the relative slopes and intercepts for one particular type of shift, e.g., F1 among different proteins. Another method would be to consolidate alignments of all shifts and compare to their corresponding HE values. In addition, non-shift data can be further addressed to eliminate any errors due to back-translation and improper coding sequence truncation.

CHAPTER SIX

RESULTS AND DISCUSSION

One of the main objectives of this thesis was to generate better DNA alignments from directly determined sequences using back-translated protein sequences from an earlier study [25]. The earlier study computed DNA alignments using BLASTN for 143 back-translated protein sequences and 10 RNA sequences. In the current study, the 10 RNA sequences were excluded and replaced with 11 of the 17 non-coding RNA sequences (ncRNA) from Barman's study [40]. A detailed list of the query proteins and ncRNA along with their source and common names is given in Appendix A. The BLASTN search correctly identified the organisms that produced significant alignments only for 96 of the 143 back-translated protein sequences as shown in Table 2. For all further analysis, this study uses the "directly determined sequences" from the NCBI database corresponding to the 96 significant back-translated protein sequences.

Typically, protein names are created as unique identifiers for specific proteins from specific organisms. For clarity and future development, this study has developed a new identifier by binding the accession number to the protein name. This new identifier is formed by concatenating the alphanumeric four-character protein identifier and the alphanumeric accession number with an underscore (e.g., 1sgt_M64471, Trypsin). All the tables and plots throughout this study use this encoded protein name terminology.

Table 2. “Proteins_Accession Number” used in this study.*

1a3c_Z99112	1aqh_X58627	2rn2_V00337	3psg_J04601
1a3s_AJ002385	1lz1_NM_000239	2rsp_V01197	3rnt_D28341
1a5z_X74302	1omd_J02705	2sga_M17103	3rp2_J02712
1a6q_S87759	1rbp_NM_006744	2sni_K02496	3sgb_M17104
1aat_X15636	1rhd_M58561	2ssi_M54887	3tln_X76986
1ab4_X06373	1sgt_M64471	2stv_L06057	4ape_X63351
1abm_X07834	2act_AF343446	2taa_M33218	4dfr_J01609
1add_NM_007398	2hmg_AJ427297	2tbv_AJ249740	4fxn_J05015
1adi_J04199	2cab_M33987	2tim_X03921	4mdh_U44846
1ae4_U46064	2cdv_D31702	2trx_M26133	4pep_M20920
1af3_U72350	2ci2_X57035.1	2tsl_J01546	4rhv_K02121
1agm_K02465_G1	2cpp_M12546	2tsc_J01710	4sbv_L34672
1ahn_M59426	2cro_V00635	2wrp_J01715	4tnc_M19027
1ai2_J02799	2cts_M21197	3app_AF295432	4xia_X59466
1ak2_M16225	2cyp_J01468	3blm_U58139	5acn_Z49931
1ako_X13002	2gn5_V00604	3cla_X53796	5cpv_AJ292212
1al8_J03492	2lbp_M29378	3cna_AF308777	5rsa_S81740
1aln_M60916	2ldx_L10389.1	3est_X04036	5rub_X00286
1amn_X56517exon2	2ltn_X66368	3gpd_J02642	6ldh_U38893
1amp_M85159	2lzt_V00428	3grs_NM_000637	7api_X01683
1an7_X79551	2mev_M88547.1	3icb_M18344	8adh_M64864
1an9_M16972	2pab_U19780	3pfk_D13095	8atc_K01472
1ao5_NM_010115	2plv_V01149	3pgk_K00553	9pap_M15203
1aob_J01710	2prk_X14689	3pgm_X58789	9wga_M25536

*A brief description including the sources and the common names of the proteins is given in Appendix A.

Table 3 shows the proteins that could not be used for various reasons. Column A lists the proteins that did not produce significant alignments in the BLASTN output. Consequently, the coding sequences (CDS) for those proteins were not retrievable in order to do further analysis. Column B gives the eight proteins that belong to the immunoglobulin family and not used in this study. Column C lists five proteins that were removed because they generated alignments in the excess of 2000, which might have given biased values for the plots. Two proteins in Column D were removed from the

NCBI protein database and Column E lists two proteins that were not found in the NCBI protein database. Column F lists the proteins that are redundant i.e., found to be similar to other proteins used in this study. Column G represents the proteins that have a low-resolution X-ray structure.

Table 3. Proteins not used in the study.

A	B	C	D	E	F	G
1a48	2fb4	1tgn	1als	2civ	1a59	2tmv
1acb	2hfl	1ubq	2gcr	2hla	1adb	
1agx	2liv	2ptn	total=2	1ab3	3dfr	
1alc	2fbj	4hvp		total=3	3lzm	
1azi	3fab	5cpa			3rn3	
1crc	2fxb	total=5			5ldh	
1ton	4fab				total=6	
2cga	3mcg					
2lz2	total=8					
2mhb						
2pka						
3adk						
3cln						
3gbp						
3hfm						
5cha						
5cts						
5tnc						
6xia						
7cat						
7pcy						
8dfr						
total=22						

Out of the 17 ncRNA sequences [40], only 11 were finally used for analysis as shown in Table 4. Table 5 enumerates the ncRNA that were not a part of this study for reasons listed. Three of them (as shown in Column A) had CDS that excludes them from the category of ncRNA. In two of them (Column B), the genes mentioned were not a part

of the significant alignments and one (in Column C) had alignments in the excess of 2000 points.

Table 4. ncRNA used.*

DicF_X07465
micF_X04888
UptR_AF272839
HIF_U85044
PCGEM1_AF223389
SRA1_AF092038
Igf2_AB030734
Msx1_AF267728
DsrA_AE008788
RprA_AE008759
CsrB_NC_003197

Table 5. ncRNA not used.

A	B	C
hoxA_AF071164	Se_Oxys	Mus_Borg
Igf2_NM_016412	St_MicF	
Sra1_NM_025291		

*A brief description including the sources and the common names of the proteins is given in Appendix A.

6.1 Frequency Distribution of Number of Base Pairs in the Query Sequence

The CDS were determined for each of the 96 directly determined sequences from the NCBI database, which in turn was used as the query to do the BLASTN search to generate alignments. The diagonal pattern bars in Figure 8 show the frequency distribution of the number of bases per query sequence that were used to do the BLASTN search. It was seen that most number of query sequences had base pairs in the ranges 501-1000 and 1001-1500. However, there were no sequences representative of the following ranges of 3001-3500, 3501-4000, 4001-4500, 4501-5000, 5001-5500, 5501-6000, and 6001-6500. The checker pattern bars in the same figure show the values obtained from back-translated data [25] for the same proteins. Here it was seen that most number of query sequences had base pairs in the range of 501-1000.

Figure 8. Frequency distribution of the number of base pairs in the 96 protein-based query sequences. The chart (diagonal pattern) shows the number of base pairs in the query sequence for each of the 96 back-translated proteins that were used to conduct the BLASTN search. There are a maximum of 32 query sequences, each with number of base pairs in the range of 501-1000 and 1001-1500. There were no query sequences with number of base pairs in the ranges of 3001-3500, 3501-4000, 4001-4500, 4501-5000, 5001-5500, 5501-6000, and 6001-6500. The mean, median, and mode for this distribution were found to be 1067, 918, and 330, respectively. The checker pattern bars show the frequency distribution for the same 96 back-translated proteins [25]. In this case, a maximum of 48 sequences were found to be in the range of 501-1000.

Table 6 illustrates the difference in the number of base pairs in the query sequences between the back and directly translated method for a sample of 14 sequences. Note that for the same protein, there are sequences with marginal differences (3-10 base pairs) and sequences with 100-5000 base pairs apart. Here Column A represents the number of base pairs retrieved using the CDS from the NCBI database and column B indicates the number of base pairs using Entelechon [25, 41].

Table 6. Sample sequences derived from back-translation versus direct method.

protein_accession #	A	B	A-B
1a3c_Z99112	546	543	3
1a3s_AJ002385	477	480	-3
1a5z_X74302	960	957	3
1aat_X15636	1239	1233	6
1ab4_X06373	2628	1431	1197
1af3_U72350	702	588	114
1amp_M85159	1299	873	426
1aqh_X58627	2010	1359	651
2ltt_X66368	828	543	285
2mev_M88547	2703	831	1872
2plv_V01149	6630	906	5724
4rhv_K02121	6540	867	5673
9pap_M15203	1038	636	402

Figure 9 gives the base pair distribution for 11,552 base pairs in the ncRNA sequences used as query for this study. It is seen that highest number of five query sequences lie in the range of 0-200. Ranges 401-600, 601-800, 1201-1400, 1401-1600, and 1801-2000 were not represented in this study.

Figure 9. Frequency distribution of the number of base pairs in the 11 ncRNA query sequences [40]. The bars show the number of base pairs in the query sequence for each of the 11 ncRNA sequences that were used to conduct the BLASTN search. There were a maximum of five query sequences with number of base pairs in the range of 0-200. There were no query sequences with number of base pairs in the ranges of 401-600, 601-800, 1201-1400, 1401-1600, and 1801-2000. The mean and median of the distribution were determined to be 1050 and 355, respectively. Due to the limited size of the data set, no valid mode was determined for the distribution.

6.2 Frequency Distribution of Number of Alignments per Query Sequence

Figure 10 shows the frequency distribution of the number of alignments per query sequence. The default was set to the highest for the BLASTN search. The number of hits for all the 96 queries totaled 26,702. All the diagonal pattern bars show the results for this study. There were 58 query sequences in the alignment range 0-200 and no queries generated outputs in the ranges 1401-1600 and 1801-2000. Five query sequences (as shown in Table 3) generated alignments in the excess of 2000 that were deleted from the data set since these points would have given biased plots. Minimum number of 2 hits was observed in the ranges 801-1000 and 1601-1800. The checker pattern bars give the number of alignments for the same back-translated query sequences [25]. It should be noted that the 88 back-translated query sequences generated alignments in the range 0-200. In addition, 5 back-translated queries exhibited alignments in the range 201-400 and two in the range 801-1000.

Figure 10. Frequency distribution of the number of alignments per query sequence for the 96 protein-based DNA sequences. The chart (diagonal pattern bars) shows the number of alignments that were generated in the BLASTN search for each of the 96 sequences in this study. It was seen that there were a maximum frequency of 58 alignments in the 0-200 range and there were no alignments in the ranges of 1201-1400 and 1401-1600. Here the mean, median, and mode of the distribution were found to be 278, 143, and 65, respectively. The checker pattern bars show the frequency distribution for the same 96 back-translated proteins [25]. In this case, a maximum of 88 query sequences gave alignments in the range of 0-200.

The 11 ncRNA sequences generated 3252 alignments. Figure 11 shows the distribution for the ncRNA sequences. It was observed that 0-50 range had the highest number of alignments per query. There were no alignments in the ranges of 251-500, 501-750 and 1251-1500.

Figure 11. Frequency distribution of the number of alignments per query sequence for the 11 ncRNA sequences [40]. The chart shows the number of alignments that were generated in the BLASTN search for each of the 11 ncRNA sequences in this study. There was a maximum frequency of 4 alignments in the 0-50 range and there were no alignments exhibited in the ranges of 201-250, 251-500, 501-750, 751-1000, and 1251-1500. Here the mean and median were calculated to be 296 and 68, respectively. No valid mode was found for the distribution.

6.3 Frequency Distribution of Bit Scores

Figure 12 shows the frequency distribution of the bit scores for all the 26,702 alignments from the BLASTN search for the 96 directly determined sequences. As seen in the figure, the distribution peaks at 8310 in the range 0-50. No alignments exhibited bit scores between the ranges 6001-6500, 6501-7000, 7001-7500, 8501-9000, 9001-9500, and 9501-10000.

Figure 12. Frequency distribution of the bit scores from the BLASTN searches for the 96 directly determined DNA sequences. A total of 26,702 alignments showed a maximum of 8310 alignments with bit scores in the range of 0-50. There were no alignments with scores in the ranges of 6001-6500, 6501-7000, 7001-7500, 8501-9000, 9001-9500 and 9501-10,000. Here the mean, median, and mode of the distribution were calculated to be 259, 68, and 42, respectively.

Figure 13 exhibits the frequency distribution of the bit scores from the BLASTN searches for the 11 ncRNA sequences [40]. A total of 3252 alignments showed a maximum of 2039 alignments with bit scores in the range of 0-50. Ranges 701-800, 901-1000, 3001-3500, and 3501-4001 were not represented at all.

Figure 13. Frequency distribution of the bit scores from the BLASTN searches for the 11 ncRNA sequences [40]. A total of 3252 alignments showed a maximum of 2039 alignments with bit scores in the range of 0-50. A minimum of 1 alignment was observed in ranges 801-900, 1001-1500, and 2501-3000. Conversely, ranges 701-800, 901-1000, 3001-3500, and 3501-4001 were not represented at all. The mean, median, and mode were determined to be 109, 46, and 46, respectively.

6.4 Correlation Plots

The correlation plots were made for two main categories:

1. Hybridization Energy (HE) versus BLASTN score
2. Hybridization Energy (HE) versus Alignment Length

6.4.1 Correlation Plots for Hybridization Energy versus BLASTN Score

- a) Discrete plots of alignments for each of the 96 directly determined DNA sequences and 11 ncRNA sequences (Appendix C).
- b) Two aggregate plots, one comprising of all the 26,702 alignments from the 96 directly determined DNA sequences and the second of the 3252 alignments from the 11 ncRNA sequences (Figures 15 and 16).

In all plots, the BLAST score is plotted on the X-axis and the computed hybridization energy values on the Y-axis. A linear relationship is observed between the BLASTN scores and their corresponding hybridization energy values in each of the plots. The linear curve-fit equation and the R-squared value for each of the plots are shown in the plots.

The individual plots were generated for each of the 96 back-translated proteins and 11 ncRNA sequences (as shown in Appendix C). A sample of an individual plot is shown in Figure 14. Note that the heading 1sgt_M64471oa gives the ID of the protein i.e., 1sgt followed by the accession number for the CDS of the query M64471 followed by “oa” which stands for “original alignment” which implies that the plot was done for the original alignment. It should also be noted that all the plots under this category could be done only for the original alignments, since BLAST does not allow the scores to be

computed for shifted alignments. Additionally, even though the sample plot (Figure 14) shows approximately 15 points, in reality there are 510 points that represent the number of alignments obtained in the BLASTN search. Many of the points overlap which is consistent in all the plots.

Figure 14. Correlation plot of alignments for directly determined Trypsin (1sgt); number of base pairs = 1059; number of alignments = 510. The correlation coefficient is -0.989; $P < 0.001$.

The 2 aggregate correlation plots as shown in Figures 15 and 16 reflect the 26,702 DNA alignments from the 96 directly determined DNA sequences and the 3252 alignments from the 11 non-coding RNA sequences respectively. Similar to the discrete

plots, a linear correlation was observed and hence a linear trend line was opted. Here the slopes are seen to be very close in value ($m_{\text{cDNA}} = -1.08$, $m_{\text{ncRNA}} = -1.02$) and the P value which indicates the probability that a random set of data with an equivalent number of points exceeds the R-value is very low i.e., $P < 0.001$ [42].

Figure 15. Aggregate correlation plot of Hybridization Energy versus BLASTN score for the 26,702 alignments of the 96 directly determined DNA sequences. Here the hybridization energies (ordinate) is plotted against the BLASTN score (abscissa) for the original alignments (oa) only. The equation for the straight line is shown on the plot and the correlation coefficient is computed to be ± 0.97 . The estimated calculated uncertainty [42] for the slope is ± 0.0017 and the intercept is ± 1.041 (see Appendix F).

Figure 16. Aggregate correlation plot of Hybridization Energy versus BLASTN score for the 3252 alignments of the 11 ncRNA sequences. Here the hybridization energies (ordinate) is plotted against the BLASTN score (abscissa) for the original alignments (oa) only. The equation for the straight line is shown on the plot and the correlation coefficient is computed to be ± 0.9834 . The estimated calculated uncertainty [42] for the slope is ± 0.0033 and the intercept is ± 0.9653 (see Appendix F).

Another plot in the same category is a correlation plot of hybridization energy values to their corresponding BLASTN scores for the subset of protein-based alignments indicating significant sequence divergence (with Expect values > 0.01). All the individual plots for 96 directly determined DNA sequences and the 11 ncRNA sequences are shown in Appendix E. For each of these plots, a linear curve fit equation and the R squared value, are shown in the graph. A sample of this plot is shown in Figure 17.

Figure 17. Correlation plot of Hybridization Energy versus BLASTN scores for the subset of protein-based alignments of Trypsin (1sgt) indicating significant sequence divergence i.e., with Expect values > 0.01. Correlation coefficient is ± 0.5255

However, there are some proteins where all the alignments generated in the BLASTN search have Expect values < 0.01 . Hence, no plots with $E > 0.01$ were generated for these proteins e.g., 2hmg and 2plv. An aggregate plot of a subset of protein-based alignments to observe the cumulative effect of all the points with significant sequence divergence (with Expect values > 0.01) is shown in Figure 18.

Figure 18. Aggregate correlation plot of 6,142 DNA alignments that form a subset of the 26,702 alignments that have the Expect values, $E > 0.01$. Here the hybridization energies (ordinate) is plotted against the BLASTN score (abscissa) for the original alignments (oa) only. The equation for the straight line is shown on the plot and the correlation coefficient is computed to be ± 0.4909 .

6.4.2 Correlation Plots for Hybridization Energy versus Alignment Length

- a) Individual plots for alignments for each of the 96 directly determined DNA sequences and 11 ncRNA sequences (Appendix D).
- b) Two aggregate plots, one comprising of all the 26,702 alignments from the 96 directly determined DNA sequences and the second of the 3254 alignments from the 11 ncRNA sequences (Figures 20 and 21).

In all the plots in this category, the BLASTN calculated alignment length is the abscissa and the corresponding hybridization energy is the ordinate. In addition, alignment shifts were incorporated as -3 to +3 nucleotides in increments of one, totaling to six shifts per alignment. In these plots, the slope that gives a measure of the average energy per aligned nucleotide for each of the shifted variants i.e., l1, l2, l3, r1, r2, and r3 will be seen along with the original alignment (oa) in each of the 96 directly determined DNA sequences and 11 ncRNA figures as shown in Appendix D. A linear trend was observed in each of plots. The linear fit equation along with the R squared value for each variant is shown in each figure. An example of an individual plot is shown in Figure 19.

Figure 19. Correlation plot of Hybridization Energy versus corresponding Alignment Length for the protein-based alignments of Aminopeptidase (1amp). The graph shows the linear fit equation for the original alignment and each of the shifts along with their corresponding R-squared values.

Shifting any one of the sequences in the alignment (the subject sequence is shifted in this study) results in a higher unstable value of hybridization energy than the original alignment. This trend is reflected in the values of slopes shown in all the plots. It can also be seen that the original alignment still has the lowest value of slope, which corresponds to the energy per nucleotide in the alignment. Lower slope values correspond to greater stability of the alignment. This correlation further corresponds to stronger bonds of attraction between the query and the complement of the subject.

Figures 20 and 21 show the aggregate correlation plots for the 26,702 DNA alignments for the 96 directly translated protein sequences and for the 3252 alignments for the 11 ncRNA sequences respectively. Since the slope-values for the shifted variants were found to be almost identical in both the plots, an average for all the shifts in each data set was taken and then plotted as shown in Figures 22 and 23.

Figure 20. Aggregate correlation plot of Hybridization Energy (ordinate) versus Alignment Length (abscissa) for BLASTN alignments of 96 directly translated protein sequences. Here, the relation between alignment lengths to their corresponding energies for 26,702 DNA alignments is being compared for the original alignment (oa) to various shifts in the alignment i.e., r1 (subject shifted one position to the right), r2 (subject shifted two positions to the right), r3 (subject shifted three positions to the right), l1 (subject shifted one position to the left), l2 (subject shifted two positions to the left), and l3 (subject shifted three positions to the left). The expression for each of the fits is shown in the plot. The correlation coefficients for each of the straight lines are as follows: oa- ± 0.9690 , r1- ± 0.9929 , r2- ± 0.9975 , r3- ± 0.9893 , l1- ± 0.9942 , l2- ± 0.9966 and l3- ± 0.991 . $P < 0.001$ for all seven fits.

Figure 21. Aggregate correlation plot of Hybridization Energy (ordinate) versus Alignment Length (abscissa) for BLASTN alignments of 11 directly translated ncRNA sequences. Here, the relation between alignment length to the corresponding energies for 3252 alignments is being compared for the original alignment (oa) to various shifts in the alignment i.e., r1 (subject shifted one position to the right), r2 (subject shifted two to the right), r3 (subject shifted three positions to the right), l1 (subject shifted one position to the left), l2 (subject shifted two positions to the left), and l3 (subject shifted three positions to the left). The expression for each of the fits is shown in the plot. The correlation coefficients for each of the straight lines are as follows: oa- ± 0.9836 , r1- 0.9846 , r2- ± 0.9917 , r3- ± 0.9573 , l1- ± 0.9813 , l2- ± 0.9925 and l3- ± 0.9557 . $P < 0.001$ for all seven fits.

Figure 22. Aggregate correlation plot of the averaged shifted energy values for the 26,702 DNA alignments for the 96 directly translated protein sequences. Since the slopes for the various shifts were found to be very close in values, the energy values for the various shifts for each alignment were averaged out and then plotted against their corresponding alignment lengths. The expressions for the straight-line fits are shown on the plot itself. The correlation coefficients for the fits are as follows: oa: ± 0.9690 and average (shifts): ± 0.9975 .

Figure 23. Aggregate correlation plot of the averaged shifted energy values for the 3252 alignments of the 11 directly translated ncRNA sequences. Since the slopes for the various shifts were found to be very close in values, the energy values for the various shifts for each alignment were averaged out and then plotted against their corresponding alignment lengths. The expressions for the straight-line fits are shown on the plot itself. The correlation coefficients for the fits are as follows: oa: ± 0.9836 and average (shifts): ± 0.9902 .

6.5 Discussion

Referring to Figure 8, observe the disparity between the number of query base pairs calculated in this study and the back-translated form [25] for the same sequences. For the same 96 back-translated amino acid sequences, the number of base pairs totaled to 78,087 compared to 102,395 base pairs in this study using directly determined sequences. Once the amino acid sequences for different proteins were obtained from Protein Data Bank (PDB), Entelechon was used to back-translate these sequences to their corresponding nucleotide sequences [25, 41]. Entelechon follows the standard genetic code for back-translation applying the principles of codon bias with respect to organism type. In this study, the query sequences were obtained by refining the back-translated sequences [25], by procuring the CDS for the sequences from the NCBI database and then subsequently using each of the CDS as the query to do a BLASTN search. The method of obtaining the CDS for the nucleotide query directly from the database can be considered more reliable compared to determining the nucleotide sequence by back-translation using the standard genetic code in that error can be induced in the process of doing so as shown in Table 6.

Subsequently, if the query sequence is not accurate while doing the BLASTN searches, it follows that the number of appropriate hits received per search might not be accurate either. This anomaly is reflected in Figure 10. Here it is seen that a maximum of 58 directly determined sequences gave hits in the range of 0-200 (shown by the diagonal pattern bars) versus the 88 back-translated sequences [25] in the same range (shown by the checker pattern bars). Observe that there seems to be a better-behaved

distribution of the number of alignments in the former approach versus the latter, which could be attributed to the errors induced in the process of back-translation.

The slopes of all plots are negative for the individual (Appendix C) as well as the aggregate hybridization energy versus BLAST bit score plots (Figures 15 and 16) for both the directly determined DNA sequences from the proteins as well as the ncRNA alignments. In addition, the correlation coefficients (-0.85 to -1.0) for the individual plots as shown in Appendix B indicate a strong linear correlation between the BLAST bit score and their corresponding hybridization energy values. This implies that as the bit score increases, the hybridization energy (binding energy between the query and the complement of the subject) decreases in value i.e., more negative slope equates to more stable bonds. Observing Figures 15 and 16, both aggregates show a slope of approximately -1 which is indicative of a one-to-one relationship between hybridization energy and the BLAST bit score. Slopes closer to -1 imply that energy-to-score relationship is self normalizing.

Even in the most divergent sequences as shown in Figure 18, it is seen that there exists a correlation between the BLAST bit score and their energy values ($R = \pm 0.4909$). However, the magnitude of their slopes as seen in the correlation plot does not show unity suggesting the need for further resolution of this correlation.

Using energies to score alignments effectively is not unusual where a scoring matrix has been derived via contact potentials for proteins [43]. Here the increment in average energy of the native structure is determined by using the amino acid contact energy and the number of contacts for each type of amino acid pair, which can then be

used to evaluate the degree of instability of the native structure by an amino acid exchange. A transition probability matrix associated with mutation similar in method to 250 PAM is derived which reflects the base mutation rates and the effects of the genetic code and conservation against amino acid exchanges. Even though the methodology used to score alignments in our case is also energy based, it is more direct and not derived as is in the aforementioned study from crystal structures.

The energy-based method makes the task of calculating energies for the alignments for each shift easy as compared to BLAST where scoring an alignment using “bl2seq” can be done only one alignment at a time. This can be a daunting task (but not impossible) especially when a BLAST search can retrieve an average of about 250 hits. Furthermore, adding six shifts to each alignment and evaluating the alignment for each shift can prove formidable in BLAST. However, the energy based system of evaluating alignments that employs the RNA structure energies applied to DNA alignments [8, 9] can be utilized with ease to any manipulated alignments. Since the RNA structure energies are experimentally determined, they can be easily used in a look up table to compute energies for different shifted alignments. A match in the DNA alignment is analogous to a complementary matching region in an RNA molecule and is given a negative energy value and the mismatches and gaps modeled as loops in the RNA are awarded positive values. Subsequently, the calculation of slopes for each shift is done with ease using the energy-based system of evaluation.

For the original alignments, there exists a strong correlation between the energy and the corresponding alignments length. Correlation coefficient values for the

individual plots range from -0.95 to -1.0 (refer to Appendix B) show a strong statistically significant linear relationship.

One of the main reasons to introduce shifts in the alignments from -3 to +3 in increments of 1 was to observe the change in energy values with respect to each shift and to subsequently explore the validity of the shifted alignments. A secondary reason was to investigate if a shift could be characterized from its observed slope value. Interestingly, the slopes for each shifted variant was found to be ~ 0.4 which is self-consistent in that the correlation is random. It was observed that the shifts were not similar in pattern to the original alignment. It should also be noted that the slopes of shifted variants were flat compared to the original alignment, which alludes to the fact that the correlation plots of the original alignment are not random. The positive slope values for the shifted variants give positive values for hybridization energy, which indicates instability in the alignment that is consistent with our expectations of the original alignment being the most stable with the lowest value of binding energy.

Employing error analysis in BLAST involves complicated statistics requiring large exhaustive sample data. Here the average energy per nucleotide approach offers a relatively simpler and direct way of doing error analysis. Using the energy-based approach one can generate simpler error statistics for Smith-Waterman and subsequent heuristics.

Applying shifts to the alignments is in itself a form of error analysis. It can be said that the alignments were manipulated to an extent by applying shifts and the energies

of the subsequent alignments have been evaluated to prove that the original alignment in all the cases still exhibit the lowest and the most stable value of binding energy.

Interestingly, the Lustig group has developed an energy based Smith-Waterman algorithm [44]. In a later version of the algorithm, the same sequence alignments and RNA derived energies were analyzed. For protein-coding and non-coding RNA alignments strong correlations were shown between the energy based Smith-Waterman and the BLASTN results. In addition, the aggregate analysis plots of hybridization energy versus alignment length showed similar results including differences in energy per nucleotide for protein coding regions compared to non-coding RNA. Applications of DNA hybridization energies [45] remain incomplete with respect to the testing of such energies.

CHAPTER SEVEN

CONCLUSIONS

Hybridization energy (HE) initially developed for the prediction of secondary and the three-dimensional structure of RNA by Turner et al. [9] was used to evaluate nucleotide sequence alignment since there is a significant correlation in free energies associated with DNA-DNA hybridization to those associated with RNA-RNA hybridization [10]. This similarity in energy properties, along with appropriate representation of mismatched and gapped regions in nucleotide sequence alignment by the loop structures in RNA, prompted the use of RNA hybridization rules to the DNA alignment search.

This study constitutes a detailed analysis of alignments procured from a BLASTN search of 96 directly translated proteins and 11 non-coding RNA sequences. A statistically significant linear relationship is observed in the individual as well as aggregate plot of BLASTN score of the sequence alignments versus their corresponding hybridization energy values. This relationship is also observed in the subset of protein-based as well as ncRNA alignments indicating significant sequence divergence (with Expect values > 0.01). Shifts were introduced as a first step in evaluating statistical significance of the original alignments by energies. By incorporating the six shifts to the alignments, and then plotting the energy values for each of the shift against their corresponding alignment length along with that of the original, it is interesting to notice that the original alignment still has the lowest slope value. From the 27,602 alignments

procured from the BLASTN search, each forward shift (F1, F2, and F3) had a positive slope. The same was observed in each of the reverse shifts (R1, R2, and R3). The positive values of the slopes for all the shifts signify that the alignments are more unstable once shifted and an average value of energy of 0.4 kcal/mol per nucleotide for each of the shifts indicate that the shifts could not be characterized by their slope value alone. Further research in this area would probably get results that are more promising.

A similar pattern was observed in the analysis of the sequences generated from the BLASTN search of the 11 non-coding RNA sequences as well. However, the slopes were found to be more negative for the non-coding RNA plots as compared to the protein plots. Here considering the limited size of the data set, further analysis using a more diverse and expanded set of sequences is recommended.

REFERENCES

1. C. Gibas and P. Jambeck, Developing Bioinformatics Computer Skills, 1st ed. (O'Reilly & Associates, Inc. California, U S A, 2001), pp. 3-21.
2. S. Kumar (2005). Bioinformatics web - Comprehensive educational resource on Bioinformatics [Online]. Available at <http://www.geocities.com/bioinformaticsweb/his.html> (accessed December 2005). WWW article.
3. D.J. Lipman and W.R. Pearson, "*Rapid and sensitive protein similarity searches*," Science, **227**, pp. 1435-1441 (1985).
4. F. Tekaa, (reference is undated). FASTA Algorithm [Online]. Available at http://bimas.dcrn.nih.gov/fastainfo/fasta_algo (accessed February 2004). WWW article.
5. S.F. Altschul, W. Gish, W. Miller, E.W. Myers and D.J. Lipman, "*Basic local alignment search tool*," J. Mol. Biol. **215**, pp. 403-410 (1990).
6. National Center for Biotechnology Information (2005). Glossary of Terms [Online]. Available at <http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html> (accessed December 2005). WWW article.
7. National Center for Biotechnology Information (2005). Growth of Genbank (1982-2004) [Online]. Available at <http://ncbi.nih.gov/Genbank/genbankstats.html> (accessed December 2005). WWW article.
8. D.S. Wander, F. Yang, M.M. McNeil and B. Lustig, "*Scoring DNA Sequence Alignment Using Energetics of Hybridization*," (Proceedings of Annual Meeting of AIChE T2/555b, 2003).
9. D.H. Turner, N. Sugimoto and S.M. Freier, "*RNA structure predictions*," Ann. Rev. Biophys. Biophys. Chem. **17**, pp. 167-192 (1988).

10. J. SantaLucia, H.T. Allawi and P.A. Seneviratne, "*Improved nearest neighbor parameters for predicting DNA duplex stability*," *Biochemistry*, **35**, pp. 3555-3562 (1996).
11. W.R. Pearson and D.J. Lipman, "*Improved tools for biological sequence comparison*," *Proc Natl Acad Sci U S A*, **85**, pp. 2444-2448 (1988).
12. Wikipedia, The Free Encyclopedia (2005). DNA [Online]. Available at <http://en.wikipedia.org/wiki/DNA> (accessed November 2005). WWW article.
13. Wikipedia, The Free Encyclopedia (2005). Central dogma of molecular biology [Online]. Available at http://en.wikipedia.org/wiki/Central_dogma (accessed November 2005). WWW article.
14. Genetics Home Reference, A service of the U.S. National Library of Medicine (2005). Frameshift mutation [Online]. Available at <http://ghr.nlm.nih.gov/ghr/glossary/frameshiftmutation> (accessed November 2005). WWW article.
15. National Center for Biotechnology Information (2005). NCBI BLAST [Online]. Available at <http://www.ncbi.nlm.nih.gov/BLAST/> (accessed December 2005). WWW article.
16. Access Excellence Resource Center (2005). Nucleic Acid Hybridization [Online]. Available at <http://www.accessexcellence.org/RC/VL/GG/nucleic.html> (accessed November 2005). WWW article.
17. H.B. Nicholas Jr., D.W. Deerfield II and A.J. Ropelewski, "*Strategies for searching sequence databases*," *Biotechniques*, **28**, pp. 1174-1191 (2000).
18. S.B. Needleman and C.D. Wunsch, "*A general method applicable to the search for similarities in the amino acid sequence of two proteins*," *J. Mol. Biol.* **48**, pp. 443-453 (1970).
19. P.H. Sellers, "*On the theory and computation of evolutionary distances*," *SIAM J. Appl. Math.* **26**, pp. 787-793 (1974).

20. S. Henikoff, "*Scores for sequence searches and alignments*," Curr. Opin. Struct. Biol. **6**, pp. 353-360 (1996).
21. M.O. Dayhoff, R.M. Schwartz and B.C. Orcutt, "*A model of evolutionary change in proteins*," in Atlas of Protein Sequence and Structure, edited by M.O. Dayoff (National Biomedical Research Foundation, 5, Washington, 1978), pp. 345-352.
22. S. Henikoff and J.G. Henikoff, "*Amino acid substitution matrices from protein blocks*," Proc Natl Acad Sci U S A, **89**, pp. 10915-10919 (1992).
23. G.H. Gonnet, M.A. Cohen and S.A. Benner, "*Exhaustive matching of the entire protein sequence database*," Science, **256**, pp. 1443-1445 (1992).
24. G. Vogt, T. Etzold and P. Argos, "*An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited*," J. Mol. Biol. **249**, pp. 816-831 (1995).
25. D.S. Wander, RNA structure energies applied to genome sequence search, (San Jose State University, San Jose, CA, 2002).
26. F. Yang, "*Sequence alignment scoring with DNA hybridization thermodynamics*," (San Jose State University, San Jose, CA) *unpublished results (November 2002)*.
27. J. SantaLucia Jr., "*A unified view of polymer, dumbbell and oligonucleotide DNA nearest-neighbor thermodynamics*," Proc Natl Acad Sci U S A, **95**, pp. 1460-1465 (1998).
28. H.T. Allawi and J. SantaLucia Jr., "*Nearest-neighbor thermodynamic parameters for internal G.A mismatches in DNA*," Biochemistry, **37**, pp. 2170-2179 (1998).
29. H.T. Allawi and J. SantaLucia Jr., "*Nearest-neighbor thermodynamics of internal A.C mismatches in DNA: Sequence dependence and pH effects*," Biochemistry, **37**, pp. 9435-9444 (1998).

30. N. Peyret, P.A. Seneviratne, H.T. Allawi and J. SantaLucia Jr., "*Nearest-neighbor thermodynamics and NMR of DNA sequences with internal A.A, C.C, G.G, and T.T mismatches,*" *Biochemistry*, **38**, pp. 3468-3477 (1999).
31. J. Zhu and R.M. Wartell, "*The effect of base sequence on the stability of RNA and DNA single base bulges,*" *Biochemistry*, **38**, pp. 15986-15993 (1999).
32. X. Guan and E.C. Uberbacher, "*Alignments of DNA and protein sequences containing frameshift errors,*" *Comput. Applic.Biosci.* **12**, pp. 31-40 (1996).
33. N.P. Brown, C. Sander and P. Bork, "*Frame: detection of genomic sequencing errors,*" *Bioinformatics*, **4**, pp. 367-371 (1998).
34. J. Posfai and R.J. Roberts, "*Finding errors in DNA sequences,*" *Proc Natl Acad Sci U S A*, **89**, pp. 4698-4702 (1992).
35. D.J. States and D. Botstein, "*Molecular sequence accuracy and the analysis of protein coding regions,*" *Proc Natl Acad Sci U S A*, **88**, pp. 5518-5522 (1991).
36. M. Claverie, "*Detecting frame shifts by amino acid sequence comparison,*" *J. Mol. Biol.* **234**, pp. 1140-1157 (1993).
37. X. Huang and J. Zhang, "*Methods for comparing a DNA sequence with a protein sequence,*" *Comput. Applic. Biosci.* **12**, pp. 497-506 (1996).
38. E. Birney, J.D. Thompson and T.J. Gibson, "*Pairwise and Searchwise: finding the optimal alignment in a simultaneous comparison of a protein profile against all DNA translation frames,*" *Nucleic Acids Res.* **24**, pp. 2730-2739 (1996).
39. Z. Zhang, W.R. Pearson and W. Miller, "*Aligning a DNA sequence with a protein sequence,*" *J. Comput. Biol.* **4**, pp. 339-349 (1997).
40. I. Barman, I. Roy and B. Lustig, "*Evaluating an alternative approach to nucleic acid sequence alignment using hybridization strategies,*" (Sixteenth Annual California State University Biotechnology Symposium, 65, San Jose, CA, 2004).

41. Entelechon (2005). A back-translational tool site [Online]. Available at <http://www.entelechon.com/> (accessed June 2005). WWW article.
42. P. Bevington and D.K. Robinson, Data Reduction and Error Analysis for the Physical Sciences, 3rd ed. (McGraw Hill, U S A, 2003) pp. 98-114.
43. S. Miyazawa and R.L. Jernigan, "*A new substitution matrix for protein sequence searches based on contact frequencies in protein structures*," Protein Eng. **6**, pp. 267-278 (1993).
44. J. Wang, F. Yang, J. Nair, and B. Lustig, "*Evaluation of an Energy-Based Smith-Waterman Algorithm for Sequence Alignment in a Parallelized Computation Setting*," (Seventeenth Annual California State University Biotechnology Symposium, 117, San Jose, CA, 2004).
45. J. SantaLucia Jr. and D. Hicks, "*The thermodynamics of DNA structural motifs*," Annu. Rev. Biophys. Biomol. Struct. **33**, pp. 415-440 (2004).

APPENDIX A

Table showing the 96 proteins and the 11 ncRNA sequences.

ID_Accession	Protein Source	Common Name
1a3c_Z99112	B. subtilis	pyrr
1a3s_AJ002385	H. sapien	ubc9
1a5z_X74302	T. maritima	lactate dehydrogenase
1a6q_S87759	H. sapien	phosphatase 2C
1aat_X15636	G. gallus	cytosolic aspartate aminotransferase
1ab4_X06373	E. coli	type II DNA topoisomerase
1abm_X07834	H. sapien	manganese superoxide dismutase
1add_NM_007398	M. musculus	adenosine deaminase
1adi_J04199	E. coli	adenylosuccinate synthetase
1ae4_U46064	S. scrofa	aldehyde reductase
1af3_U72350	Rattus	apoptosis regulator Bcl-Xl
1agm_K02465_G1	A. awamorii	α acarbose
1ahn_M59426	E. coli	flavodoxin
1ai2_J02799	E. coli	dehydrogenase
1ak2_M16225	B. taurus	adenylate kinase isoenzyme-2
1ako_X13002	E. coli	exonuclease III
1al8_J03492	S. oleracea	glycolate oxidase
1aln_M60916	E. coli	cytidine deaminase
1amn_X56517exon2	T. californica	acetylcholinesterase
1amp_M85159	A. proteolytica	aminopeptidase
1an7_X79551	T. thermophilus	ribosomal protein S8
1an9_M16972	S. scrofa	D-amino acid oxidase
1ao5_NM_010115	M. musculus	prorenin converting enzyme
1aob_J01710	E. coli	thymidylate synthase
1aqh_X58627	A. haloplanctis	alpha-amylase
1lz1_NM_000239	H. sapien	lysozyme
1omd_J02705	Rattus	oncomodulin
1rbp_NM_006744	H. sapien	retinal binding protein
1rhd_M58561	B. taurus	rhodanese
1sgt_M64471	S. griseus	trypsin
2act_AF343446	A. chinensis	actinidin
2cab_M33987	H. sapien	carbonic anhydrase form B

ID_Accession	Protein Source	Common Name
2cdv_D31702	D. vulgaris	cytochrome C3
2ci2_X57035	H. vulgare	chymotrypsin inhibitor CI-2
2cpp_M12546	P. putida	cytochrome P450 Cam
2cro_V00635	bacteriophage 434	434 Cro protein
2cts_M21197	S. scrofa	citrate synthase
2cyp_J01468	S. cerevisiae	cytochrome c peroxidase
2gn5_V00604	filamentous bacteriophage	gene 5 DNA binding protein
2hmg_AJ427297	influenza virus	hemagglutinin
2lbp_M29378	E. coli	leucine binding protein
2ldx_L10389	M. musculus	apo-lactate dehydrogenase
2ltm_X66368	P. sativum	pea lectin
2lzt_V00428	G. gallus	lysozyme
2mev_M88547	mengo virus	mengo encephalomyocarditis virus coat protein
2pab_U19780	H. sapien	pre albumin
2plv_V01149	H. sapien	poliovirus
2prk_X14689	T. album	proteinase K
2rn2_V00337	E. coli	ribonuclease H
2rsp_V01197	rous sarcoma virus	rouse sarcoma virus protease
2sga_M17103	S. griseus	proteinase A
2sni_K02496	B. amyloliquefaciens	subtilisin novo complex w/CI-2
2ssi_M54887	S. albogriseolus	streptomyces subtilisin inhibitor
2stv_L06057	tobacco necrosis virus	satellite tobacco necrosis virus
2taa_M33218	A. oryzae	taka-amylase A
2tbv_AJ249740	tomato bushy stunt virus	structure of tomato bushy stunt virus
2tim_X03921	T. brucei brucei	triosephosphate isomerase
2trx_M26133	E. coli	thioredoxin
2tsc_J01710	E. coli	thymidilate synthase
2tsl_J01546	B. stearothermophilus	Tyrosyl-Transfer RNA Synthetase
2wrp_J01715	E. coli	Trp repressor
3app_AF295432	P. janthinellum	acid proteinase
3blm_U58139	S. aureus	β -lactamase
3cla_X53796	E. coli	type III chloramphenicol acetyltransferase
3cna_AF308777	C. ensiformis	concanavalin A
3est_X04036	S. scrofa	native elastase
3gpd_J02642	H. sapien	D-glyceraldehyde-3-phosphate dehydrogenase
3grs_NM_000637	H. sapien	glutathione reductase

ID_Accession	Protein Source	Common Name
3icb_M18344	B. taurus	calcium binding protein
3pfk_D13095	B. stearothermophilus	phosphofructokinase:
3pgk_K00553	S. cerevisiae	phosphoglycerate kinase
3pgm_X58789	S. cerevisiae	phosphoglycerate mutase
3psg_J04601	S. scrofa	pepsinogen
3rnt_D28341	A. oryzae	Lys 25-RNase T ₁
3rp2_J02712	Rattus	rat mast cell protease II
3sgb_M17104	S. griseus	proteinase
3tln_X76986	B. thermoproteolyticus	thermolysin
4ape_X63351	C. parasitica	endothiapepsin
4dfr_J01609	E. coli	dihydrofolate reductase
4fxn_J05015	Clostridium beijerinckii	flavodoxin
4mdh_U44846	S. scrofa	cytoplasmic malate dehydrogenase
4pep_M20920	S. scrofa	pepsin
4rhv_K02121	human rhinovirus 14	rhinovirus 14
4sbv_L34672	southern bean mosaic virus	southern bean mosaic virus coat protein
4tnc_M19027	G. gallus	troponin C
4xia_M19027	Arthrobacter sp.	D-xylose isomerase
5acn_Z49931	S. scrofa	aconitase
5cpv_AJ292212	C. carpio	calcium-binding barvalbumin B
5rsa_S81740	B. taurus	ribonulcease A
5rub_X00286	R. rubrum	rubisco
6ldh_U38893	S. acanthias	M4 apo-lactate dehydrogenase
7api_X01683	H. sapien	modified α_1 -antitrypsin
8adh_M64864	E. caballus	apo-liver alcohol dehydrogenase
8atc_K01472	E. coli	aspartate, carbamoyl transferase
9pap_M15203	C. papaya	papain cys-25 oxidized
9wga_M25536	T. aestivum	isolectin 2

ID_Accession	ncRNA Source	Common Name
DicF_X07465	E. coli	Division inhibitor
micF_X04888	E. coli	Regulatory antisense RNA
UptR_AF272839	E. Coli	Small RNA
HIF_U85044	H. Sapien	Hypoxia inducible factor
PCGEM1_AF223389	H. Sapien	Non-coding mRNA
SRA1_AF092038	H. Sapien	Steroid receptor RNA activator

ID_Accession	ncRNA Source	Common Name
Igf2_AB030734	M. Musculus	Peg8/Igf2as mRNA, imprinting gene
Msx1_AF267728	M. Musculus	endogenous antisense RNA
St_RprA_AE008759	S. typhimurium	Regulatory RNA
St_DsrA_AE008788	S. typhimurium	Small RNA
CsrB_AE008835	S. typhimurium	Regulatory RNA

APPENDIX B

Table showing the 96 proteins and 11 non-coding RNA with their corresponding number of alignments, number of query base pairs, slope, and R-values.

Proteins ID_Accession Number	Number of alignments	Number of query base pairs	HE vs. BLASTN score slope / R	HE vs. Alignment length slope / R
la3c_Z99112	20	546	-0.911 / -0.995	-1.805/-0.988
la3s_AJ002385	269	477	-1.005 / -0.975	-1.565/-0.962
la5z_X74302	41	960	-0.958 / -0.999	-1.901 / 0.999
la6q_S87759	259	1149	-0.949 / -0.99	-1.693 / -0.988
laat_X15636	271	1239	-1.06 / -0.956	-1.644 / -0.95
lab4_X06373	931	2628	-1.06 / -0.982	-1.703 / -0.98
labm_X07834	595	669	-1.048 / -0.984	-1.691 / -0.98
ladd_NM_007398	143	1059	-1.195 / -0.912	-1.43 / -0.966
ladi_J04199	317	1299	-0.994 / -0.979	-1.788 / -0.974
lae4_U46064	173	978	-1.352 / 0.973	-1.53 / -0.986
laf3_U72350	163	702	-1.111 / -0.991	-1.821 / -0.992
lagm_K02465_G1	92	1923	-1.098 / -0.995	-1.92 / -0.994
lahn_M59426	78	531	-0.991 / -0.973	-1.627 / -0.969
lai2_J02799	633	1251	-1 / -0.99	-1.771 / -0.986
lak2_M16225	206	705	-1.204 / -0.984	-1.59 / -0.989
lako_X13002	79	807	-1.009 / -0.994	-1.87 / -0.987
lal8_J03492	144	1110	-1.426 / -0.894	-1.158 / -0.978
laln_M60916	49	885	-1.016 / -0.991	-1.903 / -0.988
lamn_X56517exon2	35	167	-1.01 / -0.998	-1.909 / -0.997
lamp_M85159	50	1299	-1.014 / -0.996	-1.923 / -0.996
lan7_X79551	20	417	-1.182 / -0.995	-2.134 / -0.992
lan9_M16972	89	1044	-1.212 / -0.938	-1.497 / -0.974
lao5_NM_010115	1029	786	-1.142 / -0.972	-1.618 / 0.982
laob_J01710	134	795	-1.004 / -0.981	-1.69 / -0.975
laqh_X58627	31	2010	-0.915 / -0.99	-1.7 / -0.99
llz1_NM_000239	306	447	-0.979 / -0.95	-1.534 / -0.963
lomd_J02705	65	330	-1.127 / -0.969	-1.538 / -0.973
lrpb_NM_006744	430	606	-1.205 / -0.969	-1.678 / -0.978
lrhd_M58561	176	894	-1.406 / -0.985	-1.645 / -0.992
lsqt_M64471	510	1059	-1.211 / -0.989	-2.1 / 0.989
2act_AF343446	65	1143	-0.983 / -0.99	-1.819 / -0.99
2cab_M33987	229	786	-1.04 / -0.965	-1.503 / -0.974
2cdv_D31702	22	393	-1.173 / -0.979	-1.771 / -0.981
2ci2_X57035	72	255	-1.057 / -0.995	-1.907 / -0.995
2cpp_M12546	12	1248	-1.056 / -0.99	-2.096 / -1
2cro_V00635	21	216	-0.927 / -0.99	-1.846 / -0.99
2cts_M21197	188	1395	-1.182 / -0.973	-1.533 / -0.985
2cyp_J01468	48	1089	-0.96 / -0.99	-1.846 / -0.99
2gn5_V00604	60	264	-0.936 / -0.99	-1.766 / -0.99

Proteins ID_Accession Number	Number of alignments	Number of query base pairs	HE vs. BLASTN score slope / R	HE vs. Alignment length slope / R
2hmg_AJ427297	1117	1701	-1.057 / -0.946	-1.285 / -0.968
2lbp_M29378	84	1107	-1.044 / -0.962	-1.637 / -0.964
2ldx_L10389	655	999	-1.018 / -0.922	-1.401 / -0.953
2ltm_X66368	151	828	-0.987 / -0.99	-1.543 / -0.99
2lzt_V00428	126	444	-1.028 / -0.99	-2.072 / -0.99
2mev_M88547	146	2703	-1.124 / -0.887	-1.174 / -0.953
2pab_U19780	175	443	-1.053 / -0.938	-1.611 / -0.957
2plv_V01149	1608	6630	-1.026 / -0.974	-1.494 / -0.974
2prk_X14689	252	1155	-1.1 / -0.984	-1.884 / -0.986
2rn2_V00337	100	757	-0.965 / -0.99	-1.773 / -0.976
2rsp_V01197	466	2106	-1.076 / -0.99	-1.936 / -0.99
2sga_M17103	74	894	-1.157 / -0.974	-2.052 / -0.978
2sni_K02496	75	1149	-1.003 / -0.954	-1.518 / -0.959
2ssi_M54887	43	435	-1.168 / -0.978	-2.219 / -0.986
2stv_L06057	8	1239	-0.962 / -0.99	-1.871 / -0.99
2taa_M33218	254	1500	-1.013 / -0.99	-1.979 / -0.99
2tbv_AJ249740	29	1167	-0.993 / -0.979	-1.629 / -0.973
2tim_X03921	275	753	-1.003 / -0.992	-1.925 / -0.986
2trx_M26133	82	330	-0.963 / -0.99	-1.831 / -0.984
2tsl_J01546	61	1260	-1.03 / -0.99	-1.924 / -0.99
2tsc_J01710	84	795	-0.976 / -0.983	-1.749 / -0.974
2wrp_J01715	30	327	-0.953 / -0.983	-1.931 / -0.977
3app_AF295432	40	1185	-0.994 / -0.99	-2.015 / -0.99
3blm_U58139	260	846	-0.787 / -0.99	-1.427 / -0.99
3cla_X53796	282	642	-0.968 / -0.995	-1.578 / -0.989
3cna_AF308777	67	873	-0.925 / -0.99	-1.788 / -0.99
3est_X04036	683	801	-1.362 / -0.975	-1.588 / -0.989
3gpd_J02642	1709	1008	-1.064 / -0.965	-1.711 / -0.965
3grs_NM_000637	131	1569	-1.014 / -0.983	-1.809 / -0.976
3icb_M18344	57	240	-1.045 / -0.933	-1.221 / -0.952
3pfb_D13095	53	960	-1.029 / -0.993	-1.975 / -0.985
3pgk_K00553	65	624	-0.928 / -0.987	-1.749 / -0.985
3pgm_X58789	226	744	-0.97 / -0.985	-1.624 / -0.987
3psg_J04601	595	1158	-1.369 / -0.971	-1.585 / -0.987
3rnt_D28341	18	393	-0.825 / -0.974	-1.647 / -0.987
3rp2_J02712	255	744	-1.294 / -0.943	-1.328 / -0.984
3sgb_M17104	164	900	-1.207 / -0.943	-1.832 / -0.966
3tln_X76986	112	1647	-0.911 / -0.99	-1.709 / -0.99
4ape_X63351	31	1260	-1.043 / -0.99	-2.17 / -0.99
4dfr_J01609	32	480	-0.985 / -0.975	-1.747 / -0.964
4fxn_J05015	6	417	-0.955 / -1	-1.893 / -1
4mdh_U44846	237	1005	-1.116 / -0.967	-1.431 / -0.979
4pep_M20920	594	1161	-1.368 / -0.968	-1.572 / -0.986
4rhv_K02121	1039	6540	-0.889 / -0.99	-1.688 / -0.99
4sbv_L34672	16	786	-1.027 / -0.99	-1.827 / -0.99

Proteins ID_Accession Number	Number of alignments	Number of query base pairs	HE vs. BLASTN score slope / R	HE vs. Alignment length slope / R
4tnc_M19027	1035	492	-1.372 / -0.977	-1.588 / -0.99
4xia_X59466	91	1188	-1.086 / -0.99	-2.091 / -0.99
5acn_Z49931	546	2343	-1.278 / -0.989	-1.666 / -0.99
5cpv_AJ292212	1054	330	-2.34 / -0.862	-1.132 / -0.99
5rsa_S81740	180	375	-1.104 / -0.856	-1.362 / -0.951
5rub_X00286	639	1401	-1.164 / -0.943	-1.745 / -0.961
6ldh_U38893	882	1002	-1.338 / -0.752	-1.093 / -0.969
7api_X01683	586	1257	-0.995 / -0.989	-1.9 / -0.989
8adh_M64864	602	1128	-1.389 / -0.974	-1.349 / -0.99
8atc_K01472	53	936	-1.022 / -0.979	-1.748 / -0.976
9pap_M15203	104	1038	-0.891 / -0.976	-1.408 / -0.969
9wga_M25536	309	639	-1.315 / -0.955	-1.796 / -0.983

Non coding RNA_Accession Number	Number of alignments	Number of query base pairs	HE vs. BLASTN score slope / R	HE vs. Alignment length slope / R
DicF_X07465	19	53	-1.038 / -0.99	-2.011 / -0.99
micF_X04888	66	129	-1.795 / -0.99	-1.558 / -0.989
UptR_AF272839	68	195	-0.925 / -0.99	-1.616 / -0.99
HIF_U85044	179	1194	-0.829 / -0.99	-1.428 / -0.99
PCGEM1_AF223389	78	1603	-0.838 / -0.99	-1.631 / -0.99
SRA1_AF092038	132	875	-1.041 / -0.99	-1.999 / -0.989
Igf2_AB030734	1512	4773	-1.055 / -0.987	-1.839 / -0.989
Msx1_AF267728	1083	2184	-1.091 / -0.985	-1.829 / -0.988
DsrA_AE008788	27	84	-0.956 / -0.986	-1.711 / -0.983
RprA_AE008759	46	107	-0.958 / -0.99	-1.729 / -0.99
CsrB_NC_003197	42	355	-1.084 / -0.959	-1.626 / -0.973

APPENDIX C

Hybridization Energy (HE) versus BLASTN Score

Correlation plots of 96 Proteins and 11 non coding RNA (nc RNA).

APPENDIX D

Hybridization Energy (HE) versus Alignment Length

Correlation plots of all 96 Proteins and 11 non-coding RNA (nc RNA). The legend shown below for Protein 1a3s is consistent for all the plots in this section.

APPENDIX E

Hybridization Energy (HE) versus BLASTN Score with $E > 0.01$

Correlation plot for the subset of protein-based and ncRNA alignments indicating significant sequence divergence (with Expect values > 0.01)

APPENDIX F

Equations to determine uncertainties

The straight line equation can be written as

$$y = a + bx$$

where:

a: intercept of the straight line

b: slope of the line

The following formulae were used to calculate the estimates of slopes (b) and intercepts

(a) for the aggregate plots that in turn were used to calculate the uncertainties i.e.

standard deviations of the above mentioned parameters [35].

$$\begin{aligned} a &= (1/\Delta') \times (\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i) \\ b &= (1/\Delta') \times (N \sum x_i y_i - \sum x_i \sum y_i) \end{aligned} \quad (6.13)$$

$$\text{where } \Delta' = N \sum x_i^2 - (\sum x_i)^2$$

Assumption: The uncertainties are equal (i.e., $\sigma = \sigma_i$)

$$\sigma^2 = (1/(N-2)) \sum (y_i - a - bx_i)^2 \quad (6.15)$$

$$\sigma_a^2 = (\sigma^2/\Delta') \sum x_i^2$$

$$\sigma_b^2 = N\sigma^2/\Delta' \quad (6.23)$$

APPENDIX G

Perl script used "ExtractBlastnData.pl"

```
#!/usr/bin/perl

#####
#
#   File:   ExtractBlastnData.pl
#   Authors: Frank Yang and Jaya Nair
#   Program to extract blastn data from htm files from BLASTN output and
#   perform shifts
#
#   Version: 1.00
#
#####

# Assign variable to collect all the files with ".htm" extension in the given
# data directory
@filelist1 = glob("*.htm");

# Assign variable to open an options file called "options.txt" that
# contains the options to interpret the blastn data
# The options include - shift left, shift right and the shift amount

$option_file = "options.txt";
$LeftShift = "No";
$RightShift = "No";
$ShiftAmount = 0;

# Open the options file and read the options from the file

open(IN_FILE, $option_file);
@optionsData=<IN_FILE>;
close(IN_FILE);

print "reading options from file...\n";
foreach $LINE_VAR (@optionsData)
{
    ($keyWord,$Value)=split(/ /,$LINE_VAR);
    if($keyWord eq "shift"){
        chomp($Value);
        if($Value eq "left"){
            $LeftShift = "Yes";
        }
        elsif($Value eq "right"){
            $RightShift = "Yes";
        }
        print "Left shift: $LeftShift, Right shift: $RightShift\n";
    }
    if($keyWord eq "amount"){
        chomp($Value);
        $ShiftAmount = $Value;
        print "Shift Amount: $ShiftAmount\n";
    }
}
}
```

```

# Now, operate on each file with a .htm extension in the file list

for ($i = 0; $i <= $#filelist1; $i++) {
    print "\n\n\n==>working on $filelist1[$i]..\n";

    open (INFILE, "<$filelist1[$i]");
    $newfile = $filelist1[$i];
    $newfile =~ s/\.htm/\.data/g;
    $infile = "";
    print "reading file in..\n";
    $found = "NO";
    $infile = "";
    # Look for a particular syntax in the htm file that signifies that this
    # is a valid blastn data file
    while (<INFILE>) {
        if ($_ =~ /<CENTER><b><FONT
            color=\\"green\\">Alignments<\/FONT><\/b><\/CENTER>/) {
            $found = "YES";
        }

        if ($found eq "YES") {
            $infile = $infile . $_;
        }
    }
    close (INFILE);

    # Create output file name that has the same name as the input
    # file name with a ".data" file extension
    # Add the "GI|heading..." column names to the file
    open (OUTFILE, ">$newfile");
    print OUTFILE "GI|heading|score|raw score|expect|identity count|identity
percent|gap count|gap percent|query sequence|subject sequence|query
start|subject start|alignment\n";

    $infile =~ s/\n\n\n/g;
    $infile =~ s/\n/####/g;

    (@templist) = split /<\/PRE>/, $infile;

    $heading = "";
    $identity = "";
    $score = "";
    $sequence1 = "";
    $sequence2 = "";
    $genbank = "";

    # Parse the data string and extract the heading, identity, score # and
    sequences

    for ($xi = 0; $xi < $#templist; $xi++) {
        $alignment = "";
        $seq1strand = "";
        $seq2strand = "";
        $temp3 = "";
        $temp4 = "";
        $gotstart = "No";

        #if the file name has a syntax that includes a string "a
        # name" then use this section to parse the input file

        if ($templist[$xi] =~ /a name =/) {

```

```

($genbank,$templist[$xi]) = split
/entrez\query.fcgi\?cmd/, $templist[$xi];
($junk, $genbank) = split /= /, $genbank;
($genbank, $junk) = split />/, $genbank;

($heading,$templist[$xi]) = split /Score = /, $templist[$xi];
(@junks) = split /<\a>/, $heading;
$heading = $junks[$#junks];
$heading =~ s/####/ /g;
$heading =~ s/ / /g;
$heading =~ s/ / /g;
$heading =~ s/ / /g;
$heading =~ s/ / /g;
($heading, $junk) = split /Length/, $heading;
if ($heading =~ /^ /){
    $heading = substr $heading,1;
    $heading =~ s/ //g;
    print "IF Heading: $heading\n";
    ($score, $templist[$xi]) = split /Expect = /,
    $templist[$xi];
    ($score, $rawscore) = split / bits/, $score;
    $score =~ s/ //g;
    $rawscore =~ s/ //g;
    $rawscore =~ s/(//g;
    $rawscore =~ s/(//g;
    $rawscore =~ s/(//g;

($expect, $templist[$xi]) = split /Identities = /, $templist[$xi];
$expect =~ s/ //g;
$expect =~ s/####//g;
print "IF Expect: $expect\n";
$gappercent=""; $gapcount="";
($identity, $templis[$xi]) = split /Strand = /, $templist[$xi];
($identitycount, $identitypercent,
$junk,$junk,$gapcount,$gappercent) = split / /, $identity;
#$identitycount =~ s/(// /g;
$identitypercent =~ s/(//g;
($identitypercent, $junk) = split /(\\/, $identitypercent;
($gappercent, $junk) = split /(\\/, $gappercent;
#$gapcount =~ s/(// /g;
$gappercent =~ s/(//g;

if ($templist[$xi] =~ /Plus \/) {
    $seq1strand = "+";
}
else {
    $seq1strand = "-";
}

if ($templist[$xi] =~ /(\\ Minus/) {
    $seq2strand = "-";
}
else {
    $seq2strand = "+";
}

$templist[$xi] =~ s/|//g;
$templist[$xi] =~ s/ / /g;
$templist[$xi] =~ s/ / /g;

```

```

$templist[$xi] =~ s/ / /g;
(@tempA) = split /Query: /, $templist[$xi];

# Do the shifting (left/right) of the sequences as described in
# the options.txt file
for ($j = 0; $j <= $#tempA; $j++) {
    if ($tempA[$j] =~ /Sbjct: /) {
        $tempA[$j] =~ s/#//g;
        $tempA[$j] =~ s/ / /g;
        ($temp1,$temp2) = split /Sbjct: /, $tempA[$j];
        if ($gotstart eq "No") {
            ($temp3,$junk3) = split / /,$temp1;
            ($temp4,$junk4) = split / /,$temp2;
            $gotstart = "Yes";
        }
        ($tempf,$junks,$temps) = split / /,$temp2;
        if($LeftShift eq "Yes"){
            $subjectLen = length($junks);
            $shftSbjct = substr $junks, $ShiftAmount, $subjectLen
            - $ShiftAmount;
            for ($sf = 0; $sf < $ShiftAmount; $sf++) {
                $shftSbjct = $shftSbjct."-";
            }
            $subjectLen2 = length($junks);
            $temp2 = $tempf.$shftSbjct.$temps;
        }
        elsif ($RightShift eq "Yes") {
            $subjectLen = length($junks);
            $shftSbjct = substr $junks, 0, $subjectLen - $ShiftAmount;
            for ($sf = 0; $sf < $ShiftAmount; $sf++) {
                $shftSbjct = "-".$shftSbjct;
            }
            $subjectLen2 = length($junks);
            $temp2 = $tempf.$shftSbjct.$temps;
        }
        else{
            $temp2 =~ s/[| ]//g;
        }
        $temp1 =~ s/[| ]//g;

        $sequence1 = $sequence1 . $temp1;
        $sequence2 = $sequence2 . $temp2;
    }
}
$seq1strand = $temp3 . " $seq1strand";
$seq2strand = $temp4 . " $seq2strand";
$sequence1 =~ s/[0-9]//g;
$sequence2 =~ s/[0-9]//g;
$seqlength = length($sequence1);
# Go through the query and subject sequence data and create the
# alignment sequence. If the data is the same, assign the
# alignment as "+". If there is a "-" in the query sequence,
# assign "Q", if there is a "-" in the subject sequence, assign
# "S". Else, assign a "-".
for ($sl = 0; $sl < $seqlength; $sl++) {
    $temp5 = substr $sequence1, $sl, 1;
    $temp6 = substr $sequence2, $sl, 1;

    if ($temp5 eq $temp6) {
        $alignment = $alignment . "+";
    }
}

```

```

    }
    elsif ($temp5 eq "-") {
        $alignment = $alignment . "Q";
    }
    elsif ($temp6 eq "-") {
        $alignment = $alignment . "S";
    }
    else {
        $alignment = $alignment . "-";
    }
}
$sequence1 =~ s/\-//g;
$sequence2 =~ s/\-//g;
$gappercent =~ s/\(//g;

#Print out the generated result into the output file

print "\n";
print "    GI#: $genbank\n";
print "  heading: $heading\n";
print "    score: $score\n";
print "  expect: $expect\n";
print "%identity: $identitypercent\n";
print "#identity: $identitycount\n";
print "    seq1: $sequence1\n";
print "    seq2: $sequence2\n";
print "seq1 strt: $seq1strand\n";
print "seq2 strt: $seq2strand\n";
print "alignment: $alignment\n";
}
# Else, use this method to parse the input file data
else {
    $tt = $templist[$xi];
    print "ELSE Heading??: $heading, Element1: $templist[$xi]\n";
    ($junk,$templist[$xi]) = split /Score = /, $templist[$xi];
    ($score, $templist[$xi]) = split /Expect = /, $templist[$xi];
    ($score, $rawscore) = split / bits/, $score;
    $score =~ s/ //g;
    $rawscore =~ s/ //g;
    $rawscore =~ s/\(//g;
    $rawscore =~ s/\)//g;
    $rawscore =~ s/\,//g;
    ($expect, $templist[$xi]) = split /Identities = /, $templist[$xi];
    $expect =~ s/ //g;
    $expect =~ s/#####//g;
    print "ELSE Expect: $expect\n";
    $gappercent=""; $gapcount="";
    ($identity, $templis[$xi]) = split /Strand = /, $templist[$xi];
    ($identitycount, $identitypercent,
$junk,$junk,$gapcount,$gappercent) = split / /, $identity;
    #$identitycount =~ s/\\ // /g;
    $identitypercent =~ s/\(//g;
    ($identitypercent, $junk) = split /\)/, $identitypercent;
    ($gappercent, $junk) = split /\)/, $gappercent;
    #$gapcount =~ s/\\ // /g;
    $gappercent =~ s/\(//g;
    if ($templist[$xi] =~ /^Plus/) {
        $seq1strand = "+";
    }
    else {

```

```

        $seq1strand = "-";
    }
    if ($templist[$xi] =~ /\// Minus/) {
        $seq2strand = "-";
    }
    else {
        $seq2strand = "+";
    }
    $templist[$xi] =~ s/|//g;
    $templist[$xi] =~ s/ / /g;
    $templist[$xi] =~ s/ / /g;
    $templist[$xi] =~ s/ / /g;
    (@tempA) = split /Query: /, $templist[$xi];
    # Create the query and subject sequences as outlined in the
    # section above
    for ($j = 0; $j <= $#tempA; $j++) {
        if ($tempA[$j] =~ /Sbjct: /) {
            $tempA[$j] =~ s/#//g;
            $tempA[$j] =~ s/ / /g;
            ($temp1,$temp2) = split /Sbjct: /, $tempA[$j];
            if ($gotstart eq "No") {
                ($temp3,$junk) = split / /,$temp1;
                ($temp4,$junk) = split / /,$temp2;
                $gotstart = "Yes";
            }
            ($tempf,$junks,$temps) = split / /,$temp2;
            if($LeftShift eq "Yes"){
                $subjctLen = length($junks);
                $shftSubjct = substr $junks, $ShiftAmount, $subjctLen -
                $ShiftAmount;
                for ($sf = 0; $sf < $ShiftAmount; $sf++) {
                    $shftSubjct = $shftSubjct."-";
                }
                $subjctLen2 = length($junks);
                $temp2 = $tempf.$shftSubjct.$temps;
            }
            elsif ($RightShift eq "Yes") {
                $subjctLen = length($junks);
                $shftSubjct = substr $junks, 0, $subjctLen -
                $ShiftAmount;
                for ($sf = 0; $sf < $ShiftAmount; $sf++) {
                    $shftSubjct = "-".$shftSubjct;
                }
                $subjctLen2 = length($junks);
                $temp2 = $tempf.$shftSubjct.$temps;
            }
            else{
                $temp2 =~ s/[| ]//g;
            }
        }
        $temp1 =~ s/[| ]//g;
        $sequence1 = $sequence1 . $temp1;
        $sequence2 = $sequence2 . $temp2;
    }
    $seq1strand = $temp3 . " $seq1strand";
    $seq2strand = $temp4 . " $seq2strand";

    $sequence1 =~ s/[0-9]//g;
    $sequence2 =~ s/[0-9]//g;

```



```

$seqlength = length($sequence1);
# Go through the query and subject sequence data
# and create the alignment sequence. If the data
# is the same, assign the alignment as "+". If there
# is a "-" in the query sequence, assign "Q", if there
# is a "-" in the subject sequence, assign "S". Else, assign
# a "-".
for ($sl = 0; $sl < $seqlength; $sl++) {
    $temp5 = substr $sequence1, $sl, 1;
    $temp6 = substr $sequence2, $sl, 1;

    5 eq $temp6) {$alignment = $alignment . "+";}
    elseif ($temp5 eq "-") {
        $alignment = $alignment . "Q";
    }
    elseif ($temp6 eq "-") {
        $alignment = $alignment . "S";
    }
    else {
        $alignment = $alignment . "-";
    }
}
$sequence1 =~ s/\-//g;
$sequence2 =~ s/\-//g;
$gappercent =~ s/\(//g;

# Print out the parsed result
print "\n";
print "    GI#: $genbank\n";
print "  heading: $heading\n";
print "    score: $score\n";
print "    expect: $expect\n";
    print "%identity: $identitypercent\n";
    print "#identity: $identitycount\n";
    print "      seq1: $sequence1\n";
    print "      seq2: $sequence2\n";
    print "seq1 strt: $seq1strand\n";
    print "seq2 strt: $seq2strand\n";
    print "alignment: $alignment\n";

}
if ($gapcount eq "") {
    $gapcount = "0/$seqlength";
}
if ($gappercent eq "") {
    gappercent = "0%";
}
$heading =~ s/\/\-\-//g;
print OUTFILE
"$genbank|$heading|$score|$rawscore|$expect|$identitycount|$identi
typercent|$gapcount|$gappercent|$sequence1|$sequence2|$seq1strand|
$seq2strand|$alignment\n";
$score = "";
$identity = "";
$sequence1 = "";
$sequence2 = "";
$expect = "";
$identitycount="";
$identitypercent="";

```

```
        $gapcount="";
        $gappercent="";
        $seq1strand="";
        $seq2strand="";
        $alignment="";
    }
    close (OUTFILE);
    print "==>done!\n";
}
```

APPENDIX H

Perl script used "CalculateBlastnData.pl"

```
#####
#
#   File:  CalculateBlast.pl
#   Author:  Frank Yang
#   Purpose:  Perl script for calculating hybridization energy,
#             alignment length, identity percent, gap percent,
#             mismatch count and gap count of the data extracted by
#             ExtractBlastData.pl
#   Version:  1.00
#
#####

#!/usr/bin/perl

# Create a file list of all the .data files available in the data
# directory
@filelist1 = glob("*.data");

# For each file in the directory, create an output file with the same
# name as the input file

for ($xyz = 0; $xyz <= $#filelist1; $xyz++) {
    print "Working on $filelist1[$xyz]..\n";
    open(INFILE, "<$filelist1[$xyz]");
    $output = <INFILE>;
    chomp($output);
    $output =~ s/\r//g;

# Add a column descriptor to the output file to describe the data in
# the file
    $output = $output . "|Blast Energy|Blast Length|Correct Blast
Identity Percent|Corrected Blast Gap Percent|Blast Mismatch Count|Blast
Gap Count\n";
    while(<INFILE>) {
        $tempstring = $_;
        chomp($tempstring);
        $tempstring =~ s/\r//g;
        (@temparray) = split /\|/, $tempstring;

# Call the restore alignment sub routine that returns the query and the
# subject sequence

($Query,$Subject) =
&RestoreAlignment($temparray[9],$temparray[10],$temparray[13]);
# Calculate the hybridization energy routine using the query and the
# subject sequences
($AsymHE) =
&CalculateAlignmentHybridizationEnergy($Query,$Subject,"subject","yes")
;
    }
}
```

```

$AsymHE = sprintf("%.3f", $AsymHE);
($num,$den) = split /\//, $temparray[5];
$BlastLength = $den;
$CorrectedBlastIdentityPercent = $num/$den * 100;
$CorrectedBlastIdentityPercent = sprintf("%.3f",
$CorrectedBlastIdentityPercent);
($num, $den) = split /\//, $temparray[7];
$CorrectedBlastGapPercent = $num/$den * 100;
$CorrectedBlastGapPercent = sprintf("%.3f", $CorrectedBlastGapPercent);

$MismatchOnly = $temparray[13];
$MismatchOnly =~ s/[+QS]//g;
$MismatchOnly = length($MismatchOnly);
$GapOnly = $temparray[13];
$GapOnly =~ s/[+|-]//g;
$GapOnly = length($GapOnly);
$tempstring = $tempstring .
"| $AsymHE| $BlastLength| $CorrectedBlastIdentityPercent| $CorrectedBlastGa
pPercent| $MismatchOnly| $GapOnly\n";

# Append the data to the output string

$output = $output . $tempstring;
}
close(INFILE);

# Replace the output file name extension from .data to .txt

$outfile = $filelist1[$xyz];
$outfile =~ s/\.data/\.txt/g;
open(OUTFILE, ">$outfile");
print OUTFILE $output;
close(OUTFILE);
}

#####
#
#      Subroutine: CalculateAlignmentHybridizationEnergy
#
#      Inputs: Query Sequence, Subject Sequence,
#      What is complementary (Subject or Query),
#      Account for assymetry (true or false)
#      Output: Hybridization energy value
#
#####

sub CalculateAlignmentHybridizationEnergy {
    my
($querySeq,$subjectSeq,$which2complementary,$account4asymmetry) = @_;
    my $gapCounter = 0;
    my $hybridizationEnergy = 0;
    my $positionCounter = 0;

```

```

my $seqLength = length($querySeq) - 1;
my $queryRNA;
my $subjectRNA;
my $queryPair;
my $subjectPair;
my $testPair;
my $gapEnd;
my $subjectGapLength;
my $queryGapLength;
my $subjectBase;
my $queryBase;
my $gapSize;
my $gapPenalty;

# Depending on the which2complementary flag, convert the query and
# subject sequences from DNA to RNA using the corresponding
# combinations of normal and complementary flags

if ($which2complementary =~ /subject/i) {
    #5'to 3'
    ($queryRNA) = &ConvertDNAToRNA($querySeq, "Normal");
    # 3' to 5'
    ($subjectRNA) = &ConvertDNAToRNA($subjectSeq, "Complementary");
}
else {
    # 3' to 5'
    ($queryRNA) = &ConvertDNAToRNA($querySeq, "Complementary");
    # 5' to 3'
    ($subjectRNA) = &ConvertDNAToRNA($subjectSeq, "Normal");
}

&InitializeEnergyValues;

# Run through the entire length of the sequence and evaluate
# pairs of query and subject sequences
while ($positionCounter < $seqLength) {
    $subjectPair = substr $subjectRNA, $positionCounter, 2;
    $queryPair = substr $queryRNA, $positionCounter, 2;
    # Create a test pair using the which2Complementary flag
    if ($which2complementary =~ /subject/i) {
        $testPair = "5". $queryPair . "3,3" . $subjectPair .
        "5";
    }
    else {
        $testPair = "5". $subjectPair . "3,3" . $queryPair .
        "5";
    }

    # If there is no energy in the test pair, we have a match. In
    # this case, just increment the position counter to evaluate the
    # next pair of the sequence values

    if ($Energy{$testPair} ne "") {
        # match
    }
}

```

```

        $hybridizationEnergy = $hybridizationEnergy +
        $Energy{"$testPair"} ;
        $positionCounter++;
    }
    else {
        # If there is a value in the energy parameter, we have a
        # mismatch or gap
        $gapEnd = "no";
        $subjectGapLength = 0;
        $queryGapLength = 0;
        $gapPenalty = 0;
        $gapSize = 0;
        $subjectBase = substr $subjectPair, 0, 1;
        $queryBase = substr $queryPair, 0, 1;
        # Perform an analysis of the subject and query pairs as long as
        # it is not the end of the gap and the sequence counter does not
        # exceed the length of the sequence

        while ($gapEnd ne "yes" && $seqLength >= $positionCounter) {
            $subjectBase = substr $subjectPair, 1, 1;
            $queryBase = substr $queryPair, 1, 1;

            # If the subject element is A and the query element is U or vice
            # versa OR If the the subject element is G and the query element
            # is C or vice versa, then, we have the end of the gap. Else, if
            # the subject element is NOT "-", increment subject gap length by
            # one. Similarly, if the query gap length is NOT "-", increment
            # query gap length

            if (($subjectBase =~ /A/i && $queryBase =~ /U/i) ||
                ($subjectBase =~ /U/i && $queryBase =~ /A/i) ||
                ($subjectBase =~ /G/i && $queryBase =~ /C/i) ||
                ($subjectBase =~ /C/i && $queryBase =~ /G/i)) {
                $gapEnd = "yes";
                $positionCounter++;
            }
            else {
                if ($subjectBase ne "-") {
                    $subjectGapLength++;
                }
                if ($queryBase ne "-") {
                    $queryGapLength++;
                }
                $positionCounter++;
                $subjectPair = substr $subjectRNA, $positionCounter, 2;
                $queryPair = substr $queryRNA, $positionCounter, 2;
            }
        }
        # If the subject gap length is NOT 0 and the query gap length is
        # NOT 0 and we have not reached the end of the sequence,
        # this is an internal loop condition
        if ($subjectGapLength ne 0 && $queryGapLength ne 0 && $seqLength
            >= $positionCounter) {
            # internal loop

```

```

if ($account4asymmetry =~ /yes/i) {
    my $asymn;
    my $asymm = 5;
    my $asymfm;
    # Compute the difference between the subject and query gap
    # lengths
    $asymn = $subjectGapLength - $queryGapLength;
    if ($asymn < 0) {
        $asymn = 0 - $asymn;
    }

    # Get the lesser of the subject and query gap lengths and
    # let the maximum value be 5
    if ($subjectGapLength < $asymm) {
        $asymm = $subjectGapLength;
    }
    if ($queryGapLength < $asymm) {
        $asymm = $queryGapLength;
    }

    # Derive a value for determining gap penalty depending on
    # the value of the gap length from above

    if ($asymm eq 1) { $asymfm = 0.7}
    elsif ($asymm eq 2) { $asymfm = 0.6}
    elsif ($asymm eq 3) { $asymfm = 0.4}
    elsif ($asymm eq 4) { $asymfm = 0.2}
    elsif ($asymm eq 5) { $asymfm = 0.1}

    # Compute gap penalty as a product of the difference
    # of the gaplengths and the factor from the table
    # above
    $gapPenalty = $asymn * $asymfm;

    # Now, compute the hybridization energy using the gap
    # penalty
    if ($gapPenalty < 6) {
        $hybridizationEnergy = $hybridizationEnergy +
        $gapPenalty;
    }
    else {
        $hybridizationEnergy = $hybridizationEnergy +
        6;
    }
    }
    $gapSize = $subjectGapLength + $queryGapLength;

    # Compute gap penalty using the internal loop penalty
    # sub routine
    $gapPenalty = &InternalLoopPenalty($gapSize);
    $hybridizationEnergy = $hybridizationEnergy +
    $gapPenalty;
}

```

```

# If either the subject or the query length is zero and we
have not reached the end of the sequence, then, we have a
bulge loop condition
elseif ($seqLength >= $positionCounter && ($subjectGapLength
eq 0 || $queryGapLength eq 0)) {
    # bulge loop
    $gapSize = $subjectGapLength + $queryGapLength;
    # Compute the bulge loop penalty using the
    # BulgeLoopPenalty subroutine

    $gapPenalty = &BulgeLoopPenalty($gapSize);
    $hybridizationEnergy = $hybridizationEnergy +
    $gapPenalty;
}
}

# Return the hybridization energy value to the calling routine
return ($hybridizationEnergy);
}

#####
#
# Subroutine: RestoreAlignment
#
# Input: Query, Subject and alignment sequences
# Output: Restored query and subject sequences
#
#####
sub RestoreAlignment {
    my ($querySeq, $subjectSeq, $alignment) = @_ ;
    my $ra;
    my $restoredquery = "";
    my $restoredsubject = "";
    my $currentchr;
    my $subjectcounter = 0;
    my $querycounter = 0;

    # Go through the alignment string, and
    # if there is a "Q", add "-" to the restored query,
    # If there is a "S", add "-" to the restored subject,
    # Else, append the corresponding characters to the restored
    # subject and query sequences

    for ($ra = 0; $ra <= length($alignment); $ra++) {
        $currentchr = substr $alignment,$ra,1;
        if ($currentchr =~ /Q/i) {
            $restoredquery = $restoredquery . "-";
            $currentchr = substr $subjectSeq,$subjectcounter,1;
            $restoredsubject = $restoredsubject . $currentchr;
            $subjectcounter++;
        }
        elseif ($currentchr =~ /S/i) {
            $restoredsubject = $restoredsubject . "-";
            $currentchr = substr $querySeq,$querycounter,1;
            $restoredquery = $restoredquery . $currentchr;

```



```

        $querycounter++;
    }
    else {
        $currentchr = substr $subjectSeq,$subjectcounter,1;
        $restoredsubject = $restoredsubject . $currentchr;
        $currentchr = substr $querySeq,$querycounter,1;
        $restoredquery = $restoredquery . $currentchr;
        $subjectcounter++;
        $querycounter++;
    }
}
return ($restoredquery, $restoredsubject);
}

```

```
#####
#
# SubRoutine: ConvertDNAToRNA
#
# Input: Query sequence and Normal or Equivalent flag
#
# Output: Converted query sequence
#
#####
sub ConvertDNAToRNA {
my ($Seq, $Type) = @_ ;
    $Seq = uc $Seq;

    if ($Type eq "Normal") { # convert to equivalent RNA
        $Seq =~ s/T/U/g;
    }
    else { # convert to equivalent complementary RNA
        $Seq =~ s/A/U/g;
        $Seq =~ s/T/A/g;
        $Seq =~ s/G/X/g;
        $Seq =~ s/C/G/g;
        $Seq =~ s/X/C/g;
    }
    return ($Seq);
}
#####
#
# SubRoutine BulgeLoopPenalty
#
# Compute the bulge loop penalty from the bulge size as per
# the formula in the routine below
#####
sub BulgeLoopPenalty {
    my ($size) = @_ ;
    $BulgeLoopPenalty[$size] = -2.73607 + 5.41783 * log($size +
2.04661);
    return ($BulgeLoopPenalty[$size]);
}
#####
#
# SubRoutine InternalLoopPenalty
#
# Compute the internal loop penalty from the loop size as per
# the formula in the routine below
#####
sub InternalLoopPenalty {
    my ($size) = @_ ;
    $InternalLoopPenalty[$size] = -28.3491 + 9.45283 * log($size +
19.8087);
    return($InternalLoopPenalty[$size]);
}

```

```
#####
# SubRoutine InitializeEnergyValues
# Function to initialize enery values table
#####
sub InitializeEnergyValues {
    undef %Energy;
    $Energy{"5AA3,3UU5"} = -0.9;
    $Energy{"5AC3,3UG5"} = -2.1;
    $Energy{"5AG3,3UC5"} = -1.7;
    $Energy{"5AU3,3UA5"} = -0.9;
    $Energy{"5CA3,3GU5"} = -1.8;
    $Energy{"5CC3,3GG5"} = -2.9;
    $Energy{"5CG3,3GC5"} = -2;
    $Energy{"5CU3,3GA5"} = -1.7;
    $Energy{"5GA3,3CU5"} = -2.3;
    $Energy{"5GC3,3CG5"} = -3.4;
    $Energy{"5GG3,3CC5"} = -2.9;
    $Energy{"5GU3,3CA5"} = -2.1;
    $Energy{"5UA3,3AU5"} = -1.1;
    $Energy{"5UC3,3AG5"} = -2.3;
    $Energy{"5UG3,3AC5"} = -1.8;
    $Energy{"5UU3,3AA5"} = -0.9;
}

```

APPENDIX I

Curve fit

The following expressions were fit separately by Yang [25] and Wang [43] to calculate the Internal loop penalty and Bulge loop penalty depending on the loop size 'x'.

$$\text{Internal Loop Penalty}_{(x)} = -28.3491 + 9.45283 * \ln(x + 19.8087)$$

$$\text{Bulge Loop Penalty}_{(x)} = -2.73607 + 5.41783 * \ln(x + 2.04661)$$

The above expressions were generated by Yang by fitting it to the following model using Sigma Plot

$$y_x = a + b * \ln(x + c)$$

Wang developed the following expressions to calculate the penalties for the gap length 'x' by fitting it to the following model.

$$y_x = \frac{a + bx + cx^2 + dx^3}{1 + ex + fx^2 + gx^3} + h$$

His expressions for Internal Loop Penalty and Bulge Loop Penalty depending on the gap length 'x' are as follows:

$$\text{InternalLoopPenalty}_x = \frac{(1.716E05) + [(-1.463E05) * (2x)] + [(4.28E03) * (2x)^2] + [(-1.631E03) * (2x)^3]}{1.0 + [(-6.408E04) * (2x)] + [(1.45E03) * (2x)^2] + [(-1.697E02) * (2x)^3]} + (-1.445E - 01)$$

$$\text{BulgeLoopPenalty}_x = \frac{(-8.137E00) + [(-5.137E01) * (x)] + [(7.243E00) * (x)^2] + [(3.542E01) * (x)^3]}{1.0 + [(3.933E00) * (x)] + [(3.047E01) * (x)^2] + [(1.839E00) * (x)^3]} + 3.524E00$$

The following plots show a comparison of the two equations to determine the penalty with the experimentally determined values by Turner et al. [9]

The plots below can be used to determine the penalty for loop sizes larger than 30 residues using Yang or Wang's equations.