COMPUTATIONAL STUDIES ON STRUCTURE AND FUNCTIONALITY OF BIOMOLECULAR COMPOUNDS

by

Emre Karakoc
B.Eng., Bilkent University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2007

APPROVAL

Name: Emre Karakoc

Degree: Doctor of Philosophy

Title of thesis: COMPUTATIONAL STUDIES ON STRUCTURE AND FUNCTIONALITY OF BIOMOLECULAR COMPOUNDS

Examining Committee: Dr. Funda Ergun
Chair

Dr. S. Cenk Sahinalp, Senior Supervisor

Dr. Jian Pei, Supervisor

Dr. Peter J. Unrau, SFU Examiner

Dr. Ming Li, External Examiner,
Professor of Department of Computer Science,
University of Waterloo

Date Approved: June 21, 2007

Abstract

This thesis is on applying standard combinatorial optimization methods, dynamic programming and linear programming, to help solve two important problems in computational molecular biology: (1) predicting the secondary structure of RNA molecules and (2) predicting the functionality of small biological compounds.

After 25 years of effort, the RNA secondary structure prediction has proven to be very elusive. Much of the available algorithms are based on total free energy minimization. Yet, despite the numerous attempts to perfect this thermodynamic approach, the end results are far from being practical.

We demonstrate that delocalizing the thermodynamic cost of forming an RNA substructure through *energy density* notion can significantly improve available secondary structure prediction methods. Because the notion of energy density is non-linear, the standard dynamic programming approach had to be updated. This updated algorithm can capture the secondary structure of many non-coding RNAs which have been difficult to approximate with alternative methods.

One key application of RNA structure prediction is in understanding how two or more RNAs interact (e.g. an mRNA and a regulatory RNA). In this thesis we formulate the RNA-RNA interaction prediction problem as a combinatorial optimization problem and show how to solve it again via dynamic programming. Because the complexity of the algorithm to solve the most involved formulation of the problem is very high, we also describe heuristic shortcuts, which, in practice, are highly accurate.

The second set of problems we tackle are related to small chemical molecules, which have key cellular functions. In particular we focus on structural similarity search among small chemical molecules, a standard approach used for in-silico drug

discovery. It is possible to use structural similarity to deduce the bioactivities of new compounds provided that the notion of similarity reflects the bioactivity in question and we have efficient data structures to perform structural similarity search.

This thesis shows how to computationally design the "optimal" weighted Minkowski distance wL_p for maximizing the discrimination between active and inactive compounds with respect to a bioactivity. It also demonstrates how to construct an iterative pruning based data structure for performing "nearest neighbor" search under the weighted L_p distance computed.

keywords: rna secondary structure prediction, energy density, rna-rna joint secondary structure prediction, small chemical compounds, k-nearest neighbor classification.

To my Family

Acknowledgments

It is with great pleasure that I would like to thank those who helped me in my Ph.D. studies.

First and foremost I wish to express my deep gratitude to my supervisor and mentor Dr. S. Cenk Şahinalp. I thank him for his continuous encouragement, confidence, support and for sharing his knowledge and experience. He taught me not only the necessary technical skills to receive a Ph.D. degree but also to be a good researcher. He guided me during my studies and he was always working beside me providing a motivating and enthusiastic research environment.

I am very thankful to my committee members for their insightful comments and advice. I would also like to thank all my co-authors. Dr. Artem Cherkasov was the first to introduce me to biochemistry and I gained invaluable experience by working in his lab. His creativity and insight for finding and solving interesting and exciting problems deserved to be mentioned. Dr. Joseph H. Nadeau got us in the world of RNA interactions where countless number of exciting problems wait to be solved. I also like to thank Dr. Kaizhong Zhang, Dr. Jeremy Buhler, H. Alex Ebhardt and Dr. Peter Unrau for their valuable discussions and guidance. I am also indebted to my fellow lab members for both their support in research and their friendship, Dr. Can Alkan, Dr. Gurkan Bebek, Cagri Aksay, Gozde Cozen, Fereydoun Hormozdiari , Phuong Dao, Rahale Salari, Iman Hajirasouliha and countless many others.

I would like to thank to Dr. Uğur Doğrusöz in Bilkent University for introducing me to the computational biology and encouraging me to continue my career in this exciting area.

Last, but not least, I would like to thank my parents and my brother whom I am most privileged to have. Their unconditional and continual support and trust was my driving force all those years. I hope I will make them proud of my achievements as I am proud of them. Nothing I did would be possible without them.

Emre Karakoc

Glossary

bp:	base pair.
C. Elegans:	*Caenorhabditis Elegans*; soil nematode.
codon:	A sequence of three adjacent nucleotides constituting the genetic code that determines the insertion of a specific amino acid during protein synthesis or the signal to stop protein synthesis.
DNA:	Deoxyribonucleic acid; a nucleic acid which is capable of carrying genetic instructions for the biological development of all cellular forms of life and many viruses.
D. Melanogaster:	*Drosophila melanogaster*; fruit fly.
E.Coli:	*Escherichia Coli*; one of the main species of bacteria that live in the lower intestines of warm-blooded animals.
genome:	the whole hereditary information of an organism that is encoded in the DNA (or, for some viruses, RNA).
gRNA:	Guide RNA; RNA that guides the insertion of uridines (RNA editing) into mRNAs.
miRNA:	Micro RNA; a form of single-stranded RNA which is typically 20-25 nucleotides long, and is thought to regulate the expression of other genes.
mRNA:	Messenger RNA; RNA that carries information from DNA to the ribosome sites of protein synthesis in the cell.
nucleotide:	A monomer or the structural unit of nucleotide chains forming nucleic acids such as RNA and DNA.
plasmid:	typically circular double-stranded DNA molecules that are separate from the chromosomal DNA. They usually occur in bacteria, sometimes in eukaryotic organisms.

RNA:	Ribonucleic acid; a nucleic acid consisting of a string of covalently-bound nucleotides.
rRNA:	Ribosomal RNA; the primary constituent of ribosomes.
siRNA:	Small interfering RNA; a class of 20-25 nucleotide-long RNA molecules that interfere with the expression of genes.
snoRNA:	Small nucleolar RNA; a class of small RNA molecules that are involved in chemical modifications of ribosomal RNAs (rRNAs) and other RNA gene.
snRNA:	Small nuclear RNA; a class of small RNA molecules that are found within the nucleus of eukaryotic cells. They are involved in a variety of important processes such as RNA splicing.
stRNA:	Small temporal RNA, small RNA duplexes that are instable and degrade quickly.
tRNA:	Transfer RNA; RNA that transfers a specific amino acid to a growing polypeptide chain at the ribosomal site of protein synthesis during translation.
small chemical molecules:	Molecules with molecular weights of 500 or below and contributes 90% of the current drugs.
QSAR:	Quantitative structure-activity relationship; the process by which chemical structure is quantitatively correlated with a well defined process, such as biological activity (bioactivity) or chemical reactivity.
bioactivity:	Beneficial or adverse effects of small chemical molecules, mostly drugs, on living matter.
metabolites:	Any substance produced by metabolism or by a metabolic process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis aims to present computational methods for solving two seemingly unrelated problems: (1) structural prediction of non-coding RNAs and (2) functional prediction of small chemical compounds. Both of these important bioinformatics problems can be formulated as combinatorial optimization problems, which are typically solved through exact or approximate combinatorial algorithms, heuristics or machine learning tools. In this thesis our goal is to develop exact algorithms for solving these combinatorial optimization problems with (small) polynomial running time and space. Our algorithms, which are variants of two powerful optimization techniques, dynamic programming and linear programming, not only have provable performance and accuracy guarantees but they also work very well in practice.

1.1 Motivation

RNA is a linear polymer with a sugar ribose and phospahate backbone linked together by phospodiester bonds and four different types of nucleotides: Adenine(A), Guanine(G), Cytosine(C) and Uracil(U). Like its cousin DNA, most of RNAs are extensively base paired to form double stranded helices in their natural form. However unlike DNA, this structure is not just limited to long double-stranded helices but rather collections of short helices packed together into substructures. Given an unpaired RNA molecule, most of the bases start to form weak hydrogen bonds with each other and fold into its native secondary structure where each base is either

paired or left empty. In this process the complementary base pairs C-G and A-U can form hydrogen bonds. It is not uncommon, however, to find other types of base pairs in RNA for example G pairing with U (wobble pair) occasionally.

Until early 2000s, RNA was considered to have two functions: (i) transferring genetic information from DNA to protein in the form of messenger RNA (mRNA) - these are the coding RNAs, and (ii) decoding the protein code and combining amino acids together in the form of ribosomal RNA (rRNA) and transfer RNA (tRNA). The discovery of RNA interference (RNAi), the post transcriptional silencing of gene expression via interactions between mRNAs and their "regulatory RNAs" has changed this simple picture of RNA functionality [56, 22]. This revolutionary discovery of RNA based gene regulation by Fire and Mello in 1998 has recently been awarded with the Nobel Prize in Physiology or Medicine in 2006.

Recent studies have revealed that regulatory RNAs are only a very small subset of "non-coding" RNAs. A large fraction of mammalian genome sequences (at least 10% in the human genome and possibly much more [51], about 60% of the mouse genome) appear to give rise to RNA transcripts that do not code for proteins [14]. Non-coding RNAs have been found to have roles in a great variety of processes, including transcriptional regulation, chromosome replication, RNA processing and modification, messenger RNA stability and translation, and even protein degradation and translocation. As a result, non-coding RNAs are now known to be far more abundant and important than initially imagined; unfortunately their functionalities are only scarcely known.

A regulatory RNA usually employs the "antisense effect", the process of forming interactions via weak hydrogen bonds between complementary unpaired nucleotides of the regulatory RNA itself and its target RNA. Native structures of both regulatory and target RNAs are important determinants of the pairing rates and have evolved for optimizing the functions of the regulatory RNA. Generally, regulatory RNAs contain one or more loop structures (unpaired regions of the native structure) that are (almost) complementary to specific sequences in the target RNAs. Interaction with a target is usually initiated at such a loop structure of the regulatory

RNA and a loop structure from the target, forming "kissing loop pairs". The thermodynamic parameters involved in establishing the kissing loop pairs (as well as loop-single stranded RNA pairs) and the specific tertiary structures they form are mostly unknown and constitute a major challenge in gene regulation research.

This thesis presents a number of results in resolving problems related to RNA based gene regulation. More specifically we introduce new algorithms and software tools to computationally predict the exact form of interactions between a non-coding RNA and its target. Our methods can determine the joint secondary structure of two interacting RNA molecules or those that predict whether two RNAs can form a stable interaction are essential to predicting how regulatory RNAs hybridize with target mRNAs and effectively downregulate the corresponding genes. These methods can also help predict how target RNAs bind to probes on a microarray or what might be the active site of a ribosome. Central to our tools for predicting the joint structure of two interacting RNAs is the accurate prediction of the independent structures of the RNA sequences before the interaction. This thesis also introduces new algorithms and software to improve the accuracy of the existing methods for predicting the independent structure of a single RNA molecule as well.

We note that determining the exact form of RNA-RNA interactions have immediate applications in medicine. In principle, regulatory RNA molecules can be employed to silence desired genes and thus used for treating a variety of human diseases such as several types of cancer, rheumatoid arthritis, brain diseases and viral infections. Regulatory RNAs have already been demonstrated to cure disease: for example, an siRNA targeted against the activated oncogene H-Ras in proliferating cancer cells, was able to revert the cells back into normal cells [24] - H-Ras is known to be involved in many types of cancer. More recently RNAi was demonstrated to effectively turn off the mutated Fibulin 5 gene - which is responsible for wet macular generation, a disease that effects 30 million elderly people in the world. The siRNA called Cand5 (discovered and named by Acuity Pharmaceuticals) which targets the mutated Fibulin 5 gene can be directly injected into a patient's eye and thus can be used as a drug - it already passed the Phase II clinical trials with top-line results. If Cand5 passes all clinical trials with success, it would provide a landmark for the

field of RNAi-based therapeutics [10].

We would like to add that developing a successful RNAi based drug necessitates the identification of all interactions between the drug and all functional RNAs. In particular, interactions of the drug with unrelated mRNAs will likely to result in severe side effects.

The second part of this thesis is on predicting the functionality of small chemical compounds. Until recently regulation of gene expression, in all organisms, is almost exclusively attributed to regulatory proteins. However mRNA gene expression and proteomic analysis do not tell the whole story of how biological processes are carried out in the cell. Almost all gene regulation mechanisms and biochemical pathways involve small chemical compounds which act as metabolites(such as metabolic intermediates, hormones and other signaling molecules). Small chemical molecules (with molecular weights ≤ 500) are very important in the exploration of molecular and cellular functions such as normal growth, development and reproduction. They also play key roles in treating diseases: almost all medicines available today are small molecules.

Novel technological advances in chemistry have given us the ability to rapidly and efficiently synthesize large numbers of novel small chemical compounds. Furthermore, new improvements in Mass Spectrometry and Nuclear Magnetic Resonance methods have made it possible to efficiently and accurately determine the chemical structures of a given compound. Unfortunately determining the functionality of these small chemical molecules, in particular those which are effective at modulating a given biological process or disease state is still far from trivial.

Chemical compounds which are structurally similar are typically similar in physiochemical properties such as boiling/melting point, solubility, etc and as a result their functionality [47]. Thus one standard tool for determining the functionality of a small chemical compound is structural similarity search among compounds with known functionalities. Alternatively one can query small molecule databases with a *probe* compound possessing desirable biological activity to *discover* chemically similar database entries, which would have a higher probability to have the bioactivity

of interest.

The above structural similarity search and classification approach is associated with two fundamental computational problems which are addressed in this thesis:

1. The notion of similarity used in search determines the molecules that are extracted from the database must be determined such that the highest level of bioactivity discrimination can be achieved. We show how to obtain such a similarity measure through a combinatorial optimization approach.

2. It is quite important to have efficient algorithms for structural and chemical similarity search as the molecular databases of interest include several millions of compounds and linear/brute force search may take significant amount of time (several days in certain large private databases). We present a new data structure that exploits the available memory as much as possible so as to minimize the running time of search. The data structure is again optimized through combinatorial algorithms.

1.2 Contributions

As mentioned earlier, this thesis studies the problems of (i) computational RNA secondary structure prediction as well as the joint secondary structure of two interacting RNA molecules and (ii) efficient and effective classification of small chemical compounds as well as efficient data structures to handle large datasets. In particular, this thesis presents the following results.

1. We introduce the notion of *normalized free energy* or *energy density* criteria to improve the accuracy of the existing algorithms for secondary structure prediction of one or more RNA sequences [4]. The algorithms we describe in this thesis minimize a linear combination of the total energy density and total free energy of an RNA sequence. Based on this optimization function, we developed the `Densityfold` program for folding a single sequence and the `MDensityfold` program for folding multiple sequences.

2. A natural follow up problem to RNA secondary structure prediction problem is the determination of interactions between two RNA sequences. We first describe the general RNA-RNA Interaction Prediction (RIP) Problem combinatorially; Given two RNA sequences S and R (e.g. a regulatory RNA and its target), RIP problem asks to predict their joint secondary structure. We aim to compute the joint structure between S and R through minimizing their *total free energy* [3]. We then show how to obtain efficient algorithms to minimize the free energy of the joint structure via dynamic programming approach and test the accuracy of our algorithms on known joint structures. We finally apply our structure prediction techniques to compute target mRNA sequences to any given non-coding RNA molecule.

3. In order to determine the structural similarity of small chemical compounds, we focus on the k-nearest neighbors (k-nn) classification method, which deduces the bioactivity of a chemical compound based on the bioactivity of its k-nn with respect to a distance measure of choice. In this thesis we introduce use of the weighted Minkowski distance of order 1, namely wL_1 such that for each bioactivity of interest, the real valued weights w_i of the wL_1 distance are determined so as to maximize the discrimination between active and inactive compounds in a training set. The (near) *optimal* values for weights w_i are computed via a linear optimization procedure [33].

4. An efficient data structure is necessary for fast nearest neighbor search queries in large datasets with millions of compounds which is generally true for small chemical molecule databases. Space Covering Vantage Point (SCVP) trees [63] where the vantage points in each level are chosen randomly until all search space is covered, are natural choice for this purpose. Clearly, it is desirable to minimize the number of vantage points that cover the search space. We first prove that the problem of minimizing the number of vantage points at each level is an NP-hard problem. However, we show how to approximate the minimum number of vantage points and thus obtain the optimum allocation of available memory through a simple polynomial time algorithm. The resulting

data structure, which we call the deterministic multiple vantage point tree (DMVP tree), when built in full, is guaranteed to have $O(\log \ell)$ levels, where ℓ is the size of the data set [33]. If the maximum number of children of an internal node at level i is c_i, the query time guaranteed by our data structure is $O(\sum_{i=1}^{\log \ell} c_i$. Because c_i is typically a small constant, the query time is only $O(\log \ell)$, a significant improvement over linear/brute force search. In case of limited memory, techniques are developed for selecting the the optimum subtree that minimize the expected query performance.

1.3 Organization of the thesis

The remainder of the thesis is structured as follows:

1. In chapter 2, we first present the description of the following problems: (i) RNA secondary structure prediction problem, (ii) RNA-RNA interaction prediction problem and (iii) clustering and classification of small chemical compounds based on structural similarity. We then give an overview of the related work.

2. In chapter 3, a novel RNA secondary structure prediction method based on *energy density* ,*Densityfold*, is developed [4]. Densityfold aims to minimize the linear combination of the total free energy and total free energy density of an RNA sequence via a dynamic programming approach. Because the running time of the most general approach is exponential with the maximum number of branches allowed in a multibranch loop, a divide and conquer approach is developed for approximating energy density of such loops. Experimental results are supplied for demonstrating Densityfold's predictive power.

3. Even though there are a number of algorithms for predicting the secondary structure of a single RNA molecule including Densityfold, no such algorithm exists for reliably predicting the *joint* secondary structure of two interacting RNA molecules or measuring the stability of such a joint structure. In chapter 4, we address the RNA-RNA interaction problem and develop efficient algorithms to solve it. Our algorithms minimize the joint free energy

between the two RNA molecules under a number of energy models with growing complexity [3]. Because the computational resources needed by our most accurate approach is prohibitive for long RNA molecules, a heuristic approach is described while experimentally maintaining the original accuracy. Equipped with this fast approach, we apply our method to discover targets for any given antisense RNA in the associated genome.

4. The problem of structural similarity search among small chemical molecules is studied in chapter 5 using k-nearest-neighbor (k-nn) search method. Not only do we develop classification methods for molecules with unknown bioactivities, we also develop methods for designing the optimal *weighted* Minkowski distance wL_p for maximizing the discrimination between active and inactive compounds with respect to bioactivies of interest [33]. The accuracy achieved by our classifier under the optimal wL_p distance is better if not as good as the alternative methods. Furthermore in terms of running time we achieve considerably faster results compared to competition.

5. Efficient data structures for performing nearest neighbor search in large dataset is addressed in chapter 6. A variation of Space Covering Vantage Point (SCVP) tree, Deterministic Multiple Vantage Point (DMVP) tree, is presented [33]. The study indicates a deterministic selection of vantage points through exploiting the available memory, can improve the search time considerably. Theoretical analysis for the search time is also presented. In case of limited memory, we show how to obtain the subtree to fit into memory for minimizing the expected search performance.

6. Finally the thesis is concluded in chapter 7 with a brief summary of our algorithms.

Chapter 2

Definition of the Problems and Background

In this chapter, we describe problems related to discovering relations between structure and function of biomolecular compounds such as RNAs and small chemical compounds. We first describe the single RNA secondary structure prediction problem and the general methodology - referred as *free energy minimization*- for solving this problem. Later we extend this methodology to determine the interactions between two RNA sequences. The second part of the chapter focuses on the structural similarity search among small chemical molecules which is one of the standard methods used in conventional in-silico drug discovery.

2.1 RNA Secondary Structure Prediction Problem

As mentioned earlier an RNA molecule can be considered as strand of four different types of bases which are Adenine (A), Cytosine (C), Guanine (G), and Uracil (U), with two chemically distinct ends, known as the 5′ and 3′ ends. Thus RNA strands are typically represented as a string over A,C,G,U, with the left end corresponding to the 5' end of the molecule. Although RNA sequences are transcribed as single stranded molecules, many bases of an RNA molecule form basepairs through weak

hydrogen bonds. The resulting form of RNA sequence is called "secondary structure" of RNA sequence. The most common basepairs are the complementary pairs C-G and A-U being the strongest and the "wobble" pair G-U being the weakest [72].

More formally a secondary structure of an RNA sequence, $R = r_1, r_2, ...r_n$, can be defined as the set of basepairs(Figure 2.1). A basepair between nucleotides r_i and r_j $(i < j)$ is denoted by $(i \cdot j)$ The following constraints are usually imposed on RNA secondary structures:

1. Two base pairs $(i \cdot j)$ and $(i' \cdot j')$ are either identical, or else $i \neq i'$ and $j \neq j'$. Thus base triples are excluded from the definition of secondary structure.

2. Sharp U-turns are prohibited. A U-turn, called hairpin loop, must contain at least three bases.

3. Pseudoknots are prohibited. That is if $(i \cdot j)$ and $(i' \cdot j')$ are basepairs in an RNA secondary structure, then a pseudoknot occurs assuming $i < i'$, $i < i' < j < j'$.

The last condition excludes pseudoknots. Pseudoknots are excluded because energy minimizing methods based on the nearest neighbor thermodynamic model, cannot deal with them. Inclusion of pseudoknot to the RNA secondary structure problem transforms the problem into NP-hard (Non-deterministic Polynomial-time hard) problem for general case [2]. For this reason, pseudoknots are often considered as belonging to tertiary structure.

Given an RNA sequence, *RNA secondary structure prediction problem* (sometimes referred to as the *RNA folding problem*) asks to compute all pairs of bases that form hydrogen bonds. Much of the literature on RNA secondary structure prediction is devoted to the *free energy minimization* approach. This general methodology (which is sometimes called the *thermodynamic* approach) aims to compute the secondary structure by minimizing the total free energy of its substructures such as *stems*, *loops* and *bulges*. This model is almost universally accepted and it is the only available model for determining the total free energy of an RNA structure.

2.1.1 The Nearest Neighbor Thermodynamic Model

The nearest neighbor thermodynamic model aims to provide a framework that can be used to calculate the free energies of RNA secondary structures more accurately. The main premise of the nearest neighbor thermodynamic model is that the energy value of a basepair is not only determined by itself but also its nearest neighbor. Energy values for the unpaired bases are estimated according to the type of sub-structure that encloses them. Total free energy of an RNA secondary structure can be approximated as the sum of independent terms for total free energies of stacked pairs and loop sequences(Figure 2.1). The thermodynamic model has been developed in conjunction with the development of dynamic programming folding algorithms, so the independence assumptions in the thermodynamic models terms have been made compatible with the independence assumptions needed for recursive dynamic programming algorithms to work.

Stacked pairs:

Two basepairs $(i \cdot j)$ and $(i' \cdot j')$ are referred as a stacked pair if they are immediately adjacent to each other where $i' = i + 1$ and $j' = j - 1$. For each possible base pair there is a certain energy value which is stored in a static table. A group of 2 or more consecutive base pairs is called a *helix*. The first and last basepairs are referred the closing basepairs of the helix. Free energy of the helix is then calculated as the total free energy of the stack pairs between closing basepairs.

Loop Structures:

A base i' or a basepair $(i' \cdot j')$ is called accessible from a basepair $(i \cdot j)$ if $i < i' (< j') < j$ and if there is not other basepair, $(k \cdot l)$ such that $i < k < i' (< j') < l < j$. The collection of unpaired bases accessible from a given basepair $(i \cdot j)$ but not including that basepair is called the loop closed by $(i \cdot j)$, $L(i, j)$. Notice that a loop sequence can contain many stacked pairs. There are different types of loops and for each different loop type there exist a function that can estimate its free energy. The possible types of loops in an RNA secondary structure and their free energy calculations can be summarized as follows:

1. The collection of unpaired bases not accessible from any basepair is called

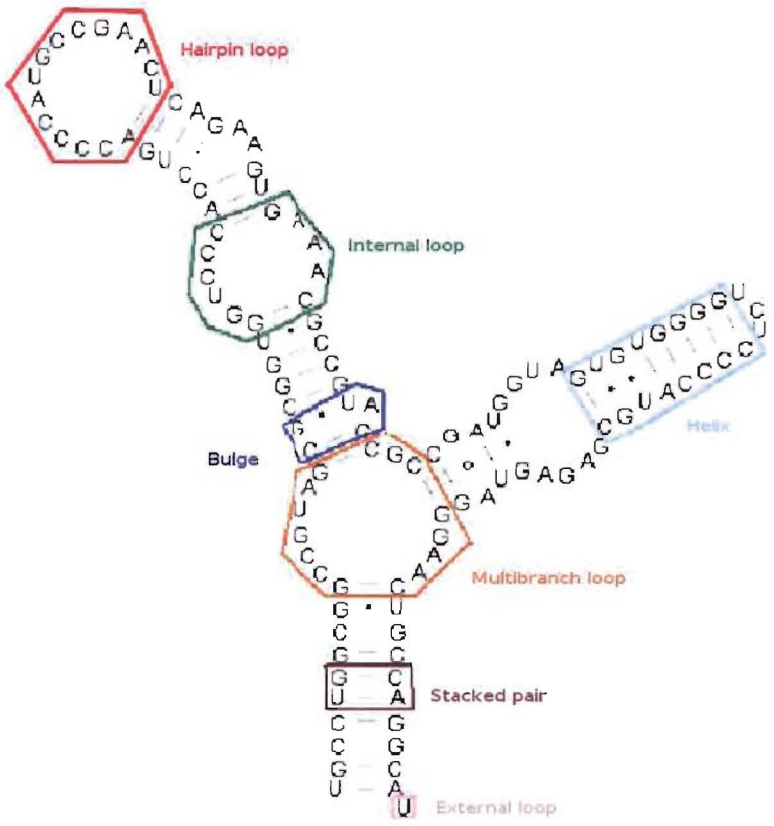


Figure 2.1: Sample pseudoknot free RNA secondary structure containing all elementary substructures.

the exterior or external loop. These loops are existent only in linear RNA sequences and the free energy of external loops can be be estimated as a function of the (i) size of the loop and (ii) terminal mismatch stacking energies (helix closing basepairs included in the loop sequence)

2. A loop, $L(i, j)$, containing no helix closing basepair is called a hairpin loop. Free energy of a hairpin loop is a linear function of (i) size of the hairpin loop, (ii) terminal mismatch stacking free energy obtained from the basepair $(i \cdot j)$, (iii) bonus free energy for hairpin loops of size < 4 and (iv) bonus or penalty free energy for special cases.

3. A loop, $L(i, j)$, containing one helix closing basepair $(i' \cdot j')$ where both $|i' - i| >$

0 and $|j - j'| > 0$, is called an internal loop. Free energy of an internal loop is a linear function of (i) size of the loop, (ii) terminal mismatch stacking energy obtained from basepair $(i' \cdot j')$ and (iii) asymmetric penalty which depends on $||i' - i| - |j' - j||$.

4. An internal loop, $L(i, j)$ (with closing basepair $(i' \cdot j')$), where either $|i' - i| = 0$ and $|j - j'| = 0$, is called a bulge. Free energy of a bulge is a linear function of (i) size of the bulge, (ii) terminal mismatch penalty and (iii) bonus or penalty free energy for special cases.

5. A loop, $L(i, j)$, containing more than one helix closing basepair is called a multi-branch loop. Because so little is known about the effects of multi branch loops on RNA stability, free energies are assigned in a way that makes the computations easy. Free energy of a multi-branch loop is approximated as a linear function of (i) number of closing basepairs in the loop, (ii) number of unpaired bases in the loop and (iii) terminal mismatch penalties.

The parameters of the nearest neighbor thermodynamic model have been determined experimentally by Turner et al. and the details of thermodynamic parameters can be found in [23]. Based on these thermodynamic parameters, a number of dynamic programming algorithms have been developed [57, 79, 44] to compute the RNA secondary structure with minimum free energy. The popular `mfold` and its more efficient version `RNAfold` (from the `Vienna package`) are implementations of these algorithms.

2.1.2 Multiple RNA secondary structure prediction

Single RNA secondary structure prediction methods disscussed above have many limitations which are usually attributed to the following factors. The total free energy is effected by tertiary interactions which are currently poorly understood and thus ignored in the energy tables [49] currently used by all structure prediction tools. There are also external, non-RNA related factors that play important roles during the folding process. Furthermore, the secondary structure of an RNA sequence is

formed as the molecule is being transcribed. A highly stable substructure, formed only after a short prefix of the RNA sequence is transcribed, can often be preserved after the completion of the transcription, even though it may not conform to a secondary structure with the minimum free energy. [1]

In order to address these issues, much of the recent research on RNA secondary structure is focused on simultaneously predicting the secondary structure of *many* functionally similar RNA sequences. The intuition underlying this approach is that functional similarity is usually due to structural similarity, which, in many cases, correspond to sequence similarity. Because this approach can utilize the commonly observed covarying mutations among aligned basepairs in a stem, the accuracy of this approach can outperform single sequence structure prediction approach.

There are three main techniques for simultaneously predicting the secondary structure of multiple sequences via energy minimization.

- The first general technique, used in particular by the `alifold` program [29] of the `Vienna package`, assumes that the multiple alignment between the input RNA sequences (in the case of `alifold`, computed by the `Clustal-W` program [67]) corresponds to the alignment between their substructures. The structure is then derived by folding the multiple alignment of the sequences. Clearly this method crucially relies on the correctness of the multiple sequence alignment; thus its prediction quality is usually good for highly similar sequences (60% or more) but can be quite poor for more divergent sequences.

- The second general technique aims to compute the sequence alignment and the structure prediction simultaneously [64, 27, 50]. When formulated as a rigorous dynamic programming procedure, the computational complexity of this technique becomes very high; it requires $O(n^6)$ time even for two sequences

[1] Another crucial issue that limits the prediction accuracy of many energy minimization based tools is that they do not allow pseudoknots. This is due to the the fact that the energy minimization problem allowing arbitrary pseudoknots is NP-hard [2]. The only software tool we are aware of which allows certain types of pseudoknots (as described by [16]) is `Pknots` [62], which suffers from efficiency problems. Thus our current implementation does not allow any pseudoknots due to efficiency considerations; however it can easily be extended to allow the class of pseudoknots captured by `Pknots`.

and is NP-hard for multiple sequences [19]. In order to decrease the computational complexity, it may be possible to restrict the number of substructures from each RNA sequence to be aligned to the substructures from other sequences. In [8], this is done through a preprocessing step which detects all statistically significant potential stems of each RNA sequence by performing a local alignment between the sequence and its reverse complement. When computing the *consensus structure*, only those substructures from each RNA sequence which are enclosed by such stems are considered for being aligned to each other. This strategy is successfully implemented by the `RNAscf` program recently developed by Bafna et al. [8].

- The final approach to multiple sequence structure prediction is the so called *consensus folding* technique. Rather than minimizing free energy, the consensus folding technique first extracts all potential stems of each input RNA sequence. The consensus structure is then computed through determining the largest set of compatible potential stems that are common to a significant majority of the RNA sequences. A good example that uses the consensus folding technique is the `comRNA` program [32] which, once all stems of length at least ℓ are extracted from individual sequences, computes the maximum number of compatible stems [2] that are common to at least k of the sequences via a graph theoretic approach. As one can expect, the consensus technique also relies on the availability of many sequences that are functionally (and hopefully structurally) similar.

2.1.3 Machine Learning Approaches

All methods described above rely on physics models of RNA structure in the form of the traditional thermodynamic model. Note that the thermodynamic parameters used by these methods are determined through experimentation where there are some limitations on which parameters are measurable, and with what accuracy these

[2] The notion of compatibility here allows the types of pseudoknots that are captured by the `Pknots` program.

parameters are measured. Some of the accuracy loss using the thermodynamic model is attributed to these limitations of the thermodynamic parameter determination. Recently machine learning methods that estimate thermodynamic parameters based on known RNA secondary structures are emerging as an alternative to the physics based methods.

A popular machine learning approach, the stochastic context-free grammars (SCFGs), provides a probabilistic method for predicting RNA secondary structure [21, 35, 36]. An implementation of this general approach, CONTRAfold, is based on the conditional log-linear models (CLLMs), a flexible class of probabilistic models which generalize upon SCFGs by using discriminative training and feature-rich scoring [20]. It is possible to define an RNA secondary structure as a vector of RNA substructures where each substructure is associated with a certain weight value. Using this model and a given set of known RNA secondary structures, CONTRAfold estimates the weights using maximum likelihood methods.

Another recent parameter estimation method is proposed by Andronescu et al. based on the constraint generation method. [6] Here each RNA secondary structure is represented using probabilistic methods where estimation of the parameters can be formulated as an optimization problem. For each RNA sequence, constraints ensures that the known RNA structure has a better energy value than all the alternative foldings. However the number of the constraints depends on the possible RNA secondary structures which is exponential. The solution for the optimization problem is found using a heuristic iterative constraint generation method.

Although both of these methods improve the prediction accuracy of the training data, the quality of the predictions highly depends on the training data. The errors in the structures and the errors in the annotations have a significant effect on these methods. Another problem is that these methods still assume the minimum energy state and ignores the kinetics of the folding process. For our research purposes, we are going to focus mostly on the physics based RNA secondary structure methods.

2.2 RNA-RNA Interaction Prediction Problem

As described above there are a number of computational methods for predicting the secondary structure of a *single* RNA molecule. However, there are only a few studies related to the problem of predicting the secondary structure formed by two RNA molecules.

2.2.1 Problem Definition

Given two RNA molecules, one being an regulatory RNA and the other its potential target, RNA-RNA Interaction prediction problem asks to compute the joint structure formed by those RNA molecules that has the minimum total free energy. A joint secondary structure between two RNA sequences is a set of basepairs where each nucleotide is paired with at most one other nucleotide, either internal or external. An illustration of the RNA-RNA joint secondary structure is given in Figure 2.2 between two RNA sequences CopT and CopA.

Figure 2.2 shows the natural joint structure of interacting RNA molecules CopA and CopT. Similarly, Figure 2.3 shows the natural joint structure of interacting RNA molecules OxyS and fhlA. The sequence written in black is the target RNA, where the red one is the regulatory RNA CopA. Target RNA is given in 5' to 3' direction whereas the regulatory RNA is given in the reverse order from 3' to 5' in order to represent the joint secondary structure easier to understand. In Figures 2.4 and 2.5 the same interactions are presented in a more illustrative manner: here blue links represent *internal bonds* whereas red links represent *external bonds* between nucleotides. The green boxes are used to mark the nucleotides which do not form any kind of bonds.

CopT is a part of the RNA that encodes repA gene in E. coli plasmid R1, and the encoded protein is responsible for the plasmid replication. But when the number of CopT in the cell increases, CopA comes into the picture. CopA is the plasmid copy number regulator RNA which is actually transcribed from the same portion of the plasmid; so CopT and CopA are cis-encoding. The CopA molecules tend to come across to the CopT molecules; they form a joint structure as given in these figures;

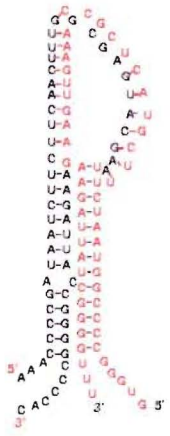


Figure 2.2: Natural joint structure between small RNA molecules CopA (regulatory[red]) and CopT (its target[black]) in *E.Coli*.

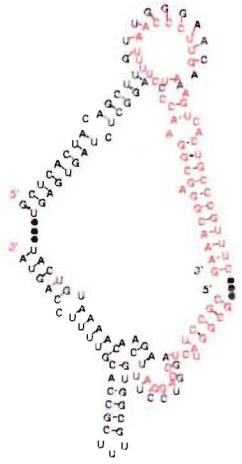


Figure 2.3: Natural joint structure between small RNA molecules fhlA (target[black]) and OxyS (regulatory[red]) in *E.Coli*. Notice that there are dots in OxyS and fhlA sequences. Actually these sequences are much longer, but whole sequence was not given in [70], and the missing sequence was not effective in the interaction [37].

which will essentially block the repA gene translation [48].

2.2.2 Previous Work

There are a number of computational tools for predicting the secondary structure of a *single* RNA molecule [44, 62, 78, 79]; these tools are especially accurate if the length of the RNA sequence is relatively *short*. There are also several algorithms to compute the "similarity" or "alignment" between two *non-interacting* RNA molecules [15, 45, 55]. However, there have only been a few studies related to the problem of predicting the secondary structure formed by two RNA molecules.

The *HyTher* package [59, 60] predicts the hybridization thermodynamics of a given duplex given the two strands; it does not aim to minimize the joint free energy

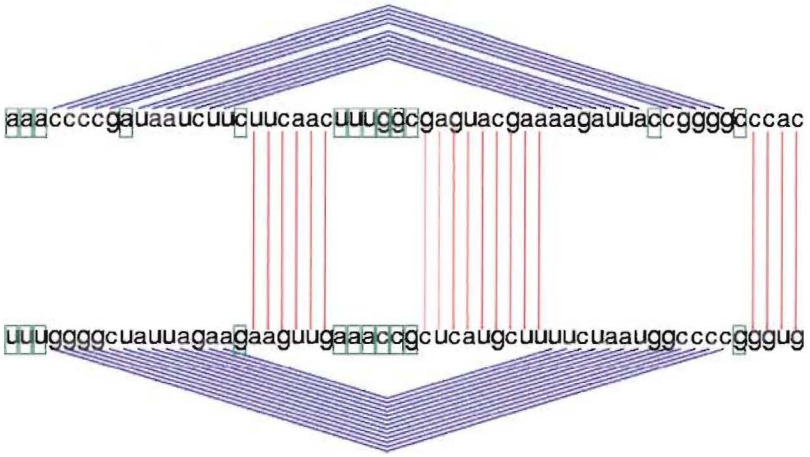


Figure 2.4: Known joint structure between small RNA molecules CopA (regulatory[bottom], in 3′ to 5′ direction) and CopT (its target[top], in 5′ to 3′ direction) in *E.Coli*.

or predict the secondary structure of the interacting RNA strands. The *Pairfold* program [5] aims to predict the secondary structure of two interacting RNA sequences by simply concatenating two RNA strands and performing a secondary structure prediction as if there is only one strand, using the *mfold* algorithm (for folding a single strand [44, 78, 79]). Because *mfold* avoids pseudoknots, possible topologies that can be predicted by *PairFold* are very limited; in fact *Pairfold* can not predict any "kissing" hairpin loops, which are essential to joint structure prediction of two RNA sequences (See Figure 2.6 for example). In principle, *PairFold* can employ the *pknots* method of Rivas and Eddy [62] which can predict certain types of pseudoknots. However the pseudoknot types allowed by *pknots* (as per the characterization in [16]) do not capture any non-trivial kissing loop complex such as the ones explored in this work. Thus even by employing *pknots*, the *PairFold* approach would not be able to predict the joint structure of interacting RNA molecules of interest. A more recent paper [58] describes the *IRIS* tool which aims to solve the joint structure prediction problem in a more formal manner. *IRIS* is based on a simple energy model that considers the free energies of paired bases only. This is quite similar to the energy model of Nussinov and Jacobson [57] for a single RNA

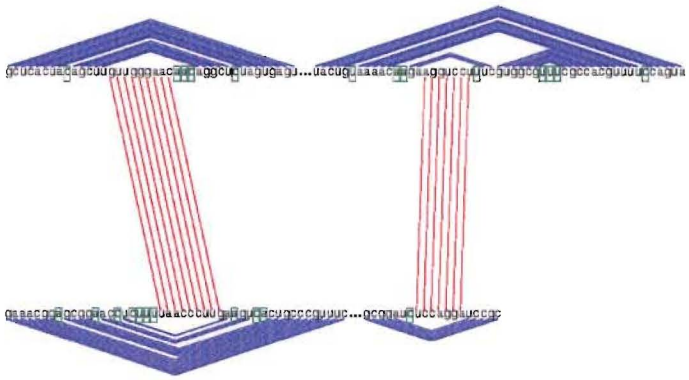


Figure 2.5: Known joint structure between small RNA molecules fhlA (target[bottom], in 3′ to 5′ direction) and OxyS (regulatory[top], in 5′ to 3′ direction) in *E.Coli*.

molecule, and almost identical to the *basepair energy model*, which is described in Section 4.2 as a warm-up exercise. As can be expected, this energy model leads to unreliable predictions: on the only known natural joint RNA structure examined in [58] (OxyS-fhlA pair), the structure predicted by *IRIS* is topologically different from the natural one.

2.3 Classification of Small Chemical Molecules

Given a notion of similarity among data elements, it is usually possible to obtain a corresponding *distance* measure; searching for structurally most similar molecules to a query molecule in this context corresponds to searching for molecules with the smallest distance to the query molecule. The key premise of this approach is that the notion of a distance is mathematically well defined and algorithms for handling distance based classification, clustering and search are better understood. For example, the search for the most similar molecule to a query compound becomes the Nearest Neighbor Search (NN) problem in the distance domain. This problem is well studied in computer science and a number of efficient algorithms are available for it.

There are various ways to define the descriptors/parameters for the chemical

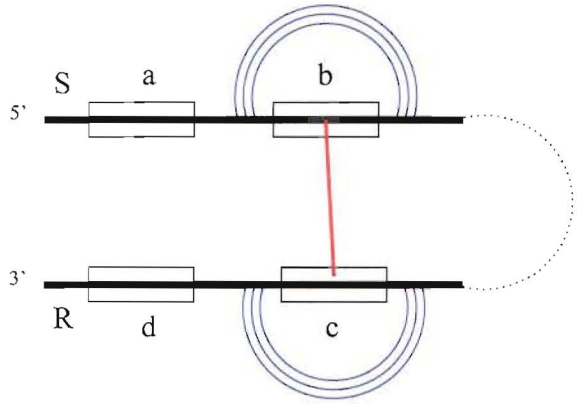
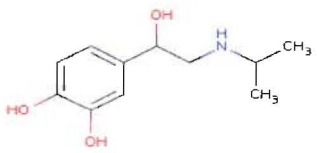


Figure 2.6: Sample RNA-RNA interaction that can't be captured by *Pairfold* employing *mfold*. Concatenating two sequences end to end makes such a kissing hairpin structure be treated as a pseudoknot by single RNA folding algorithms.

structures [9, 1], which either (1) merely reflect the structural organization of molecules in qualitative manner, such as those used in the popular *structural fingerprints* (employed in NCBI's PubChem database), e.g. the existence of a doubly bonded Carbon pair, a three membered ring, an aromatic atom etc. [46] or (2) reflect various local and global physical-chemical molecular features (chemical descriptors) which are quantitative, such as atomic weight, aromaticity, hydrophobicity, the number of specific atoms, charge, density, etc(See Figure 2.7). These descriptors serve as independent variables for *QSAR* (Quantitative Structure-Activity Relationship) tools including the structural similarity search engines in chemical compound databases.

Given an adequate set of descriptors, it is desirable to have a measure of similarity or alternatively a distance measure under which chemically equivalent molecules have a high level of similarity or small distance, and non-equivalent compounds have a low level of similarity or large distance. The most common measure of similarity amongst sets of molecular descriptors is the so called *Tanimoto coefficient* [73]. Given two descriptor sets (which can be organized in arrays) X and Y, the Tanimoto coefficient is defined to be the ratio of the number of descriptors that are identical in X and Y and the total number of descriptors available for X and Y. The Tanimoto coefficient is in the range $[0, 1]$; a value close to 1 implies similarity and a value close to 0 implies a dissimilarity among the two descriptor sets compared.



Structural Fingerprints:

is_aromatic	1
has_ring	1
has_Sulfur	0
has_Carbon	1

B(x)=10100001000011010100110101000

Chemical (Conventional) Descriptors:

Number of atoms	25
Density	0.24
Number of rings	1
Weight	12.3

B(x)=(25,0.24,1,12.3,.....5,2.12.....)

Figure 2.7: Structural and Conventional Chemical Descriptor representation of a given small chemical molecule.

Often a collection of descriptors are represented as a bit-vector (e.g. structural fingerprints) where each one of the n possible descriptors is assigned a *dimension*, i.e. natural number between 1 and n (this is the representation used by PubChem and other databases). Let $B(x)$ represent the bit-vector corresponding to a molecule x and let $B(x)[i]$ represent its i^{th} dimension. Given two compounds x and y, the Tanimoto coefficient $T(x,y)$ is then defined as $T(x,y) = (\sum_{i=1}^{n}(B(x)[i] \wedge B(y)[i])) \ / \ (\sum_{i=1}^{n}(B(x)[i] \vee B(y)[i]))$.

Although the Tanimoto coefficient provides a measure of *similarity*, it is possible to define a *Tanimoto distance measure* as $D_T(x,y) = 1 - T(x,y)$. Notice that a Tanimoto distance close to 0 implies a Tanimoto coefficient close to 1, i.e. a high level of similarity and a Tanimoto distance close to 1 implies a Tanimoto distance close to 0, i.e. a low level of similarity between x and y.

The Tanimoto coefficient is very popular mostly due to its simplicity. For real valued descriptor arrays (where each dimension has a real value) it is also quite common to use the Minkowski distance of order p, denoted L_p for measuring their similarity. Given two real valued n dimensional descriptor arrays X and Y, their Minkowski distance of order p, namely L_p, is defined as $L_p(X,Y) = (\sum_{i=1}^{n} |X[i] - Y[i]|^p)^{1/p}$. When comparing two structural fingerprints $B(x)$ and $B(y)$, the Minkowski distance

of order 1 is equivalent to the well known Hamming distance (see for example [11]): $H(B(x), B(y)) = \sum_{i=1}^{n} |B(x)[i] - B(y)[i]|$.

In order to capture the similarity between compounds more accurately with respect to a particular bioactivity, more sophisticated distance measures can be used. For example, it is possible to assign a relative importance to each structural descriptor in the form of a weight $w_i \in [0, 1]$. The resulting *weighted* Minkowski distance of order 1 can then be defined for two descriptor arrays X and Y as $wL_1(X, Y) = \sum_{i=1}^{n} w_i \cdot |X[i] - Y[i]|$.[3]

2.3.1 Classification methods for small molecules.

The descriptor arrays described above can be used for classification of compounds according to a given bioactivity.

One of the most popular classification techniques is the MLR (Multiple Linear Regression) [18] method which quantifies the activity level of a descriptor array X as: $Activity(X) = c + \sum_{i=1}^{n} \sigma_i \cdot X[i]$ where c is a constant. If $Activity(X) \geq t$ for a (user specified) threshold value t then it is likely that the molecule is active with respect to the bioactivity of interest. Notice that the MLR classifier is described by a planar separator in the multi-dimensional descriptor array space; those points on one side of the separator are classified as active and those on the other side are classified as inactive. There are many different optimization criteria for determining the separator plane, i.e. the coefficients σ_i. The most widely used one (which we used in our experiments) is the partial least squares criteria [25], which suggests to minimize the sum of the squares of differences between actual and predicted activity levels of the compounds in a training set. The separator plane which satisfies this criteria is NP-hard to compute deterministically but can be approximated through genetic algorithms, local search heuristics, etc.

Another popular statistical classification method is Linear Discriminant Analysis(LDA) [43]. Given a set of descriptor arrays, LDA computes a linear projection

[3]To the best of our knowledge all recent studies in this direction show how to assign *binary values* to weights w_i i.e. how to choose the specific descriptors that are most relevant for the application of interest (e.g. [77, 31]). As will become clear later in this Chapter, we show how to compute optimal *real valued weights* so as to improve the predictive power of our classifier.

of the descriptor array space into a Euclidean space with 2 or 3 dimensions (i.e. each descriptor array is mapped to a point in the 2/3-D Euclidean space). The projection aims to maximize the ratio of between-class variance and within-class variance. The projection of descriptor arrays to points in the Euclidean space is followed by the computation of a line/plane which best separates the active and inactive compounds, i.e. maximizes the accuracy of the classifier. For a given query compound with unknown activity, its class is then simply determined by checking to which subspace its projection falls into; clearly this can be performed very fast.

It is also possible to perform compound classification via well known machine-learning techniques such as SVM (Support Vectors Machines) [75] and, more commonly, ANN (Artificial Neural Networks) [80].

All these QSAR techniques (i.e. compound classifiers) have their own advantages and drawbacks. Statistical techniques such as LDA and MLR typically produce lower accuracy compared to the machine-learning approaches. On the other hand ANN only returns a binary value for the bioactivity (YES or NO) and provides no insight into the level of the bioactivity or the importance of the descriptors with respect to the bioactivity. It also does not provide a way of probing/similarity search, and can be somewhat slow.

2.3.2 Similarity search among small molecules.

The number of the public small chemical compound databases is fastly increasing. More importantly the number of compounds in these databases are also increasing exponentially. Currently one of the major small chemical compound databases, PubChem, contains 100.000 compounds with known bioactivities and a total of 10 million unique small chemical compounds. This initiative is expected to lead new techniques to reveal the relationship between the structural information of chemical compounds and their bioactivities. It is anticipated that these projects will also facilitate the development of new drugs by providing early stage chemical compounds to validate new drug targets which could be then move into drug-development pipeline.

How to use a distance measure for capturing the functional similarity among chemical compounds is described above. Classification of new compounds under

this distance measure can be performed through nearest neighbor queries. Another possibility is to use range queries where all the compounds within a search range is returned. It is quite easy to modify range queries by iteratively increasing the the range until all nearest neighbors are found.

The primary example of these *distance based* proximity search data structures is the Vantage Point (VP) Trees [69] which exploits the triangle inequality satisfied by the metric distance measures. In a VP tree, efficient similarity search in a large data set is achieved through iterative pruning. Among the data elements, the VP Tree randomly picks a Vantage Point V and partitions the data set into two equal size subsets according to their proximity to V. Those which are *close* to V form the *inner partition* and those which are *far* form the *outer partition*. The two subsets are further partitioned via the iterative application of the above procedure until each subset includes a single data element.

When performing a similarity search, the query element X is first compared to the Vantage Point of the entire set. If X is sufficiently close to V the search is performed in the its *inner partition*. If X is sufficiently far from V the search is performed in the *outer partition*. It is possible the X is neither too close nor too far; in this situation the search is performed simultaneously in both partitions implying that no pruning has been achieved.

A modification to traditional VP trees, which we call Space Covering VP Trees (or SCVP trees) was described by Sahinalp et al. [63] to avoid situations in which pruning is not achieved. At each level of the SCVP tree there are multiple vantage points which are chosen in a way that the union of the inner partitions of these vantage points cover the entire data set. In other words, each data element is included in at least one of the inner partitions of a vantage point. Thus a SCVP tree has multiple branches at each internal node, each representing a vantage point and its inner partition. No branch exists for representing an outer partition. If a query element is not close to any of the vantage points at a given level, it is deduced that there are no similar items to it in the data set.

The SCVP trees introduce some redundancy in the representation of the data elements: clearly each data element may be included in more than one inner partition

and thus need to be represented in more than one subtree. Thus the memory requirements of the SCVP tree can be fairly large. In case the full SCVP Tree requires more memory than available, some of the lower levels could be cut out - after which linear search needs to be employed.

Chapter 3

RNA Structure Prediction via Densityfold

As described in chapter 2, the most commonly used objective in secondary structure prediction is total free energy minimization. In the context of multiple sequence structure prediction, this objective can be used in conjunction with additional criteria such as covariation in mutations on predicted stems etc.

The goal of this thesis is to show that delocalizing the thermodynamic cost of forming an RNA substructure by considering the notion of *energy density* can improve on secondary structure prediction via total free energy minimization. We describe a new algorithm and a software tool that we call `Densityfold` which aims to predict the secondary structure of an RNA sequence by minimizing the sum of energy densities of individual substructures. We believe that our approach may help understand the process of nucleation that is required to form biologically relevant RNA substructures.

Our starting observation is that potential stems that are most commonly realized in the actual secondary structure are those whose *free energy density* (i.e. length normalized free energy) is the lowest. Figure 3.1(a) depicts the known secondary structure of the *E.coli* 5S rRNA sequence. This sequence is one of the central examples used in [8] for illustrating the advantage of multiple sequence structure prediction approach (i.e. `RNAscf`) over single sequence structure prediction (i.e. `mfold/RNAfold`). Indeed, the `mfold/RNAfold` prediction for this sequence is quite

poor as can be seen in figure 3.1(d). However, although `RNAscf` prediction using 20 sequences from 5s rRNA family is quite good, as reported in [8], the accuracy of the prediction deteriorates considerably when only 3 sequences, *E.coli*, *asellus aquaticus* and *cyprinus carpio* are used; this is illustrated in figure 3.1(e).[1] The prediction accuracy of the `alifold` program is also poor as depicted in figure 3.1(f). Most importantly, all of the above programs miss the most significant stem (enclosed by the basepair involving nucleotides 79 and 97) depicted in figure 3.1(b); when normalized by length, the `mfold/RNAfold` free energy table entry of this basepair is the smallest among all entries. (Compare this to the prediction of our program `Densityfold`, given in figure 3.1(c).)

We believe that some of the accuracy loss in structure prediction via total energy minimization can be attributed to "chance stems" which are sometimes chosen over "actual stems" due to problems commonly encountered in *local sequence alignment*. A stem is basically a local alignment between the RNA sequence and its reverse complement. Some of the energy minimization approaches (e.g. `RNAscf` program [8]) explicitly perform a local alignment search between the input RNA sequence and its reverse complement, in order to extract all potential stems of interest. However not all significant potential stems are realized in the actual secondary structure.

In the context of searching for significant alignments, the problems attributed to Smith-Waterman approach is usually considered to be a result of:
(1) the *shadow effect*, which refers to long alignments with relatively low conservation levels often having a higher score (and thus higher priority) than short alignments with higher conservation levels, and
(2) the *mosaic effect*, which refers to two highly conserved alignments with close proximity being identified as a single alignment, hiding the poorly aligned interval in between.
It is possible that the stem discovery process, which is performed either explicitly (e.g. in `RNAscf`) or implicitly (e.g. in `mfold`), may encounter with similar problems. For example, two potential stems, which, by chance, occur in close proximity, can

[1] This example is particularly interesting as the independent `mfold/RNAfold` prediction for some of these sequences are very accurate.

easily be chosen over a conflicting longer stem due to the mosaic effect: the free energy penalty of an internal loop (which will be left in between the two chance stems) is often insignificant compared to the benefit of "merging" two stems.

In the context of local sequence alignment, the impact of these effects could be reduced by the use of *normalized sequence alignment* introduced by Arslan, Egecioglu and Pevzner [7]. The normalized local alignment problem asks to find a pair of substrings with maximum possible alignment score, normalized by their length ($+L$, a user defined parameter to avoid "trivial" alignments of length 1).

Inspired by this approach we propose to apply a *normalized free energy* or *energy density* criteria to compute the secondary structure of one or more RNA sequences. The algorithms we present aim to minimize the sum of *energy densities* of the substructures of an RNA secondary structure.[2] The *energy density of a basepair* is defined as the free energy of the substructure that starts with the basepair, normalized by the length of the underlying sequence. The energy density of an unpaired base is then defined to be the energy density of the closest basepair that encloses it. The overall objective of secondary structure prediction is thus to minimize the total energy density of all bases, paired and unpaired, in the RNA sequence.

The algorithms we describe also enables one to minimize a linear combination of the total energy density and total free energy of an RNA sequence. Based on these algorithms, we developed the `Densityfold` program for folding a single sequence and the `MDensityfold` program for folding multiple sequences. We tested the predictive power of our programs on the RNA sequence families used by Bafna et al. [8] to measure the performance of the `RNAscf` program. We compare `Densityfold` and `MDensityfold` against all major competitors based on energy density minimization criteria - more specifically `mfold/RNAfold`, the best example of single sequence energy minimization, `RNAscf`, the best example of multiple sequence energy minimization without an alignment and `alifold`, the best example of multiple sequence energy minimization with an alignment. We show that when only one or a small

[2]Note that, unlike the Arslan, Egecioglu, Pevzner approach we do not need to introduce an additive factor, L, artificially: a basepair in an RNA structure has at least three nucleotides in between.

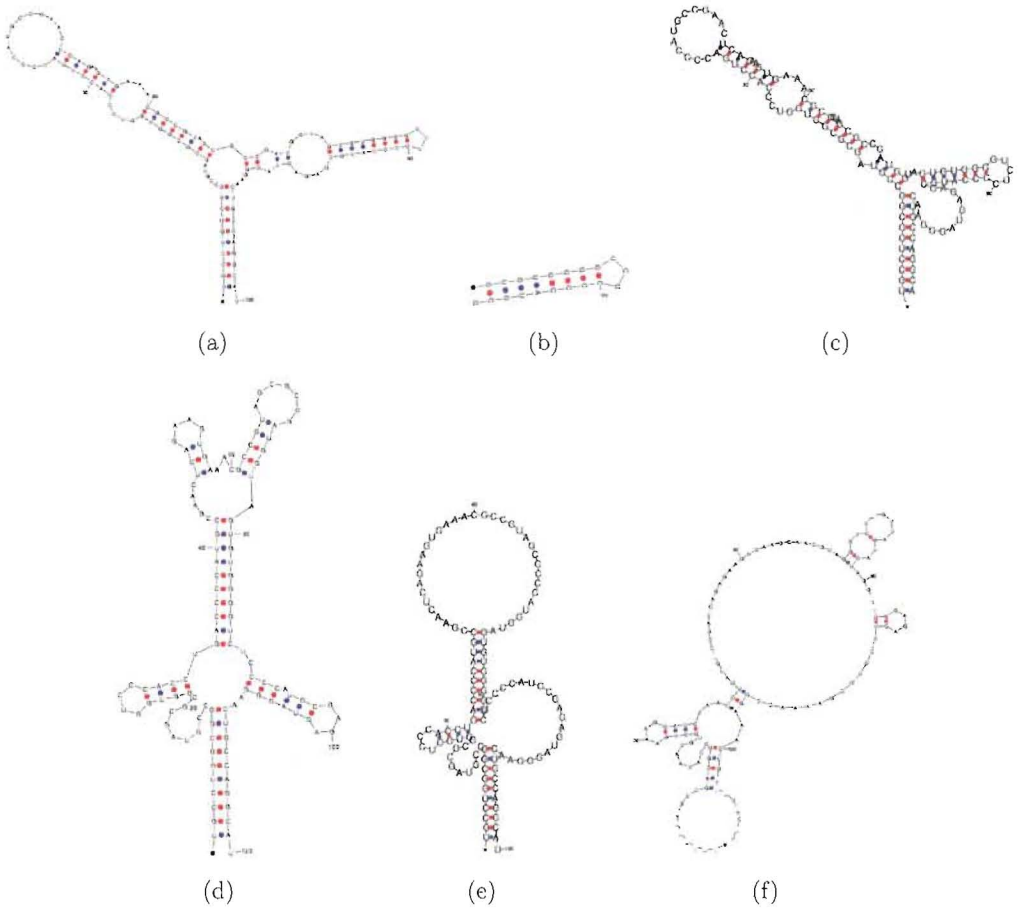


Figure 3.1: (a) Known secondary structure of the *E.coli* 5S rRNA sequence. (b) The substructure with minimum energy density (missed by `mfold/RNAfold`, `RNAscf` and `alifold` programs). (c) Structure prediction by our `Densityfold` program. We capture the substructure with minimum energy density and correctly predict 28 of the 37 basepairs in the known structure. (d) Structure prediction by `mfold/RNAfold` program - only 10 of the 37 basepairs correctly predicted (e) Structure prediction by `RNAscf` program (consensus with the the *asellus aquaticus* and *cyprinus carpio* 5S rRNA sequences) - only 10 of the 37 basepairs correctly predicted (f) Structure prediction by `alifold` program (consensus with the *asellus aquaticus* and *cyprinus carpio* 5S rRNA sequences) - only 3 of the 37 basepairs correctly predicted.

number of functionally similar sequences are available, `Densityfold` can overperform the competitors, establishing the validity of energy density criteria as an alternative to the total energy criteria for RNA secondary structure prediction.

In the remainder of the chapter we first describe a dynamic programming approach for predicting the secondary structure of an RNA sequence by minimizing the total free energy density. Then we show how to generalize this approach to minimize a linear combination of the free energy density and total free energy, a criteria that seems to capture the secondary structure of longer sequences. Because the running time of the most general approach is exponential with the maximum number of branches allowed in a multibranch loop we show how to approximate the energy density of such loops through a divide and conquer approach which must be performed iteratively until a satisfactory approximation is achieved. We finally provide some experimental results.

3.1 Energy Density Minimization for a Single RNA Sequence

We start with description of our dynamic programming formulation for minimizing the total free energy density of the secondary structure of an RNA sequence. We denote the input sequence by $S = S[1 : n]$; the i^{th} base of S is denoted by $S[i]$ and $S[i].S[j]$ denotes a basepair. Given input sequence S, its secondary structure $ST(S)$ is a collection of basepairs $S[i].S[j]$. A substructure $ST(S[i,j])$ is always defined for a basepair $S[i].S[j]$ and corresponds to the structure of the substring $S[i,j]$ within $ST(S)$. The basepair $S[i].S[j]$ is said to *enclose* the substructure $ST(S[i,j])$. The free energy of the substructure $ST(S[i,j])$ is denoted by $E_S(i,j)$. Thus the free energy density of $ST(S[i,j]$, denoted by $ED_S(i,j)$, is defined to be $E_S(i,j)/(j-i+1)$.

The notion of the free energy density of a substructure enables us to attribute an energy density value to each base $S[i]$. The individual energy density of $S[i]$, denoted $ED(i)$ is defined as the energy density of the smallest substructure that

encloses $S[i]$. More specifically, let k be the largest index in S such that $S[k].S[\ell]$ form a basepair in $ST(S)$ for some ℓ with the property that $k < i < \ell$. Then the energy density attributed to $S[i]$ is $ED_S(k, \ell)$.

Our goal is to compute a secondary structure where the total energy density attributed to the bases is minimum possible. We now show how to minimize the total free energy density for S.

We first give some notation. The values of the following thermodynamic energy functions are provided in [49].

1. $eH(i, j)$: free energy of a hairpin loop enclosed by the basepair $S[i].S[j]$.

2. $eS(i, j)$: free energy of the basepair $S[i].S[j]$ provided that it forms a stacking pair with $S[i + 1].S[j - 1]$.

3. $eBI(i, j, i', j')$: free energy of the internal loop or a bulge that starts with basepair $S[i].S[j]$ and ends with basepair $S[i'].S[j']$ (an internal loop becomes a bulge if $i' = i + 1$ or $j' = j - 1$).

4. $eM(i, j, i_1, j_1, i_2, j_2, \ldots, i_k, j_k)$: free energy of a multibranch loop that starts with basepair $S[i].S[j]$ and branches out with basepairs $S[i_1, j_1], S[i_2, j_2], \ldots, S[i_k, j_k]$.

5. $eDA(j, j - 1)$: free energy of an unpaired dangling base $S[j]$ when $S[j - 1]$ forms a basepair with any other base (used for approximating eM).

By using the above functions we need to compute the following tables that correspond to total energies and energy densities of potential substructures.

1. $ED(j)$: minimum total free energy density of a secondary structure for substring $S[1, j]$.

2. $E(j)$: free energy of the energy density minimized secondary structure for substring $S[1, j]$.

3. $ED_S(i, j)$: minimum total free energy density of a secondary structure for $S[i, j]$, provided that $S[i].S[j]$ is a basepair.

4. $E_S(i,j)$: free energy of the energy density minimized secondary structure for the substring $S[i,j]$, provided that $S[i].S[j]$ is a basepair.

5. $ED_{BI}(i,j)$: minimum total free energy density of a secondary structure for $S[i,j]$, provided that there is a bulge or an internal loop starting with basepair $S[i].S[j]$.

6. $E_{BI}(i,j)$: free energy of an energy density minimized structure for $S[i,j]$, provided that a bulge or an internal loop starting with basepair $S[i].S[j]$.

7. $ED_M(i,j)$: minimum total free energy density of a secondary structure for $S[i,j]$, such that there is a multibranch loop starting with basepair $S[i].S[j]$.

8. $E_M(i,j)$: free energy of an energy density minimized structure for $S[i,j]$, provided there is a multibranch loop starting with basepair $S[i].S[j]$.

The above tables are computed via the following dynamic programming formulation. Note that as per `mfold/RNAfold` method we do not have any penalty for the unpaired bases at the very ends of the secondary structure.

$$ED(j) = \min \left\{ \begin{array}{l} ED(j-1) \\ \min_{1 \le i \le j-1} \{ED(i-1) + ED_S(i,j)\} \end{array} \right\}$$

$$ED_S(i,j) = \min \left\{ \begin{array}{ll} +\infty, & (i) \\ eH(i,j), & (ii) \\ 2\frac{eS(i,j)+E_S(i+1,j-1)}{j-i+1} + ED_S(i+1,j-1), & (iii) \\ ED_{BI}(i,j), & (iv) \\ ED_M(i,j) & (v) \end{array} \right\}$$

$$ED_{BI}(i,j) = \min_{i',j' | i<i'<j'<j} \left\{ \frac{eBI(i,j,i',j') + E_S(i',j')}{j-i+1} \cdot [(i'-i) + (j-j')] + ED_S(i',j') \right\}$$

$$ED_M(i,j) = \min_{\substack{i_1,j_1,\ldots,i_k,j_k \\ |i<i_1<j_1<\ldots i_k<j_k<j}} \left\{ \begin{array}{ll} \frac{e_M(i,j,i_1,j_1,\ldots i_k,j_k)+E_S(i_1,j_1)+\ldots E_S(i_k,j_k)}{j-i+1} \cdot [i_1 - i \ldots + j - j_k] \\ + \quad [ED_S(i_1,j_1) + \ldots ED_S(i_k,j_k)] \end{array} \right\}$$

For each (i,j), once the total energy density under the three possible structures (stack, bulge/internal loop and multibranch loop) are computed, the corresponding free energies can be computed as follows.

$$E_S(i,j) = \left\{ \begin{array}{ll} (i): & +\infty, \\ (ii): & eH(i,j), \\ (iii): & E_S(i+1,j-1) + eS(i,j), \\ (iv): & E_{BI}(i,j), \\ (v): & E_M(i,j) \end{array} \right\}$$

$$E_{BI}(i,j) = eBI(i,j,i',j') + E_S(i',j') \qquad for\ i',j'\ computed\ above$$

$$E_M(i,j) = eBI(i,j,i_1,j_1,\ldots i_k,j_k) + E_S(i_1,j_1)\ldots + E_S(i_k,j_k),$$
$$for\ i_1,j_1\ldots i_k,j_k\ computed\ above$$

The algorithm above assumes that the maximum number of branches in a multibranch loop is k. Under this assumption the running time of the algorithm is $O(n^{k+2})$ and the space complexity is $O(n^2)$. Clearly this is not very practical for large values of k. Thus for $k > 2$ we make a number of simplifying assumptions on the free energy of a multibranch loop akin to the assumptions made by the `mfold/RNAfold` method. In particular we assume that the multibranch loop energy $eM(i,j,i_1,j_1,\ldots i_k,j_k)$ is a linear function of the number of unpaired bases and the dangling energies of the bases that follow the basepairs in the multibranch loop, namely $eDA(i+1,i), eDA(j-1,j), \ldots$. This assumption helps `mfold/RNAfold` to partition a multibranch loop into two iteratively, so that its minimum possible free energy can be computed in time linear with the size of the loop.

However, because we want to minimize the normalized free energy of the multi-branch loop, which is non-linear, we can not apply the same divide-and-conquer approach directly. Thus we provide an alternative formulation which (at least in practice) converges to the correct value of the multibranch loop energy density in a small number of iterations. We describe this formulation in the following section.

3.2 Minimizing a linear combination of the energy density and energy

The initial tests we performed on the above dynamic programming formulation provided good outcomes for short RNA sequences; however as the sequence length increased, the predictive performance of this formulation deteriorated considerably. We noticed that although the energy density itself can help identify short structural motifs well, it may not provide the right criteria for "stitching them together". Thus, in this section we describe a modified version of the dynamic programming formulation we gave above for energy density minimization. The goal of this modified version is to minimize a linear combination of the energy density and the total free energy. More specifically, for any $x \in \{S, BI, M\}$ let $ELC_x(i,j) = ED_x(i,j) + \sigma \cdot E_x(i,j)$. The function we would like to optimize is thus $ELC(n) = ED(n) + E(n)$.

$$ELC(j) = \min \left\{ \begin{array}{l} ELC(j-1) \\ \min_{1 \le i \le j-1} \{ELC(i-1) + ELC_S(i,j)\} \end{array} \right\}$$

$$ELC_S(i,j) = \min \left\{ \begin{array}{ll} +\infty, & (i) \\ eH(i,j) \cdot (1+\sigma), & (ii) \\ 2\frac{eS(i,j)+E_S(i+1,j-1)}{j-i+1} + ELC_S(i+1,j-1) + \sigma \cdot eS(i,j), & (iii) \\ ELC_{BI}(i,j), & (iv) \\ ELC_M(i,j) & (v) \end{array} \right\}$$

$$ELC_{BI}(i,j) = \min_{i',j'|i<i'<j'<j} \left\{ \begin{array}{l} \frac{eBI(i,j,i',j')+E_S(i',j')}{j-i+1} \cdot [(i'-i)+(j-j')] \\ +ELC_S(i',j') + \sigma \cdot eBI(i,j,i',j') \end{array} \right\}$$

For computing the value of our optimization function for multibranch loops efficiently we have to perform an approximation to the multibranch loop energy density through a divide and conquer approach For this we have to define a new energy table $\overline{ELC}_M^{[i,j]}(k,\ell) = \overline{ED}_M^{[i,j]}(k,\ell) + \sigma \cdot \overline{E}_M^{[i,j]}(k,\ell)$ where $\overline{E}_M^{[i,j]}(k,\ell)$ and $\overline{ED}_M^{[i,j]}(k,\ell)$ are the free energy and the energy density of the optimal substructures for $S[k,\ell]$ provided that both $S[k]$ and $S[\ell]$ are on a multibranch loop starting with the basepair $S[i].S[j]$.

$$ELC_M(i,j) = \sigma \cdot a + \min_{i<k<j} \left\{ \overline{ELC}_M^{[i,j]}(i,k) + \overline{ELC}_M^{[i,j]}(k+1,j) \right\}$$

Here a is the multibranch loop opening score. Define:

$$\overline{b} = \frac{\widehat{E}_M(i,j)}{(j-i+1)}$$

where $\widehat{E}_M(i,j)$ is an estimation (a lower bound) for $E_M(i,j)$ of the optimal structure. The initial value of $\widehat{E}_M(i,j)$ is obtained through the following dynamic programming routine.

$$\widehat{E}_M(i,j) = a + \min_{i<k<j} \left\{ \overline{E}_M(i,k) + \overline{E}_M(k+1,j) \right\}$$

$$\overline{E}_M(k,k) = b$$

$$\overline{E}_M(k,\ell) = \min \left\{ \begin{array}{l} E_S(k,\ell) + c + eDA(k-1,k) + eDA(\ell,\ell+1) \\ \min_{k\leq h<\ell} \{ \overline{E}_M(k,h) + \overline{E}_M(h+1,\ell) \} \end{array} \right\}$$

Here c is the contribution for each basepair on the multibranch loop and b is the unpaired base penalty. Based on this initial estimation $\widehat{E}_M(i,j)$ we have:

$$\overline{ELC}_M^{[i,j]}(k,k) = \overline{b} + \sigma \cdot b$$

$$\overline{ELC}_M^{[i,j]}(k,\ell) = \min \left\{ \begin{array}{l} ELC_S(k,\ell) + \sigma \cdot [c + eDA(k-1,k) + eDA(\ell,\ell+1)] \\ min_{k \le h < \ell} \{ \overline{ELC}_M^{[i,j]}(k,h) + \overline{ELC}_M^{[i,j]}(h+1,\ell) \} \end{array} \right\}$$

The corresponding energies of the substructures are as in the previous section:

$$E_S(i,j) = \left\{ \begin{array}{ll} (i): & +\infty, \\ (ii): & eH(i,j), \\ (iii): & E_S(i+1,j-1) + eS(i,j), \\ (iv): & E_{BI}(i,j), \\ (v): & E_M(i,j) \end{array} \right\}$$

$$E_{BI}(i,j) = eBI(i,j,i',j') + E_S(i',j') \qquad for\ i',j'\ computed\ above$$

$$E_M(i,j) = eM(i,j,i_1,j_1,\ldots i_k,j_k) + E_S(i_1,j_1)\ldots + E_S(i_k,j_k),$$
$$for\ i_1,j_1\ldots i_k,j_k\ computed\ above$$

Note that if $E_M(i,j) \geq \widehat{E}_M(i,j) + \epsilon$ for some user defined (small) value of ϵ we set $\widehat{E}_M(i,j) = \widehat{E}_M(i,j) + \epsilon$ and re-iterate the above procedure for computing $ELC_M(i,j)$. The reader can easily verify that the running time of this dynamic programming algorithm is $O(n^4)$.

3.2.1 Multiple Sequence Energy Density Minimization

The dynamic programming algorithm for minimizing $ELC(n)$ for a single sequence is generalizable to multiple sequences without difficulty. Here we assume that the multiple alignment between the input RNA sequences which can be computed by any multiple alignment method (e.g. `Clustal-W` program [67]), corresponds to the alignment between their structures. The consensus structure is then derived by folding the multiple alignment of the sequences where the linear combination of

energy and energy density of all bases of input RNAs is minimized. The *total energy and total energy density* of each substructures in the alignment is assigned to the energy and, respectively, energy density of the corresponding consensus substructure. The gaps are also included in the calculations as a base.

The reader can verify that for m sequences the running time of this dynamic programming algorithm is $O(m \cdot n^4)$.

3.3 Experimental Results and Discussion

We implemented and tested the performance of our algorithms for minimizing the linear combination of the energy density and the total free energy of a single sequence as well as of multiple sequences, respectively called `Densityfold` and `MDensityfold`. Two datasets are selected for evaluating performance of our algorithms which are used by two recent RNA secondary structure prediction tools, respecively [8] and [20].

Our first test set is comprised of the same 12 RNA families from the Rfam database [28] used by Bafna et al. [8] for testing the performance of `RNAscf` program. Using this test set, we compared the performance of `Densityfold` and `MDensityfold` with varying values of σ (which determines the contribution of the total energy to the optimization function) against `mfold/RNAfold`, the best single sequence energy minimization program, `alifold` the best multiple sequence energy minimization program that uses the alignment between the input sequences, and `RNAscf` the best multiple sequence energy minimization program that computes the alignment and the folding simultaneously. In the context of multiple sequence folding, our goal is to demonstrate the predictive power of `MDensityfold` when only a limited number of sequences are available; thus we only report on the jointly predicted structures of a pair of sequences, randomly selected from each family.

The most common measure for demonstrating the predictive power of a single sequence secondary structure determination method is the number of correct base-pairs (see for example [32]). Unfortunately the Rfam database only provides the

consensus structure of a family and not individual sequences; thus it is not possible to reliably count the number of predicted basepairs which appear in the actual structure of an individual sequence and vice versa. To overcome this problem Bafna et al. used an alternative, *stack counting* measure [8] which is defined as the number of actual stacks and predicted stacks that overlap. As mentioned in [8] this measure is intended for comparing methods that explicitly extract stacks - which is not performed by most of the methods we compare.

We thus measure the predictive power of the programs we tested under the *structural edit distance* measure [42, 45]. which considers the differences between two RNA molecules in terms of both sequence/stack composition and structural elements. Given the *tree representation* of two RNA secondary structures, where each branch is labeled with a stack and every node represents a loop, their structural edit distance is defined to be the minimum possible sum of edit distances between the stack compositions of branch pairs and sequences of node pairs that are aligned to each other.

We computed the structural edit distances (SED) between the actual (consensus) structure of each of the 12 test families and the structure predictions by each test program via the `RNA_align` tool, publicly available on the web [76]. A distance of 0 corresponds to an identical sequence and structure, i.e. a perfect prediction. A higher distance value implies a poorer prediction.

The results of our comparative tests are summarized in the table below. (In addition, figure 3.1 demonstrates the outcome of `Densityfold` on the *E.coli* 5s_rRNA sequence (from RF00001 family) with that of `mfold/RNAfold`, `alifold` and `RNAscf`.) We used the default parameters in all programs we tested. We list the outcome of `Densityfold` for $\sigma = 1.5,\ 3.0$ and 5.0, and list the outcome of `MDensityfold` for the best possible σ value. As can be seen, `Densityfold` is at the top or near the top for most of the families. `Densityfold` with $\sigma = 5.0$ is always better than `Densityfold` with $\sigma = 3.0$. However `Densityfold` with $\sigma = 1.5$ outperforms both in a number of examples. Note that as σ approaches ∞ the outcome of `Densityfold` gets more and more similar to the outcome of `mfold/RNAfold`.[3] However `Densityfold` with

[3]In fact, we observed that for the families tested $\sigma = 100$ gives almost indistinguishable results

$\sigma = 5.0$ (the highest value we report) significantly outperforms `mfold/RNAfold` in a number of examples. Furthermore there is no clear winner between `Densityfold` and `MDensityfold`, each one outperforming the other in almost equal number of examples. However, in general, the longer the sequence gets, the better `MDensityfold` seemed to perform.

Name (Rfam_id)	Single sequence methods				Multiple sequence methods		
	`mfold/ RNAfold`	`Densityfold`			`MDensity_ fold`	`RNAscf`	`alifold`
		$\sigma = 1.5$	$\sigma = 3$	$\sigma = 5$			
5s_rRNA (RF00001)	149	84	89	89	92	134	122
Rhino_CRE (RF00220)	94	93	93	93	77	88	30
ctRNA_pGA1 (RF00236)	45	83	83	83	48	91	44
glmS (RF00234)	194	288	230	230	189	249	198
Hammerhead_3 (RF00008)	2	2	2	2	74	2	88
Intron_gpII (RF00029)	100	93	103	103	85	113	78
Lysine (RF00168)	182	256	194	186	178	131	173
Purine (RF00167)	64	103	103	103	133	56	141
Sam_riboswitch (RF00162)	124	129	129	99	110	133	121
Thiamine (RF00059)	156	170	179	149	187	179	149
tRNA (RF00005)	31	67	67	67	50	31	32
ykok (RF00380)	158	200	189	189	168	203	157

Table 3.1: Structural edit distances between the actual (consensus) structure of a family and the predicted structures by each one of the programs tested.

The results for the 12 RNA families clearly demonstrates that each RNA secondary structure prediction method performs quite well on certain RNA families while performing poorly on other RNA families. It is quite desirable to be able to identify certain characteristic of different prediction methods in terms of their prediction quality. It may be possible to improve the overall accuray of the RNA secondary structure prediction by determining which tool to use based on the RNA sequence. It is our aim to identify the characteristics of these prediction methods in terms of RNA sequences that they can predict uniqely and RNA equally well.

Our second test set is the same as the test set used by Do et al. [20] for testing the perdormance of their `CONTRAfold` program. Test set is composed of 151 RNA sequences from 151 unique Rfam [28] RNA families where the sequence that has

to that by `mfold/RNAfold`.

the best alignment to the consensus family secondary structure is selected. RNA secondary structures of all 151 RNA families are verified through physical methods.

Using this test set, we compared the performance of `Densityfold` with varying values of σ (which determines the contribution of the total energy to the optimization function) against `mfold/RNAfold`, the best pysics-based single sequence energy minimization program, `CONTRAfold` the best statistical learning single RNA secondary structure prediction method.

In order to identify the similarities and differences among these prediction methods we focused on the set of RNA sequences that can be predicted perfectly by any of these prediction methods. This resulting set is represented using a Venn diagram where each set represents one of the tested prediction methods and intersections represent the RNA sequences that are predicted correctly by more than one method. For a more accurate analysis we should consider the RNA sequences that are predicted almost perfectly by one the prediction methods. For a given RNA sequence X and a prediction method A, lets define $SED_A(X)$ as the structural edit distance between the known structure of X and the structure predicted by A. The secondary structure of an RNA molecule, X, is considered as almost perfect by a prediction method A, if for any other prediction method B, $\frac{SED_A(X)-SED_B(X)}{SED_A(X)+SED_B(X)}$ is above a threshold value which is selected as 0.5 for our tests.

The results of our comparative tests for single RNA structure prediction tools; `mfold`, `CONTRAfold` and `Densityfold`, are summarized in the figure below.

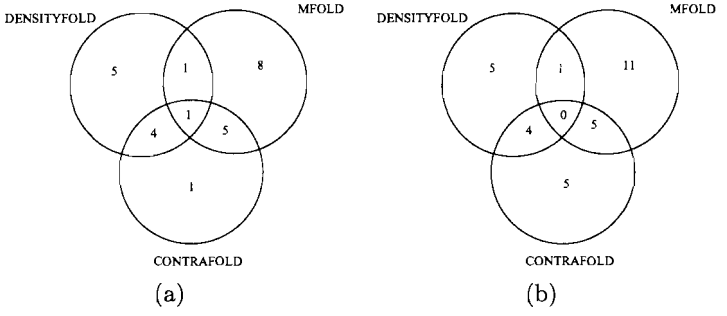


Figure 3.2: (a) RNA secondary structures predicted perfectly by using Densityfold, mfold and CONTRAfold. (b) RNA secondary structures predicted almost perfectly by using Densityfold, mfold and CONTRAfold.

In conclusion, `Densityfold` demonstrates that an energy density minimization objective is a valid alternative to the total energy minimization objective. It can be used both on a single sequence or on multiple sequences. Our goal for the future is to test non-linear combinations of energy density and total energy as well as non-linear normalizations of the free energy as objective functions; we hope that such variations can explain the better performance of `MDensityfold` over `Densityfold` on longer sequences.

Chapter 4

RNA-RNA Interaction Prediction

The second problem described in chapter 2 as an extension to RNA secondary structure prediction is the prediction of joint structure of two interacting RNAs which we call the general RNA-RNA Interaction Prediction (RIP) Problem. Given two RNA sequences S and R (e.g. an antisense RNA and its target), RIP problem asks to predict their joint secondary structure. A joint secondary structure between S and R is a set of "pairings" where each nucleotide of S and R is paired with at most one other nucleotide, either from S or R.

Interactions between nucleotides of two such RNA molecules can be established in the molecular level in two different ways. If the number of bases in the interaction is more than the length of one turn of a double helix (~10nt), a helical structure is formed as is the case in CopA/CopT interaction (see Figure 4.1, courtesy of Dr. Gerhart Wagner).

If the interaction is not long enough to form a stable double helix, the interacting bases on the sugar backbone of the interacting RNAs flip outside and a line up structure is formed as in Figure 4.2, similar to that in a pseudoknot on a single RNA molecule.

Let the i^{th} nucleotide of an RNA sequence S be denoted by $S[i]$ and the substring of S extending from $S[i]$ to $S[j]$ denoted by $S[i,j]$. As a notational convenience, let $S[k,k]$ denote $S[k]$, $S[i,i-1]$ denote an empty sequence and $S[i,i-1]^r$ denote the reverse of $S[i-1,i]$. In the rest of the definitions and algorithms, it is assumed that $S[1]$ denotes the $5'$ end of S and $R[1]$ denotes the $3'$ end of R.

Figure 4.1: The helical structure of the interaction between CopA and CopT pair. [38].

We compute the joint structure between S and R through minimizing their *total free energy*, which is, in general, a function of (stacked) pairs of bases as well as the topology of the joint structure.

Three models are considered for computing the free energy of the joint structure of interacting RNA sequences.

1. We first use the sum of free energies of individual WatsonCrick basepairs as a crude approximation to the total joint free energy. This *basepair* energy model is quite similar to that used by Nussinov and Jacobson [57] for predicting the structure of a single RNA molecule. Although the basepair energy model is known to be inaccurate, it provides a good starting point for further explorations.

2. Our second free energy model is based mostly on stacked pair energies given by Mathews et al [49], which provide the main contribution to the energy model employed by the `mfold` program for pseudoknot free single RNA structure prediction. Unfortunately, there is very little thermodynamic information on pseudoknots or kissing loops in the literature. Thus we employ the approach used by Rivas and Eddy [62] to differentiate the thermodynamic parameters

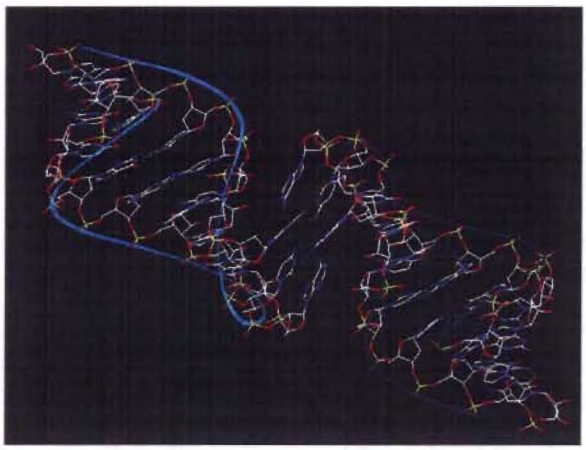


Figure 4.2: Establishment of interactions between bases of a short kissing loop pair at the molecular level [34].

of "external" bonds from "internal" bonds by by multiplying the external parameters with a *weight* slightly smaller than 1. This *stacked pair* energy model turns out to be quite accurate, especially in predicting the joint structure of shorter ($\leq$ 150 bases) RNA molecule pairs.

3. The final energy model enriches the above models by summing up the free energies of various types of internal loops and stacked pairs as per [79, 49] as well as the weighted free energies of externally interacting ("kissing") loops. This model, which will be referred to as the *loop* energy model, appears to be more accurate especially for longer ($\geq$ 150 bases) RNA molecules.

Although we allow arbitrary loops to form kissing pairs, we impose the following constraints on the topology of a joint structure between RNA sequences. First, a joint structure can have no *internal pseudoknots*; i.e., if $S[i]$ bonds with $S[j]$ then no $S[i']$ for $i < i' < j$ can bond with any $S[j']$ for $j < j'$. The same property will be satisfied by the nucleotides of R as well. Second, a joint structure can not have any *external pseudoknots*; i.e., if $S[i]$ bonds with $R[j]$ then no $S[i']$ for $i' > i$ can bond with any $R[j']$ for $j' < j$.

These assumptions are satisfied by all examples of complex RNA-RNA interactions encountered in the literature search. Furthermore allowing arbitrary pseudoknots in the secondary structure of even a single RNA molecule makes the energy minimization problem NP-hard [2]. In fact we proved in in Section 4.1 that the RIP problem is NP-hard for each one of the energy models, even when no internal or external pseudoknots are allowed. This necessitates the addition of one more natural constraint on the topology of the joint secondary structure prediction, which is again satisfied by all known joint structures in the literature. Under this constraint it is then shown how to obtain efficient algorithms to minimize the free energy of the joint structure under all three energy models and testing the accuracy of the algorithms on known joint structures are presented. Finally the structure prediction techniques are applied to search for target mRNA sequences to any given small RNA molecule in whole genomic or plasmid sequences.

4.1 RIP problem for Both Basepair and Stacked Pair Energy Models is NP-Complete

First NP-Completeness of the RIP problem under both the basepair and the stacked pair energy models will be proven.

Theorem 1 *RIP problem under the Basepair Energy Model is NP-Complete.*

Proof: The NP-Completeness of RIP is established through a reduction from the longest common subsequence of multiple binary strings (mLCS) which is a known NP-Complete problem. This proof is an extension to the one in [2] for the single RNA secondary structure prediction problem with pseudoknots.

The decision version of the mLCS problem is as follows: Given a set of *binary* strings $L = \{S_1, S_2, \cdots, S_m\}, (|S_1| = \cdots = |S_m| = n)$ and an integer k, decide whether there exists a sequence C of length k which is a subsequence of each S_i. Here we assume that m is an odd number; if it is even, one can simply add a new string $S_{m+1} = S_m$ to L.

From an instance of mLCS, first construct two "RNA" sequences S and R, using an extended nucleotide alphabet $\Sigma^e = \{a, b, c, d, e, f, u, w, x, y, z\}$. (The NP-hardness proof for the -more interesting- stacked pair energy model below uses the standard RNA nucleotide alphabet $\{A, C, G, U\}$.)

Let v^j denote the string formed by concatenating j copies of character v and let $\overline{v}$ denote the *complementary residue* of v. In our extended alphabet, we set $\overline{x} = w$, $\overline{y} = z$, $\overline{a} = b$, $\overline{c} = d$, and $\overline{e} = f$. Given a string T, its *reverse complement* is denoted by $\overline{T}$.

For $i = 1, \cdots, m$, construct strings D_i and E_i as follows. Note that we set $s_{i,j}$ to x if the j^{th} character of string S_i is 0; if it is 1, $s_{i,j}$ is set to be y.

$$
\begin{aligned}
D_i &= a\ s_{i,1}\ a\ s_{i,2}\ a\ \cdots\ a\ s_{i,n}\ a, && \textit{if } i \textit{ is odd;}\\
D_i &= a\ \overline{s_{i,n}}\ a\ \overline{s_{i,n-1}}\ a\ \cdots\ a\ \overline{s_{i,1}}\ a, && \textit{if } i \textit{ is even;}\\
E_i &= b\ \overline{s_{i,1}}\ b\ \overline{s_{i,2}}\ b\ \cdots\ b\ \overline{s_{i,n}}\ b, && \textit{if } i \textit{ is odd;}\\
E_i &= b\ s_{i,n}\ b\ s_{i,n-1}\ b\ \cdots\ b\ s_{i,1}\ b, && \textit{if } i \textit{ is even.}
\end{aligned}
$$

Now we construct the RNA sequences S and R as follows.

$$
\begin{aligned}
S &= u^k, D_1, c^1, D_2, D_3, d^1, c^2, D_4, D_5, d^2 \cdots c^{(m-1)/2}, D_{m-1}, D_m, d^{(m-1)/2}\\
R &= e^1, E_1, E_2, f^1, e^2, E_3, E_4, f^2 \cdots e^{(m-1)/2}, E_{m-2}, E_{m-1}, f^{(m-1)/2}, E_m, u^k
\end{aligned}
$$

Note that the lengths of S and R are polynomial with the total size of all sequences $S_1 \ldots S_m$.

Now we set the energy function for bonded nucleotides pairs. The bond between each nucleotide with its complement has a free energy of -1.0. The bond between u with x, y, z, w also has a free energy of -1.0. For other bonds between nucleotide pairs, the free energy is 0.0.

In the basepair energy model, the free energy of the overall structure is defined to be the sum of the free energies of all bonded pairs of nucleotides. Thus, according to the above setting, each nucleotide other than u will tend to get bonded with their complementary nucleotides, and u will tend to get bonded with any of x, y, z, w and vice versa. Such bondings are called *valid* bondings. The free energy of the joint

structure is minimized when the number of valid bondings between nucleotide pairs is maximized.

Now it will be shown that there exists a common subsequence of length k among $S_1, \ldots, S_m$ if and only if there exists a joint secondary structure of S and R where *every* nucleotide forms a valid bonding. Suppose that $S_1 \ldots S_m$ have a common subsequence C of length k; one can construct a secondary structure of S and R where every nucleotide forms a valid bonding as follows.

- For each i, form a bond between the i^{th} a in S with the i^{th} b in R.
- For each i, bond the substring c^i to the substring d^i in S and bond the substring e^i to the substring f^i in R.
- For each string $S_i \in L$ there is a corresponding substring D_i in S and E_i (which is the complement of D_i) in R. Consider for each S_i the sequence that remains when the common subsequence C is deleted out; denote this sequence by C'. Bond each nucleotide in D_i that corresponds to a character in C' to its corresponding complementary nucleotide in E_i.
- All that remains in S and R are those nucleotides that correspond to the common subsequence C in each string S_i. There is also the substring u^k at the left end of S and another substring of the form u^k at the right end of R. Bond the u^k block in S to the unbonded nucleotides (that correspond to C) in D_1. For all $1 \leq i \leq (m-1)/2$, bond the unbonded nucleotides in E_{2i-1} to those in E_{2i}. Similarly bond the unbonded nucleotides in D_{2i} to those in D_{2i+1}. Finally bond the unbonded nucleotides in E_m to the u^k block in R.

The reader can easily verify that this construction establishes a valid bonding for all nucleotides in S and R. The process of constructing S and R and establishing the bonds described above is demonstrated in Figure 4.3. Here $L = \{s_1 = xyxx,\ s_2 = xxyx,\ s_3 = xyyx\}$.

Now we show that if there is a joint secondary structure between S and R where every nucleotide forms a valid bonding, then there is a common subsequence of strings $S_1, S_2, \cdots, S_m$ of length k.

- Nucleotides a and b are complementary and do not form bonds with u. S only has as and R only has bs. If all as and bs form valid bonds, the i^{th} a *must* form a bond with the i^{th} b.

- Nucleotides c, d only occur in S and only form valid bonds with each other. Because allow internal pseudoknots are not allowed, each c^i block will be bonded with the d^i block. Similarly, nucleotides e, f only occur in R and only form valid bonds with each other. Again, because there are no internal pseudoknots, each e^i block will be bonded with the f^i block.

- The above bondings necessitate that nucleotides of the u^k block in S must bond with those in D_1 and nucleotides of the u^k block in R must bond with those in E_m. The remaining nucleotides of D_1 must bond with corresponding nucleotides in E_1 and the remaining nucleotides of E_m must bond with corresponding nucleotides in D_m.

- The nucleotides that are left in E_1 are the nucleotides that correspond to those in D_1 which have been bonded to u^k block - they must be bonded to complementary nucleotides in E_2. The bonds between E_1 and E_2 corresponds to a common subsequence of S_1 and S_2 of size k.

- Inductively, for $i = 1 \ldots (m-1)/2$, the nucleotides left out in E_{2i} must form bonds with corresponding nucleotides in D_{2i}. The ones that are left out in D_{2i} must form bonds with complementary nucleotides in D_{2i+1}. The bonds between D_{2i} and D_{2i+1} corresponds to a common subsequence of S_{2i} and S_{2i+1}.

- Similarly, the nucleotides left out in D_{2i+1} must form bonds with corresponding nucleotides in E_{2i+1}. The ones that are left out in E_{2i+1} must form bonds with complementary nucleotides in E_{2i+2}. The bonds between E_{2i+1} and E_{2i+2} corresponds to a common subsequence of S_{2i+1} and S_{2i+2}.

- Finally, the nucleotides that are left out in E_m must be bonded to nucleotides in u^k block in R.

The bonds between consecutive D_i, D_{i+1} pairs and E_i, E_{i+1} pairs correspond to common subsequences between S_i and S_{i+1}. Thus the strings $S_1, \ldots, S_m$ must have a common subsequence of length k. ∎

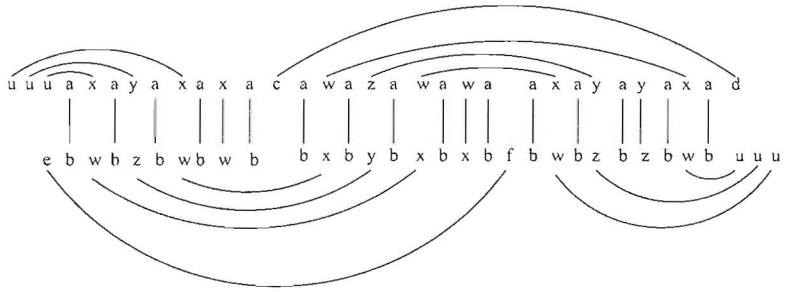


Figure 4.3: Sample RIP solution for mLCS problem on $S_1 = \{xyxx\}, S_2 = \{xxyx\}, S_3 = \{xyyx\}$. The mLCS is determined with the internal bondings, here it is xyx.

Now we established the NP-hardness of the RIP problem under the stacked pair energy model.

Theorem 2 *RIP problem under the Stacked Pair Energy Model is NP-Complete.*

Proof: The proof is through an indirect reduction from the mLCS problem as per Theorem 1. Consider the reduction of the mLCS problem to the RIP problem under the basepair energy model. Given sequences S and R that were obtained as a result of this reduction, it is possible to construct two new RNA sequences S' and R' from the standard nucleotide alphabet by replacing each character in S and R with quadruples of nucleotides as follows: $a \leftarrow CCGU$, $b \leftarrow GGCU$, $c \leftarrow GCCU$, $d \leftarrow CGGU$, $e \leftarrow CGCU$, $f \leftarrow GCGU$, $u \leftarrow AAAU$, $x \leftarrow ACAU$, $z \leftarrow CACU$, $y \leftarrow AGAU$, $w \leftarrow GAGU$.

The energy function for stacked pairs of nucleotides will be determined as follows. The free energy of the following stacked pairs are all set to -0.5:
$(A-A, A-C), (A-A, C-A), (A-A, A-G), (A-A, G-A), (A-C, A-A), (A-C, C-A), (A-G, A-A), (A-G, G-A), (C-A, A-A), (C-A, A-C), (C-G, C-G), (C-G, G-C), (G-A, A-A), (G-A, A-G), (G-C, C-G), (G-C, G-C)$.

For other bondings between nucleotides, the free energy is set to 0.0. Thus bonding U with any nucleotide will not reduce the free energy of the joint structure.

In the stacked pair energy model, the free energy of the overall structure is defined to be the sum of the free energies of all stacked pairs of bonded nucleotides. The reader can verify that above setting of stacked pair energies ensure that the bonds between the characters of S and R presented in Theorem 1 will be preserved between S' and R'. (e.g. a bond between a and b has free energy -1.0. Because a corresponds to $CCGU$ and b corresponds to $GGCU$, the stacked pairs obtained will be $(C-G, C-G)$ and $(C-G, G-C)$ each with free energy -0.5. The total free energy will thus be -1.0.) ∎

4.1.1 Additional topological constraints on joint structures

The hardness of the RIP problem under both basepair and stacked pair energy models necessitate one more constraint on the topology of the interaction between two RNA molecules. Based on our observations of known joint structures of RNA molecule pairs in Figure 2.2, the following constraint (which is satisfied by all known structures in the literature) is imposed. Let $S[i]$ be bonded with $S[j]$ and $R[i']$ be bonded with $R[j']$. Then exactly one of the following must be satisfied:

1. There are no $i < k < j$ and $i' < k' < j'$ such that $S[k]$ bonds with $R[k']$.
2. For all $i < k < j$, if $S[k]$ bonds with some $R[k']$ then $i' < k' < j'$.
3. For all $i' < k' < j'$, if $R[k']$ bonds with some $S[k]$ then $i < k < j$.

The condition simply states that if two "substructures" $S[i, j]$ and $R[i', j']$ interact, then one must "subsume" the other. A joint structure of two RNA sequences S and R is considered to be *valid* if all above conditions are satisfied.

4.2 Structure prediction in the Basepair Energy Model

The basepair energy model approximates the free energy of the joint structure between interacting RNA molecules as the sum of the free energies of bonded nucleotide pairs. The Watson-Crick free energy of a bond between nucleotides x and y is denoted by $e(x, y)$ if they are on the same RNA strand (this is called an *internal bond*) and by $e'(x, y)$ if they are on different strands (this is called an *external bond*). Although in our experiments $e' = e$ is preset, this formulation also allows to differentiate these two energy functions. Below, we obtain a valid pairing between the nucleotides of S and R that minimizes the free energy of their joint structure through the computation of $E(S[i, j], R[i', j'])$ the free energy between interacting RNA strands $S[i, j]$ and $R[i', j']$ for all $i < j$ and $i' < k'$. Clearly E gives the overall free energy between S and R when $i = i' = 1$ and $j = |S|$ and $j' = |R|$. $E(S[i, i], R[i', i'])$ is set to $e'(S[i], R[i'])$ and the value of $E(S[i, j], R[i', j'])$ is computed inductively as the minimum of the following:

1. $\min_{i-1 \le k \le j; i'-1 \le k' \le j'' : (k \ne i-1 \text{ or } k' \ne i'-1), (k \ne j \text{ or } k' \ne j')} E(S[i, k], R[i', k']) + E(S[k+1, j]), R[k'+1, j'])$.
2. $E(S[i+1, j-1], R[i', j']) + e(S[i], S[j])$.
3. $E(S[i, j], R[i'+1, j'-1]) + e(R[i'], R[j'])$.

The above dynamic programming formulation will return the optimal structure by considering the following two cases:

1. Consider the case that either $S[i]$ or $S[j]$ or $R[i']$ or $R[j']$ bonds with a nucleotide on the other RNA strand. Wlog, let $S[i]$ bond with $R[h']$; then either (i) $R[i']$ bonds with $R[j']$ for which condition (3) will be satisfied, or (ii) $i' = h'$ so that $R[i']$ bonds with $S[i]$ for which condition (1) will be satisfied for $k = i$ and $k' = i'$, or (iii) $R[i']$ bonds with some $R[\ell']$ for which condition (1) will be satisfied for some "break-point" $S[k]$, $R[k']$, for $i \le k \le j$ and $i' \le k' \le j'$ such that $S[i, k]$ interacts only with $R[i', k']$ and $S[k+1, j]$ interacts only with $R[k'+1, j']$.

2. If the above condition is not satisfied then wlog one can assume that $S[i]$ bonds with $S[h]$ and $R[i']$ bonds with $R[h']$. If for no $\ell > h$, $S[\ell]$ interacts with any $R[\ell']$ for $\ell' > h'$ then condition (1) will be satisfied with $k = h$ and either $k' = i' - 1$ or $k' = j' + 1$. If for no $\ell < h$, $S[\ell]$ interacts with any $R[\ell']$ for $\ell' < h'$ then condition (1) will be satisfied again with $k = h$ and either $k' = i' - 1$ or $k' = j' + 1$. The possibility of none of these two cases hold is excluded by the topological constraints described earlier.

Table E is a four dimensional table $E[i, i', j, j]$ where $i, i' \in \{1 \cdots |S|\}$ and $j, j' \in \{1 \cdots |R|\}$, requiring space $O(|S|^2 \cdot |R|^2)$. Step 1 in the dynamic programming formulation partitions the table E around breakpoints $k \in \{1 \cdots |S|\}$ and $k' \in \{1 \cdots |R|\}$ and recurses around these points, making the run time $O(|S|^3 \cdot |R|^3)$.

4.2.1 Testing the Basepair Energy Model

We tested the basepair energy model on naturally occurring joint structures of interacting RNA molecule pairs CopA-CopT and OxyS-fhlA. The results are given in Figures 4.4 and 4.5. Perhaps not surprisingly, the predicted joint structures by

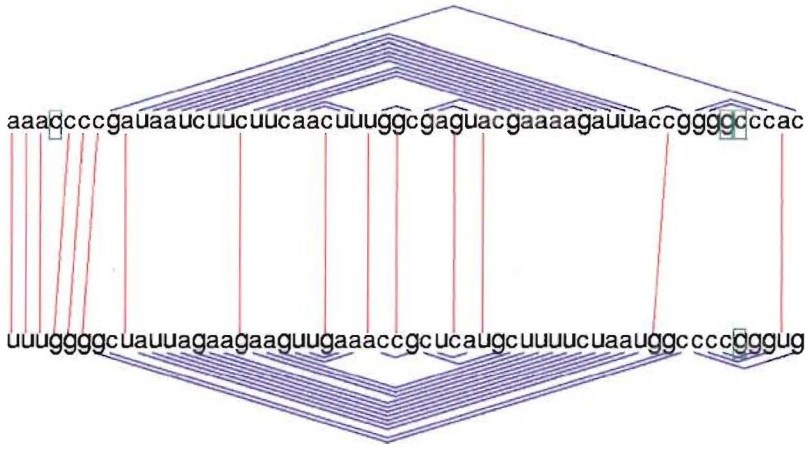


Figure 4.4: Joint structure of CopA and CopT as predicted by Basepair Energy Model.

the Basepair Energy Model is quite different from the natural secondary structures

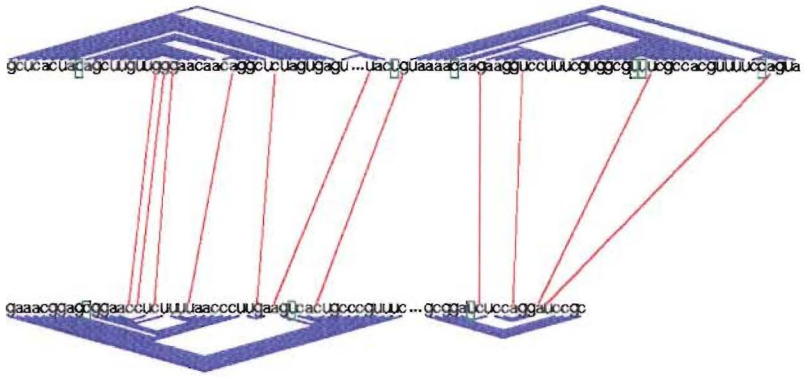


Figure 4.5: Joint structure of OxyS and fhlA as predicted by Basepair Energy Model.

(Figures 2.4 and 2.5). Observe that in natural joint structures, internal or external bonds usually form stacked pairs; i.e., a bond $S[i] - S[j]$ usually implies bonds $S[i+1] - S[j-1]$ and $S[i-1] - S[j+1]$. Similarly a bond $S[i] - R[i']$ usually implies bonds $S[i+1] - R[i'+1]$ and $S[i-1] - R[i'-1]$. Furthermore, in natural joint structures unbonded nucleotides seem to form uninterrupted sequences rather than being scattered around.

4.3 Structure Prediction Based on Stacked Pair Energy Model

The limitations of the Basepair Energy Model promotes the use of a Stacked Pair Energy Model where the bonds between nucleotide pairs form uninterrupted sequences. Let $ee(X[i, i+1], X[j-1, j])$ denote the energy of the internal stacked pair $(X[i] - X[j], X[i+1] - X[j-1])$ and $ee'(X[i, i+1], Y[j, j+1])$ denote the energy of the external stacked pair $(X[i] - Y[j], X[i+1] - Y[j+1])$. As per the *pknots* approach [62] one can set $ee' = \sigma \cdot ee$ for a user defined weight parameter $0 < \sigma \leq 1$ (externally kissing pairs are similar in nature to pseudoknots). The thermodynamic free energy parameters that are used in our tests are taken from [49], and listed on Table 4.3. Note that the energy functions $ee(.,.)$ and $ee'(.,.)$ are not symmetric; they can differ according to the relative directions of the stacked pairs ($3' - 5'$ or

$5' - 3'$) involved.

	A-U	C-G	G-C	G-U	U-A	U-G	$3'$
A-U	-0.9	-2.2	-2.1	-0.6	-1.1	-1.4	
C-G	-2.1	-3.3	-2.4	-1.4	-2.1	-2.1	
G-C	-2.4	-3.4	-3.3	-1.5	-2.2	-2.5	
G-U	-1.3	-2.5	-2.1	-0.5	-1.4	1.3	
U-A	-1.3	-2.4	-2.1	-1.0	-0.9	-1.3	
U-G	-1.0	-1.5	-1.4	0.3	-0.6	-0.5	
$5'$							

Table 4.1: Free energy parameters for stacking pairs used in Stacked Pair Energy Model as given in [49].

To compute the joint structure between S and R under the Stacked Pair Energy Model we introduce four energy functions.

1. $E_S(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $S[i]$ bonds with $S[j]$.

2. $E_R(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $R[i']$ bonds with $R[j']$.

3. $E_l(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $S[i]$ bonds with $R[i']$.

4. $E_r(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $S[j]$ bonds with $R[j']$.

The complete dynamic programming formulation is given in Table 4.2, and the initial settings of the energy functions E_S, E_R, E_r, E_l are listed in Table 4.3. Note that because sequence R is assumed to be in $3' - 5'$ direction, reversing the stacked pairs involved is required for the correct use of *ee* function in E_R.

The dynamic programming formulation is an extension to the algorithm for Base-pair Energy Model, thus it obeys the same constraints on joint structures as described above. Tables E_S, E_R, E_l, E_r, E are each four dimensional tables $E[i,i',j,j]$ where $i, i' \in \{1 \cdots |S|\}$ and $j, j' \in \{1 \cdots |R|\}$, requiring space $O(|S|^2 \cdot |R|^2)$. The

$$E(S[i,j],R[i',j']) = \min \left\{ \begin{array}{l} E_S(S[i,j],R[i',j']), E_R(S[i,j],R[i',j']), \\ E_r(S[i,j],R[i',j']), E_l(S[i,j],R[i',j']), \\ \min_{i\le k\le j-1;i'\le k'\le j'-1} \left\{ \begin{array}{l} E(S[i,k],R[i',k'])+ \\ E(S[k+1,j]),R[k'+1,j']) \end{array} \right\} \\ \min_{i\le k\le j-1} \left\{ \begin{array}{l} E(S[i,k],-)+ \\ E(S[k+1,j]),R[i',j']) \end{array} \right\} \\ \min_{i\le k\le j-1} \left\{ \begin{array}{l} E(S[i,k],R[i',j'])+ \\ E(S[k+1,j]),-) \end{array} \right\} \\ \min_{i'\le k'\le j'-1} \left\{ \begin{array}{l} E(S[i,j],R[i',k'])+ \\ E(-,R[k'+1,j']) \end{array} \right\} \\ \min_{i'\le k'\le j'-1} \left\{ \begin{array}{l} E(-,R[i',k'])+ \\ E(S[i,j]),R[k'+1,j']) \end{array} \right\} \end{array} \right\}$$

$$E_l(S[i,j],R[i',j']) = \min \left\{ \begin{array}{c} E_l(S[i+1,j],R[i'+1,j']) + ee'(S[i,i+1],R[i',i'+1]), \\ E(S[i+1,j],R[i'+1,j']) \end{array} \right\}$$

$$E_r(S[i,j],R[i',j']) = \min \left\{ \begin{array}{c} E_r(S[i,j-1],R[i',j'-1]) + ee'(S[j-1,j],R[j'-1,j']), \\ E(S[i,j-1],R[i',j'-1]) \end{array} \right\}$$

$$E_S(S[i,j],R[i',j']) = \min \left\{ \begin{array}{c} E_S(S[i+1,j-1],R[i',j']) + ee(S[i,i+1],S[j-1,j]), \\ E(S[i+1,j-1],R[i',j']) \end{array} \right\}$$

$$E_R(S[i,j],R[i',j']) = \min \left\{ \begin{array}{c} E_R(S[i,j],R[i'+1,j'-1]) + ee(R[j'-1,j']^r,R[i',i'+1]^r), \\ E(S[i,j],R[i'+1,j'-1]) \end{array} \right\}$$

Table 4.2: Complete description of the dynamic programming algorithm for Stacked Pair Energy Model.

dynamic programming formulation also partitions the overall energy table E around breakpoints $k \in \{1 \cdots |S|\}$ and $k' \in \{1 \cdots |R|\}$ and recurses around these points, making the run time $O(|S|^3 \cdot |R|^3)$.

4.3.1 Testing Stacked Pair Energy Model

The Stacked Pair Energy Model as defined above has only one user defined parameter (as per [62]), σ, which is the ratio between the free energies of internal and external

$$\begin{array}{ll} E_l(S[i,j],-) = \infty & E_r(S[i,j],-) = \infty \\ E_l(-,R[i',j']) = \infty & E_r(-,R[i',j']) = \infty \\ E_l(S[i,i],R[i',i']) = 0 & E_r(S[i,i],R[i',i']) = 0 \\ E_S(S[i,i],-) = \infty & E_R(-,R[i',i']) = \infty \end{array}$$

Table 4.3: Initial settings of the energy functions for Stacked Pair Energy Model.

stacked pairs. Unfortunately no miracle prescription for determining the right value of σ is available (see for example [62]). It is possible to approximately determine the value for σ by closely inspecting the natural joint structure of CopA-CopT pair (Figure 2.4). CopA and CopT sequences are perfectly complimentary to each other, thus they can, in principle, form a stable duplex structure that would prevent any internal bonding pairs. However, as one can observe from Figure 2.4 this does not happen. The ratio between the length of the external bonding sequences in the joint structure and that of the internal bonding sequences implies that $\sigma \in [0.7, 0.8]$.

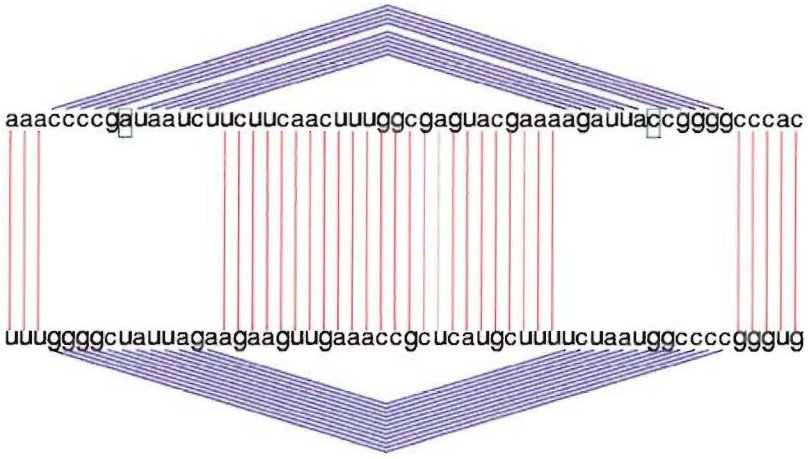


Figure 4.6: Joint structure of CopA and CopT as predicted by Stacked Pair Energy Model.

Under these observations we tested our algorithm that implements the Stacked Pair Energy Model is tested with $\sigma \in [0.7, 0.8]$. The secondary structures predicted by our algorithm on CopA-CopT and OxyS-fhlA pairs are given in Figures 4.6 and 4.7 respectively. As one can observe, there are only very slight differences between the natural joint structure and the predicted joint structure of the RNA pairs. For example, the predicted joint structure of OxyS-fhlA pair (Figure 4.7) has 53 internal-bonds, 14 external-bonds, and 23 unbonded nucleotides. In *all* aspects, these figures are superior to the natural joint structure of the pair (Figure 2.5), which has 50 internal-bonds, 16 external-bonds, and 25 unbonded nucleotides. Because the external bond scores are smaller than internal ones, under *any selection of* $\sigma < 1$

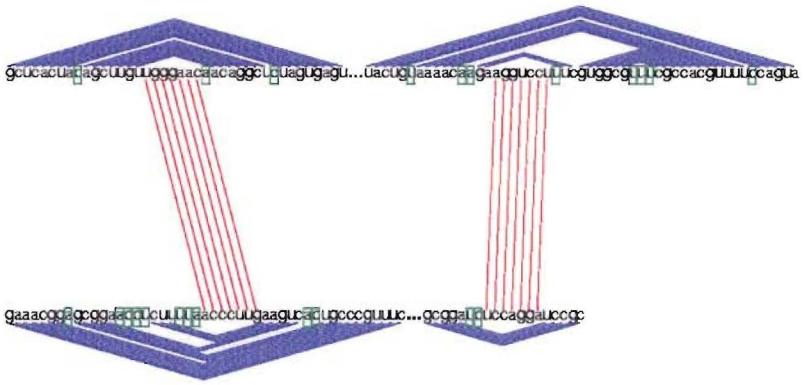


Figure 4.7: Joint structure of OxyS and fhlA as predicted by Stacked Pair Energy Model.

the prediction of this algorithm results in a higher score/lower free energy than that implied by the natural joint structure of OxyS-fhlA pair. Nevertheless, the differences between the natural structures and the predicted ones are very small implying that the Stacked Pair Energy Model can be used as the central tool of the RNA target prediction algorithm.

4.3.2 A More General Stacked Pair Energy Formulation

As will be discussed below, the Stacked Pair Energy Model formulation works very well with the joint structure prediction problems considered in this dissertation. However this formulation does not necessarily aim to cluster gaps in uninterrupted sequences, as observed in natural joint structures. Thus, a more general formulation is also provided for the Stacked Pair Energy Model, that employs an "affine" cost model for the gaps involved. Also considered in this formulation are penalties for switching from internal to external bonds (and vice versa). This general formulation does not necessarily improve the predictions for the joint structures considered above; however it could be useful for other examples and thus provided below.

The more general formulation of the Stacked Pair Energy Model adds two more energy functions e and e', and two penalty parameters g and G. This necessitates the addition of four more energy tables $E_{S,l}, E_{S,r}, E_{R,l}, E_{R,r}$ to the set (E_S, E_R, E_l, E_r)

already used in Section 4.3:

1. $E_{S,l}(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $S[i]$ remains unbonded.

2. $E_{S,r}(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $S[j]$ remains unbonded.

3. $E_{R,l}(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $R[i']$ remains unbonded.

4. $E_{R,r}(S[i,j], R[i',j'])$ denotes the free energy between S and R such that $R[j']$ remains unbonded.

The addition of four more parameters (and four new degrees of freedom) makes this approach more adjustable to specific properties of the input RNA strands.

In addition to the stacked pair energies, this formulation also considers the free energies of an internally and externally bonded individual nucleotide pairs denoted $e(X[i], Y[j])$ and $e'(X[i], X[j])$ respectively. For further generality, this formulation induces an additive penalty for switching between the two types of bonds. More specifically, the energy function has an additive penalty g to any nucleotide $X[k]$ (X could be S or R), if (i) $X[k]$ is bonded with $X[j]$ however $X[k+1]$ is not bonded with $X[j-1]$, (ii) $X[k]$ is bonded with $X[j]$ however $X[k-1]$ is not bonded with $X[j+1]$, (iii) $X[k]$ is bonded with $Y[k']$ however $X[k+1]$ is not bonded with $Y[k'+1]$, (iv) $X[k]$ is bonded with $Y[k']$ however $X[k-1]$ is not bonded with $Y[k'-1]$. For unbonded nucleotides $X[k]$ another additive penalty G is charged if (i) $X[k+1]$ is bonded, (ii) $X[k-1]$ is bonded. The gap penalties are also added to the first and last nucleotides of X - this is only for avoiding further complexity in the dynamic programming formulation and does not affect the energy minimization process or the resulting structure prediction.

This more general energy formulation is given in Table 4.4, and the initializations are in Table 4.5.

$$E(S[i,j],R[i',j']) = \min \left\{ \begin{array}{l} E_S(S[i,j],R[i',j'])+2g, \quad E_R(S[i,j],R[i',j'])+2g, \\ E_r(S[i,j],R[i',j'])+2g, \quad E_l(S[i,j],R[i',j'])+2g, \\ E_{S,l}(S[i,j],R[i',j'])+G, \quad E_{S,r}(S[i,j],R[i',j'])+G, \\ E_{R,l}(S[i,j],R[i',j'])+G, \quad E_{R,r}(S[i,j],R[i',j'])+G, \\ \min_{i\le k\le j-1;i'\le k'\le j'-1} \left\{ \begin{array}{l} E(S[i,k],R[i',k'])+ \\ E(S[k+1,j]),R[k'+1,j']) \end{array} \right\} \\ \min_{i\le k\le j-1} \left\{ \begin{array}{l} E(S[i,k],-)+ \\ E(S[k+1,j]),R[i',j']) \end{array} \right\} \\ \min_{i\le k\le j-1} \left\{ \begin{array}{l} E(S[i,k],R[i',j'])+ \\ E(S[k+1,j]),-) \end{array} \right\} \\ \min_{i'\le k'\le j'-1} \left\{ \begin{array}{l} E(S[i,j],R[i',k'])+ \\ E(-,R[k'+1,j']) \end{array} \right\} \\ \min_{i'\le k'\le j'-1} \left\{ \begin{array}{l} E(-,R[i',k'])+ \\ E(S[i,j]),R[k'+1,j']) \end{array} \right\} \end{array} \right\}$$

$$E_l(S[i,j],R[i',j']) = e'(S[i],R[i'])+ \min \left\{ \begin{array}{c} E_l(S[i+1,j],R[i'+1,j'])+ee'(S[i,i+1],R[i',i'+1]), \\ E(S[i+1,j],R[i'+1,j'])+2g \end{array} \right\}$$

$$E_r(S[i,j],R[i',j']) = e'(S[j],R[j'])+ \min \left\{ \begin{array}{c} E_r(S[i,j-1],R[i',j'-1])+ee'(S[j-1,j],R[j'-1,j']), \\ E(S[i,j-1],R[i',j'-1])+2g \end{array} \right\}$$

$$E_S(S[i,j],R[i',j']) = e(S[i],S[j])+ \min \left\{ \begin{array}{c} E_S(S[i+1,j-1],R[i',j'])+ee(S[i,i+1],S[j-1,j]), \\ E(S[i+1,j-1],R[i',j'])+2g \end{array} \right\}$$

$$E_R(S[i,j],R[i',j']) = e(R[i'],R[j'])+ \min \left\{ \begin{array}{c} E_R(S[i,j],R[i'+1,j'-1])+ee(R[j'-1,j']^r,R[i',i'+1]^r), \\ E(S[i,j],R[i'+1,j'-1])+2g \end{array} \right\}$$

$$E_{S,l}(S[i,j],R[i',j']) = \min \left\{ \begin{array}{l} E_{S,l}(S[i+1,j],R[i',j'])+e(S[i],-), \\ E(S[i+1,j],R[i',j'])+e(S[i],-)+G \end{array} \right\}$$

$$E_{S,r}(S[i,j],R[i',j']) = \min \left\{ \begin{array}{l} E_{S,r}(S[i,j-1],R[i',j'])+e(-,S[j]), \\ E(S[i,j-1],R[i',j'])+e(-,S[j])+G \end{array} \right\}$$

$$E_{R,l}(S[i,j],R[i',j']) = \min \left\{ \begin{array}{l} E_{R,l}(S[i,j],R[i'+1,j'])+e(R[i'],-), \\ E(S[i,j],R[i'+1,j'])+e(R[i'],-)+G \end{array} \right\}$$

$$E_{R,r}(S[i,j],R[i',j']) = \min \left\{ \begin{array}{l} E_{S,r}(S[i,j],R[i',j'-1])+e(-,R[j']), \\ E(S[i,j],R[i',j'-1])+e(-,R[j'])+G \end{array} \right\}$$

Table 4.4: The full dynamic programming algorithm for Stacked Pair Energy Model.

$E_l(S[i,j],-) = \infty$	$E_r(S[i,j],-) = \infty$
$E_l(-,R[i',j']) = \infty$	$E_r(-,R[i',j']) = \infty$
$E_l(S[i,i],R[i',i']) = e'(S[i],R[i']) + 2g$	$E_r(S[i,i],R[i',i']) = e'(S[i],R[i']) + 2g$
$E_S(S[i,i],-) = \infty$	$E_R(-,R[i',i']) = \infty$
$E_{S,l}(S[i,i],-) = e(S[i],-) + G$	$E_{R,l}(-,R[i',i']) = e(-,R[i']) + G$
$E_{S,r}(S[i,i],-) = e(S[i],-) + G$	$E_{R,r}(-,R[i',i']) = e(-,R[i']) + G$

Table 4.5: The initializations for full dynamic programming algorithm for Stacked Pair Energy Model.

4.4 Structure Prediction Based on Loop Energy Model

The structure prediction algorithm to find the optimal joint structure between two RNA molecules based on Stacked Pair Energy Model requires substantial resources in terms of running time and memory. On a Sun Fire v20z server with 16GB RAM and AMD Opteron 2.2GHz processor, the running time for predicting the joint secondary structure of OxyS-fhlA pair is 15 minutes; this could be prohibitive for predicting the targets of sufficiently long RNA molecules. Here we describe a number of observations on the natural joint structures of RNA molecule pairs for speeding up the previous approach through heuristic shortcuts - without losing its (experimental) predictive power.

An interesting observation is that the (predicted) self structures are mostly preserved in the joint secondary structures. In fact, external interactions only occur between pairs of predicted hairpins. Thus it may be be sufficient to compute the joint structure of two RNA sequences by simply computing the set of loop pairs that form bonds to minimize the total joint free energy.

The above observation prompts an alternative, simpler approach which is described below. This new approach maintains that each RNA sequence will tend to preserve much of its original secondary structure after interacting with the other RNA sequence, which is achieved by means of preserving what we call "independent subsequences" that form hairpins. More formally:

Definition 1 *Independent Subsequences:*

Given an RNA sequence R and its secondary structure, the substring $R(i,j)$ is an independent subsequence of R if it satisfies the following conditions.

- $R[i]$ is bonded with $R[j]$.
- $j - i \leq \kappa$ for some user specified length κ.
- There exists no $i' < i$ and $j' > j$ such that $R[i']$ is bonded with $R[j']$ and $j' - i' \leq \kappa$. (This condition prohibits overlaps between independent subsequences).

It is possible to compute the (locations of) independent sequences of a given RNA molecule, from its secondary structure predicted by `mfold`, through a simple greedy algorithm as follows.

1. Let IS be the set of independent subsequences in R; initially set $IS = \emptyset$.
2. Starting from the first nucleotide of R find the first nucleotide $R[i]$ which bonds with another nucleotide $R[j]$, $(j > i)$.
3. If $j - i \leq \kappa$ then update $IS = IS \cup R[i,j]$ and move to $R[j+1]$.
   Else move to $R[i+1]$.
4. Repeat Steps 2 and 3.

The preprocessing step of computing single RNA folding assumes that there are no pseudoknots in the RNA molecule. Thus, for any pair i, j, when $R[i]$ bonds with $R[j]$ all the bases between $R[i]$ and $R[j]$ must either form no bonds or form bonds with bases in the same subsequence. Such a subsequence has no interactions with the rest of the RNA that it lies on, making $R[i,j]$ an independent subsequence. Step 2 in the above formulation captures subsequences where start and end points form base pairing with each other, and Step 3 ensures that the length of the independent subsequence is less than or equal to the user-specified parameter κ and that it is not subsumed by a larger independent subsequence that satisfies the same constraints. Each character in the input RNA sequence R is visited at most once, thus the independent subsequences of R is computed in $O(|R|)$ time.

4.4.1 Computing the Interactions between Independent Subsequences

In our new model, the external bondings between nucleotide pairs will be permitted among the independent subsequences of the two RNA sequences S and R, predicted by `mfold`. Below it is given how to compute the external bonds between such nucleotides which minimize the total free energy in the interacting RNA sequences.

From this point on each RNA molecule will be treated as an (ordered) set of independent subsequences (IS), where each IS is indeed a string of nucleotides. The i^{th} IS of an RNA molecule S is denoted by $S_{IS}[i]$. The sequence of ISs between $S_{IS}[i]$ and $S_{IS}[j]$ are thus denoted as $S_{IS}[i,j]$.

The joint structure between S and R is calculated by minimizing the total free energy of their ISs via means of establishing bonds between their nucleotides as follows. Let the minimum free energy of the joint secondary structure of the two ISs $S_{IS}[i]$ and $R_{IS}[j]$ be $e_{IS}(i,j)$. The value of $e_{IS}(i,j)$ can be computed via the algorithm described in Section 4.3.

The minimum joint free energy between the consecutive sets of ISs of S and R is calculated once $e_{IS}(i,j)$ is computed for all i,j. Let n and m denote the number of ISs in S and R respectively. Now let $E(S_{IS}[i], R_{IS}[j]) = E[i,j]$ be the smallest free energy of the interacting independent subsequence lists $S_{IS}[1,i]$ and $R_{IS}[1,j]$ (which satisfy the distance constraint) provided that $S_{IS}[i]$ and $R_{IS}[j]$ interact with each other.

Before showing how to compute the values of $E[i,j]$, we make one final observation on the OxyS-fhlA pair that the "distance" between two interacting subsequences in OxyS appears to be very close to that in fhlA. This may be due to the limited flexibility of "root stems" that support the independent subsequences when they interact with each other. In order to ensure that the predictions made by our algorithm satisfy such limitations, restrictions are imposed on the "distances" between interacting independent subsequences as follows.

Definition 2 *Let $S_{IS}[i]$ and $S_{IS}[j]$ be two independent subsequences in a given RNA sequence S. The distance between $S_{IS}[i]$ and $S_{IS}[j]$, denoted $d(S_{IS}[i], S_{IS}[j])$*

is defined as the number of nucleotides $S[k]$ that do not lie between a bonded pair of nucleotides $S[h]$ and $S[h']$ that are both located between $S_{IS}[i]$ and $S_{IS}[j]$.

The above definition simply ignores all nucleotides that lie in the independent subsequences between $S_{IS}[i]$ and $S_{IS}[i']$ regardless of their lengths. Our algorithm ensures that if $S_{IS}[i] - R_{IS}[j]$ and $S_{IS}[i'] - R_{IS}[j']$ are pairs of consecutive independent subsequences that interact with each other and if $d(S_{IS}[i], S_{IS}[i']) \geq d(R_{IS}[j], R_{IS}[j'])$ then $d(S_{IS}[i], S_{IS}[i']) \leq (1+\epsilon) \cdot d(R_{IS}[j], R_{IS}[j']) + \delta$; here $\epsilon < 1$ and $\delta > 0$ are user defined parameters.

The value of $E[i,j]$ can be computed through dynamic programming given in Table 4.6 with one exception. Rather than calculating the free energy of a kissing loop pair only by the Rivas and Eddy approach [62], the pair is also allowed to establish a double helix structure. Every turn of the double helix (of length ~10nt) must now be compensated by a non-interacting counter turn with length ~3nt (see, for example, the interaction between CopA and CopT).

$$E[i,j] = \min_{i'<i, j'<j \mid d(S_{IS}[i'], S_{IS}[i]) \leq (1+\epsilon) \cdot d(R_{IS}[j'], R_{IS}[j]) + \delta} \begin{pmatrix} E[i',j'] + e_{IS}(i,j) + \\ \sum_{i'<i''<i} e_{IS}(i'',0) + \\ \sum_{j'<j''<j} e_{IS}(0,j''). \end{pmatrix}$$

Table 4.6: Energy table for the loop energy model.

In the energy tables given in Table 4.6, $e_{IS}(i'',0)$ and $e_{IS}(0,j'')$ denote the free energy of independent subsequences $S_{IS}[i'']$ and $R_{IS}[j'']$ respectively.

The overall free energy of the interacting independent subsequence sets of S and R is thus:

$$\min_{\forall i,j} E[i,j] + \sum_{i<i'} e_{IS}(i',0) + \sum_{j<j'} e_{IS}(0,j')$$

The run time of the algorithm for Stacked Pair Energy Model is $O(|S|^3 \cdot |R|^3)$ for input sequences of size $|S|$ and $|R|$. Because the lengths of all independent subsequences are limited by κ, the computation of each $e_{IS}(i,j)$ takes $O(\kappa^6)$. If there exists n independent subsequences in S, and m independent subsequences in R, the total cost of computing all $e_{IS}(i,j)$ values is $O(n \cdot m \cdot \kappa^6)$. Due to the fact that $\sum_{i'<i''<i} e_{IS}(i'',0)$ and $\sum_{j'<j''<j} e_{IS}(0,j'')$ can be computed in $O(1)$ time by a

preprocessing step of $O(n+m)$ time prefix sum, the values of E can be computed in time $O(n^2 \cdot m^2)$. The total cost of the overall algorithm is then $O(n \cdot m \cdot \kappa^6 + n^2 \cdot m^2)$. Because $n \leq |S|/\kappa$ and $m \leq |R|/\kappa$ the worst case running time of this algorithm is $O(|S| \cdot |R| \cdot \kappa^4 + |S|^2 \cdot |R|^2/\kappa^4)$.

This is substantially faster than the earlier approach requiring $O(|S|^3 \cdot |R|^3)$ time. In fact this version can predict the joint structure of the OxyS-fhlA pair in 5 seconds using the same hardware, improving the earlier approach by a factor of 180.

4.4.2 Testing the Loop Energy Model

We tested the Loop Energy Model on the interacting RNA pairs CopA-CopT and OxyS-fhlA, with the same σ values used in Stacked Pair Energy Model: $\sigma \in [0.7, 0.8]$. Joint structure predictions obtained by Loop Energy Model are given in Figures 4.8 for CopA-CopT pair, and 4.9 for OxyS-fhlA pair.

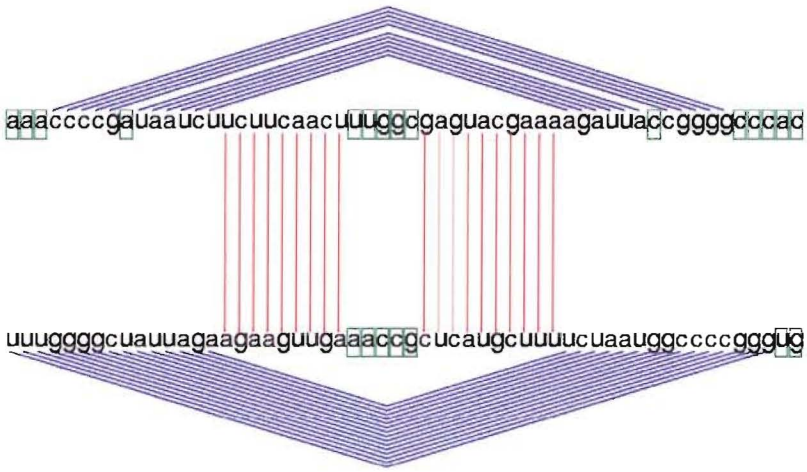


Figure 4.8: Joint structure of CopA and CopT as predicted by Loop Energy Model.

Although there is a slight loss in the prediction quality in CopA-CopT pair with respect to the Stacked Pair Energy Model prediction (Figure 4.6), the "kissing" hairpin sequence is predicted correctly. This test also includes a post processing step that leaves one third of the interacting part unbonded and then does an extra free energy test to check the stability of this modified version. The aim here is to

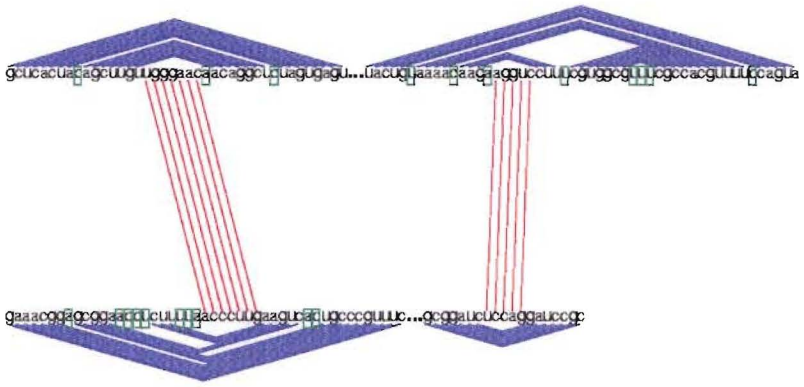


Figure 4.9: Joint structure of OxyS and fhlA as predicted by Loop Energy Model.

capture the topological property of the helical structure as explained earlier. This extra post-processing step can be used if the interacting part is longer than a certain threshold; because shorter interactions do not tend to form a helical structure as it is the case in the OxyS/fhlA pair. In the OxyS-fhlA test, notice that the predictions obtained by the Loop Energy Model and the Stacked Pair Energy Model are even more similar. Furthermore, careful observation shows that the total free energy in the predicted structure is still better than the natural joint structure (Figure 2.5).

4.5 Target Prediction for Antisense RNAs

An important byproduct of our algorithms for the RIP problem is the ability to search for target sequences for specific antisense RNA molecules in whole genomic and plasmid sequences. Because of the time and space constraints, the Stacked Pair Energy Model is not efficient when searching through large sequences. Therefore, the target prediction approach is based on Loop Energy Model. Our search strategy employs the following steps:

1. First, we need to find the "candidate" target sequences from a given genome sequence (or plasmid) that is known or suspected to include the target. This is achieved via using the gene annotation available for genomic sequences. To compute the potential mRNA each such gene is extended towards 5′ and 3′

UTR ends as follows.

(a) Each gene is extended up to l_1 nucleotides at its 5′ UTR, and by l_2 nucleotides at its 3′ UTR, where l_1 and l_2 are user defined parameters. (In the experiments these parameters are set as $l_1 = 250$ and $l_2 = 25$).

(b) Then each "extended" gene sequence is trimmed from both ends via a dynamic programming routine in order to compute its subsequence which has the lowest "energy density" (this will be the subsequence of the extended gene sequence whose secondary structure is most stable.) We predict the resulting mRNA of each such gene as its trimmed extension and after replacing each T character with a U.

2. The joint secondary structure prediction algorithm based on Loop Energy Model is then run to determine if there are any external bonds formed between each candidate target sequence and the antisense RNA sequence under the following constraints. (1) At least one IS in the candidate target sequence which lies before the start codon (i.e. AUG) should interact with an independent subsequence in the query sequence. We impose this constraint in order to capture the ribosome binding site interactions. (2) All predicted interactions between pairs of ISs should include at least ξ uninterrupted bonds for some user specified constant ξ. We impose this constraint to favor long uninterrupted external bonds, since ribosomes are capable of breaking shorter interactions. (3) At least two pairs of independent sequences must be interacting with each other.

4.5.1 Testing the Target Prediction Strategy

We tested the above approach on both RNA-RNA interactions that are considered in the previous tests. (1) First, the target mRNA sequences for CopA are searched in the R1 plasmid sequence in *E.coli*. It is known that CopA regulates the copy number of R1 plasmid by binding to the CopT sequence which is a part of the 125Kb long plasmid [37, 70]. Our search program needed about 12 hours on a PC equipped with 3.2 Ghz Pentium IV processor and 2 GB of main memory to detect all targets of

the CopA sequence on the complete R1 plasmid. Out of the 141 potential mRNA segments obtained from the annotated gene sequences it returned the correct target CopT as a potential target, along with 8 other potential targets. But when the returned "hits" are sorted with respect to their free energies, CopT ranks in the first place.

(2) We then used our program to detect the target mRNA sequences of the OxyS antisense RNA on a 130Kb long portion of *E.coli* genome that included the known target fhlA [71]. Out of the 100 potential mRNA segments obtained from the annotated gene sequences, the program returned 9 hits including the known target fhlA, again ranking in the first place when the hits are sorted with respect to their free energies.

Notice that the joint structure between CopA and CopT are much more stable than that between OxyS and fhlA (the former one has a half-life of about an hour where as the latter one has a half-life of only a couple of minutes). It is possible that OxyS may have other targets in the *E.coli* genome with which it may establish unstable joint structures, not strong enough to make impact.

Chapter 5

Classification of Small Chemical Molecules

In the second part of this thesis, we focus on the functional prediction of small chemical compounds. How to use a distance measure for capturing the similarity among small chemical compounds is described in chapter 2. Here we describe how to compute a distance measure that will maximize the discrimination between active and inactive compounds with respect to a given bioactivity.

The k-nearest neighbor (k-nn) classification method, deduces the level of the bioactivity of a query molecule based on the number (and the bioactivity levels) of active elements among its k-nn with respect to a distance measure of choice. Although k-nn classification is a well known data mining method, it was not considered for small molecule classification until recently [77, 31]. The few known applications of k-nn method to compound classification aim to select the most relevant set of chemical descriptors to reduce the size of the descriptor arrays used. The compounds are then compared under the standard (unweighted) L_1 or L_2 distance.

We introduce use of the (more general) weighted Minkowski distance of order 1, namely wL_1 for classification of small chemical compounds. For each bioactivity of interest, we determine real valued weights w_i of the wL_1 distance so as to maximize the discrimination between active and inactive compounds in a training set. (Thus, earlier applications of k-nn to compound classification can be seen as limited versions of our approach where the weights w_i are set to either 0 or 1.) We compute the

optimal values for weights w_i via a linear optimization procedure.

Our experiments show that our k-nn classifier with respect to wL_1 distance provides better accuracy than the LDA and MLR, sometimes significantly so. Note that, as per LDA and MLR, our classifier is also based on a projection of molecules to a metric space. As per MLR (and in contrast to LDA) the number of dimensions in the projection space is equal to the number of descriptors. However, unlike MLR and LDA, our classifier is not described by a simple planar cut on the projection space but by a complex surface defined by the combination of surfaces in the form of *balls* with specific data elements in their center. Although our classifier uses more complex surfaces (which results in higher accuracy) we can still perform fast classification, thanks to the efficient data structures we develop for nearest neighbor - see below. Our method is comparable to the ANN classifier in terms of accuracy. Yet it is superior to the ANN classifier in the sense that it determines the level of bioactivity (rather than giving a simple YES or NO answer) as per the MLR based solutions. It turns out that our classifier is also faster than the ANN classifier - this we achieve through an efficient data structure we develop for efficient similarity search as described below.

5.1 Distance measures for small molecules and distance based classification

Given a chemical compound s, its descriptor array S is defined to be an n dimensional vector in which each dimension i, denoted by $S[i]$, is a real value corresponding to the descriptor associated with dimension i. For a given bioactivity, it is of significant interest to come up with a distance measure $D(S, R)$ between pairs of descriptor arrays S and R that correspond to the similarity in the bioactivity levels of the corresponding compounds s and r: if the bioactivity levels are similar, the distance must be small and vice versa. Such a distance measure could be very useful in the classification of *new* chemical compounds in terms of the bioactivity of interest: the bioactivity level of the new compound is likely to be identical to the bioactivity level

of the set of compounds that have the smallest distance to the new compound.

A distance measure D forms a metric if the following conditions are satisfied. (i) $D(S,S) = 0$ for all S and $D(S,R) \geq 0$ for all S and R (non-negativity). (ii) $D(S,R) = D(R,S)$ (symmetry). (iii) $D(S,R) \leq D(S,Q) + D(Q,R)$ (triangle inequality). Metric distance of interest include the Hamming distance, Euclidean distance and the Tanimoto distance. Metric distances are of particular interest due to the availability of efficient data structures they admit for fast similarity search.

The commonly used QSAR approach estimates the level of bioactivity of a compound via a linear combination of its descriptors each of which correspond to a specific dimension of its descriptor array. In *distance based* compound classification, it is thus natural to consider a distance between two descriptor arrays which is a linear combination of the differences in each one of the dimensions. More specifically one can define $D(S,R) = \sum_{i=1}^{n} w_i \cdot |S[i] - R[i]|$ where w_i, the weight of the dimension i is a real value in the range $[0,1]$. It is easy to show that this distance, which is usually called the weighted Minkowski distance of order 1 forms a metric.

In this thesis we focus on classification of biomolecules according to five specific bioactivities: (i) being an antibiotic, (ii) being a bacterial metabolite, (iii) being a human metabolite, (iv) being a drug, and (v) being drug-like. The biomolecular data sets available usually do not specify the level of bioactivity of interest but rather provide whether a compound is active or inactive. Thus we only perform a binary classification of compounds for each bioactivity, although our methods are general to provide a real valued level of bioactivity.

Our classification method for a given bioactivity first computes a distance measure for a training data set which *separates* the subset of active compounds from those that are inactive. Given a training set of descriptor arrays $T = \{T_1, T_2, ..., T_\ell\}$ (each of which belonging to a compound) we determine the distance measure D, more specifically compute the associated weights w_i, through a combinatorial optimization approach.

Given the training set T, let $T^A = \{T_1^A, T_2^A, ..., T_m^A\}$ denote its subset of active compounds and $T^I = \{T_1^I, T_2^I, ..., T_{\ell-m}^I\}$ denote its subset of inactive compounds. Clearly $T = T^I \cup T^A$.

We obtain a linear program for determining each w_i as follows. The objective function of the linear program which is to be minimized is

$$f(T) = (\sum_{h=1}^{m}\sum_{j=1}^{m}\sum_{i=1}^{n} w_i \cdot |T_h^A[i] - T_j^A[i]|)/m^2 \tag{5.1}$$

$$+ (\sum_{h=1}^{\ell-m}\sum_{j=1}^{\ell-n}\sum_{i=1}^{n} w_i \cdot |T_h^I[i] - T_j^I[i]|)/(\ell-m)^2 \tag{5.2}$$

$$- (\sum_{h=1}^{m}\sum_{j=1}^{\ell-m}\sum_{i=1}^{n} w_i \cdot |T_h^A[i] - T_j^I[i]|)/(m \cdot (\ell-m)) \tag{5.3}$$

subject to the following conditions

$$\forall T_h^A \in T^A \quad (\sum_{j=1}^{m}\sum_{i=1}^{n} w_i \cdot |T_h^A[i] - T_j^A[i]|)/m^2$$

$$\leq (\sum_{j=1}^{\ell-m}\sum_{i=1}^{n} w_i \cdot |T_h^A[i] - T_j^I[i]|)/(m \cdot (\ell-m)) \tag{5.4}$$

$$\forall i \quad 0 \leq w_i \leq 1 \quad \& \quad \sum_{i=1}^{n} w_i \leq C \tag{5.5}$$

where C is a user defined constant.

The objective function $f(T)$ has three components: Component (1) is the average distance among active compounds and component (2) is the average distance among the inactive compounds; their sum provides the *within-class* average distance. Component (3), on the other hand, is the average distance between an active compound and an inactive one; thus it stands for the *between-class* average distance. As a result our linear programming formulation aims to maximize the difference between the average between-class distance and the average within-class distance. The distance measure obtained will *separate* the typical active compound from the typical inactive compound, while *clustering* all active compounds and all inactive compounds as much as possible.

There are three types of constraints on the weights w_i in our linear programming formulation. Constraint (4) ensures that the average distance among active compounds is no more than the average distance between active and inactive compounds.[1] Constraints (5) impose bounds on the values of weights w_i and their sum.[2]

A note on the performance. We used CPLEX, an open-source linear programming solver for computing the distance measure for a given bioactivity. Because the number of constraints is proportional to the number of active compounds, which is no more than 1500 for the bioactivities we considered, the running time for computing all distance measures of interest was quite reasonable, no more than 2 minutes on a standard 3.2Ghz Intel Pentium D Workstation.

k-nearest neighbor classification of biomolecules

A distance measure defined as above can be used for the classification of compounds with unknown levels of bioactivity as the bioactivity level of a compound is likely to be similar to the bioactivity levels of compounds within its close proximity. Our *k-nn classifier* estimates the (binary) bioactivity of a given compound by (1) either taking the majority of the bioactivities of its k-nearest compounds w.r.t. the distance measure or by (2) checking whether sum of the binary bioactivity levels of the k-nearest neighbors normalized by their distances to the compound is above a threshold value. Under each approach, it is possible to select the value of k which maximizes the accuracy of the estimator, i.e. the ratio of the sum of true positives and true negatives to the size of the training data set.

---

[1] A more stringent set of constraints can be imposed on active compounds such that the distance between a given active compound T_h^A and any other active compound is no more than the distance between T_h^A and any inactive compound. Such a set of constraints can, in principle, can separate active and inactive compounds into tighter clusters. Unfortunately, the number such constraints, $m^2 \cdot (\ell - m)$, turns out to be impractical, even for the most advanced linear program solvers.

[2] The number of descriptors related to a specific bioactivity is usually no more than a few, thus it is desirable to simplify the distance measure by limiting the number of non-zero weights. The final constraint aims to achieve this by imposing an upper bound on the sum of the weights. Although this constraint does not guarantee to upper bound the number of non-zero weights, in practice, the number of non-zero weights obtained are no more than $2C$.

5.2 Experimental Results

Here we aim to provide some insight into the comparative performance of our k-nn classifier, both in terms of accuracy and efficiency. We applied our classifier to five types of bioactivities: (i) being antibiotic, (ii) being a bacterial metabolite, (iii) being a human metabolite, (iv) being a drug, and (v) being drug-like.

The first data set we used is the complete small molecule collection from [12], which includes 520 antibiotics, 562 bacterial metabolites, 958 drugs, 1202 drug-like compounds, and an additional 1104 human metabolites. The total number of the compounds in the data set is 4346. Each compound in the dataset is represented with a descriptor array of 62 dimensions, which is a combination of 30 inductive QSAR descriptors [12] and 32 physicochemical properties such as molecular weight, number of specific atoms (O, N, S), acidity, density, etc. This data set was used for testing the classification quality of our approach. A second data set which enriches the first data set by the addition of 20000 additional drug like compounds was later used for testing the running time of our approach. . For each bioactivity, a wL_1 distance is determined to establish a *model* for compound classification w.r.t. this bioactivity using our k-nn method. Note that the descriptors of each compound are normalized according to the observed maximum and minimum values in the data set in order to remove the bias to parameters with larger values.

The comparative results of the four classification methods, namely k-nn, LDA, MLR and ANN are provided in Table 5.1. For each bioactivity, we provide the sensitivity,specificity and accuracy obtained by each classifier. We demonstrate the performance of our k-nn classifier only for $k = 1$; i.e. given a query compound, our classifier returns the bioactivity of its nearest neighbor in the training data set. We constructed the wL_1 measure for three different values of C - the upper bound on the sum of weights, i.e., $\sum_{i=1}^{n} w_i \leq C$. Setting $C = \infty$ removes the restriction on the sum of weights and thus computes the wL_1 distance that achieves the best classification. We also set C to 3 and 10 to restrict the number of non-zero weights, with the aim of focusing only on the C most relevant descriptors to the bioactivity of interest. As the resulting non-zero weights turned out to be equal to or very

Model		T_P	T_N	F_P	F_N	SPEC	SENS	ACCUR	PPV	NPV
Antibacterial Model, C=∞	Train	269	2610	69	95	.97	.74	.95	.8	.96
	Test	117	1119	28	39	.98	.75	.95	.81	.97
Antibacterial Model, C=10	Train	224	2538	141	140	.95	.62	.91	.61	.95
	Test	92	1085	62	64	.95	.59	.90	.60	.94
Antibacterial Model, C=3	Train	201	2526	153	163	.94	.55	.90	.57	.94
	Test	75	1074	73	81	.94	.48	.88	.51	.93
Antibacterial Model, LDA	Train	364	0	2679	0	0.00	1.00	0.12	0.12	-
	Test	156	0	1147	0	0.00	1.00	0.12	0.12	-
Antibacterial Model, MLR	Train	194	564	2115	170	0.21	0.53	0.25	0.08	0.77
	Test	61	1129	18	95	0.98	0.39	0.91	0.77	0.92
Antibacterial Model, ANN	Train	294	2651	27	70	0.99	0.81	0.97	0.92	0.97
	Test	129	1132	16	27	0.99	0.83	0.97	0.89	0.98
Bacterial Metabolite Model, C=∞	Train	311	2537	112	83	.96	.79	.94	.74	.97
	Test	135	1091	44	33	.96	.80	.94	.75	.97
Bacterial Metabolite Model, C=10	Train	220	2436	213	174	.92	.56	.87	.51	.93
	Test	98	1038	97	70	.91	.58	.87	.50	.94
Bacterial Metabolite Model, C=3	Train	152	2376	273	242	.90	.39	.83	.36	.90
	Test	80	1018	117	88	.90	.48	.84	.41	.92
Bacterial Metabolite Model, LDA	Train	240	2587	62	154	0.98	0.61	0.93	0.79	0.94
	Test	90	1088	47	78	0.96	0.54	0.90	0.66	0.93
Bacterial Metabolite Model, MLR	Train	301	2525	124	93	0.95	0.76	0.93	0.71	0.96
	Test	119	1073	62	49	0.95	0.71	0.91	0.66	0.96
Bacterial Metabolite Model, ANN	Train	338	2597	52	55	0.98	0.86	0.96	0.87	0.98
	Test	159	1076	59	10	0.95	0.94	0.95	0.73	0.99
Drug Model, C=∞	Train	474	2158	214	197	.91	.71	.86	.69	.92
	Test	204	928	88	83	.91	.71	.87	.70	.92
Drug Model, C=10	Train	349	2072	300	322	.87	.52	.80	.54	.87
	Test	151	861	155	136	.85	.53	.78	.49	.86
Drug Model, C=3	Train	305	2026	346	366	.85	.45	.77	.47	.85
	Test	126	846	170	161	.83	.44	.75	.43	.84
Drug Model, LDA	Train	0	2372	0	671	1.00	0.00	0.78	-	0.78
	Test	0	1014	2	287	0.99	0.00	0.78	0.00	0.78
Drug Model, MLR	Train	279	2234	138	392	0.94	0.42	0.83	0.67	0.85
	Test	109	951	65	178	0.94	0.38	0.81	0.63	0.84
Drug Model, ANN	Train	489	2178	194	182	0.92	0.73	0.88	0.72	0.92
	Test	177	978	39	110	0.96	0.62	0.89	0.82	0.90
Druglike Model, C=∞	Train	674	2043	158	168	.93	.80	.89	.81	.92
	Test	281	866	77	79	.92	.78	.88	.0.78	.92
Druglike Model, C=10	Train	560	1959	242	282	.89	.67	.83	.70	.87
	Test	239	842	101	121	.89	.66	.83	.70	.87
Druglike Model, C=3	Train	467	1813	388	375	.82	.55	.75	.55	.83
	Test	197	275	168	163	.82	.55	.75	.54	.83
Druglike Model, LDA	Train	683	1917	284	159	0.87	0.81	0.85	0.71	0.92
	Test	295	801	142	65	0.85	0.82	0.84	0.68	0.92
Druglike Model, MLR	Train	665	1951	250	177	0.89	0.79	0.86	0.73	0.92
	Test	282	812	131	78	0.86	0.78	0.84	0.68	0.91
Druglike Model, ANN	Train	734	2086	114	107	0.95	0.87	0.93	0.87	0.95
	Test	334	891	52	27	0.94	0.93	0.94	0.87	0.97
Human Metabolite Model, C=∞	Train	773	2270	0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00
Human Metabolite Model, C=10	Train	772	2266	4	1	.99	.99	.99	.99	.99
	Test	330	972	0	1	1.00	0.99	.99	1.00	.99
Human Metabolite Model, C=3	Train	772	2270	0	1	1.00	0.99	.99	1.00	.99
	Test	330	972	0	1	1.00	0.99	.99	1.00	.99
Human Metabolite Model, LDA	Train	773	2270	0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00
Human Metabolite Model, MLR	Train	773	2270	0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00
Human Metabolite Model, ANN	Train	773	2270	-0	0	1.00	1.00	1.00	1.00	1.00
	Test	331	972	0	0	1.00	1.00	1.00	1.00	1.00

Table 5.1: Binary classification of the bioactivities of the test set according to four classification methods: k-nn, LDA, MLR, ANN.

close to 1, these two classifiers are quite similar to those described in recent papers (e.g. [77, 31]) that focus on determining the most relevant descriptors for modeling a bioactivity of interest.

We used MOE(Molecular Operating Environment) PLS module for MLR classification and SNNS (Stuttgart Neural Network Simulator) with default parameters (52 nodes and 420 connection network) for ANN classification. LDA classification is performed through the use of standard C libraries for matrix operations.

For each bioactivity, a *training data set* comprising of 70 percent of both the active and the inactive compounds are formed via random selection. The remaining compounds are used as the *test data set*. Each training data set is used for building the four classifiers corresponding to the related bioactivity and the test data is used for the evaluating their performance.

For each bioactivity/classifier pair we report the following test results: The number of true positives (T_P), the number of true negatives (T_N), the number of false positives (F_P), the number of false negatives (F_N), sensitivity (T_P/(T_P+F_N)), specificity (T_N/(T_N+F_P)), accuracy ((T_N+T_P)/(T_P+T_N+F_P+F_N)), positive predictive value (T_P/(T_P+F_P)), negative predictive value (T_N/(T_N+F_N)).

We have demonstrated that our k-nn classifier with respect to wL_1 distance obtains better accuracy than the LDA and MLR, sometimes significantly so. It is comparable to the ANN classifier in terms of accuracy and is superior in the sense that it is capable of determining a real valued level of bioactivity rather than giving a simple YES or NO answer.

5.2.1 Separation of Drugs, Nondrugs, Antimicrobials, and Metabolites in Descriptor Space

To gain a better understanding of the distinctive behavior of human metabolites and their positioning in the descriptors space we considered a data set of more than 2 million druglike chemical structures downloaded from the ZINC database [30]. For every substance in that data set we calculated the same 62 descriptors selected for modeling and assumed that such large data set should sufficiently cover all feasible

values of QSAR parameters.

To assess separation between the studied groups in the descriptors space and to sample their compactness and overlaps, we utilized the previously trained distance measures for each bioactivity. Thus, for each studied class of chemical substances we considered every constituent molecule as a probe that has then been placed into chemical space consisting of 4346 studied compounds mixed with 2.066.095 ZINC structures. For each probe we applied the corresponding distance function to identify all active entries located within a certain radius R. For each studied group of compounds, we have continued such probing until all active elements in the class are identified. Understandably, the established number of the required probe-based queries strongly depended on the probe radius. Figure 5.1 features probe-based recovery of antimicrobials, bacterial metabolites, drugs, druglikes, and human metabolites from the pool of 2.071.251 entries as the neighbors of the query compounds with different radius values of 0.10, 0.15, and 0.20. The blue recovery curves in Figure 5.1 , corresponding to probing with radius $R = 0.20$, illustrate that k-nn recovery of the majority of antimicrobials, drugs, nondrugs, and metabolites can be accomplished in less than 100 iterations. When the database has been queried with $R = 0.15$ and, particularly $R = 0.10$ probes (red and green curves, respectively), the complete recovery may require as many as 500-700 steps.

Figure 5.1 also feature random recovery of 520 antimicrobials, 959 drugs, 1202 druglikes, and 562 bacterial and 1104 human metabolites from the total of 2.071.251 chemicals structures (the corresponding curves are marked in dashed lines). As random recovery curves illustrate, only members of the inactive druglike compounds group could be found somewhat efficiently by random placing of probes into chemical space. On another hand, active probe-based recovery of druglike substances was not very efficient either (see Figure 5.1(d)). These observations may justify that druglike entries are spread throughout the descriptors space without distinctive clustering. In contrast, other types of substances, particularly human metabolites, could be recovered very rapidly by the k-nn search, which characterizes them as compact collections of entries.

One useful criterion for assessing clustering of active entries in a large database is

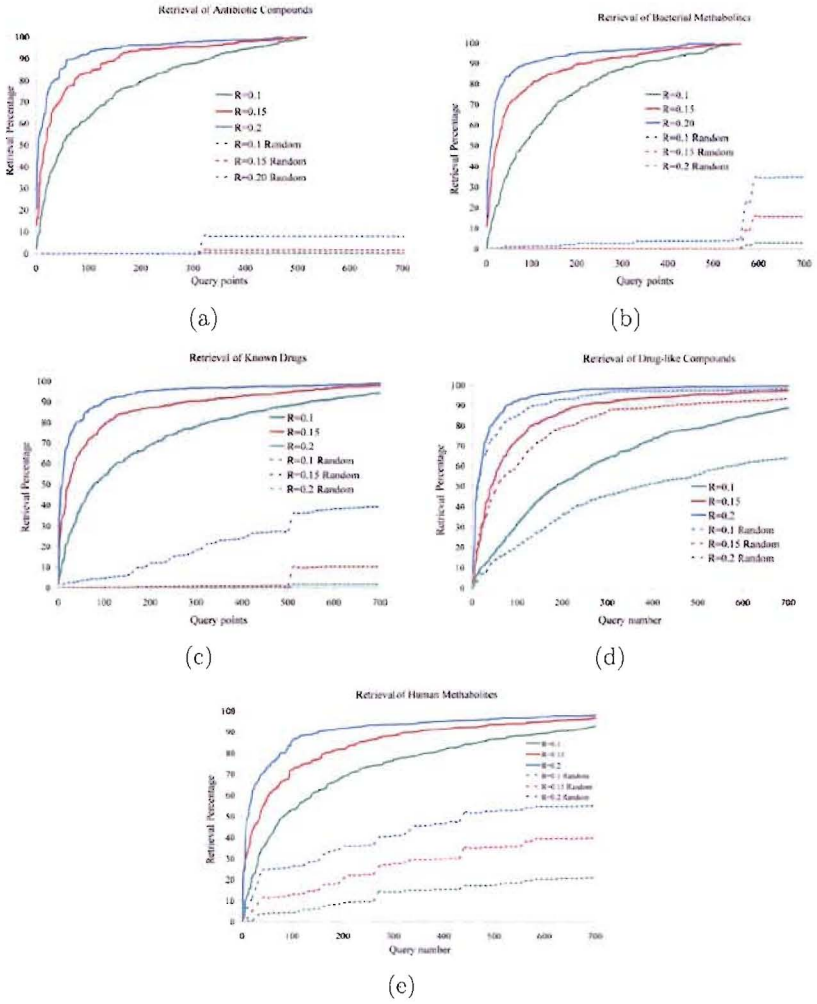


Figure 5.1: (a) Retrieval of antimicrobial compounds from the general molecular database (> 2M entries) using the range queries with varying distance constrains (solid lines). The dashed lines correspond to random identification of antimicrobial compounds. This representation (solid/dashed line) is same for the remaining bioactivities. (b) Retrieval of bacterial metabolite compounds from the general molecular database. (c) Retrieval of drugs from the general molecular database. (d) Retrieval of druglike compounds from the general molecular database. (e) Retrieval of human metabolite compounds from the general molecular database.

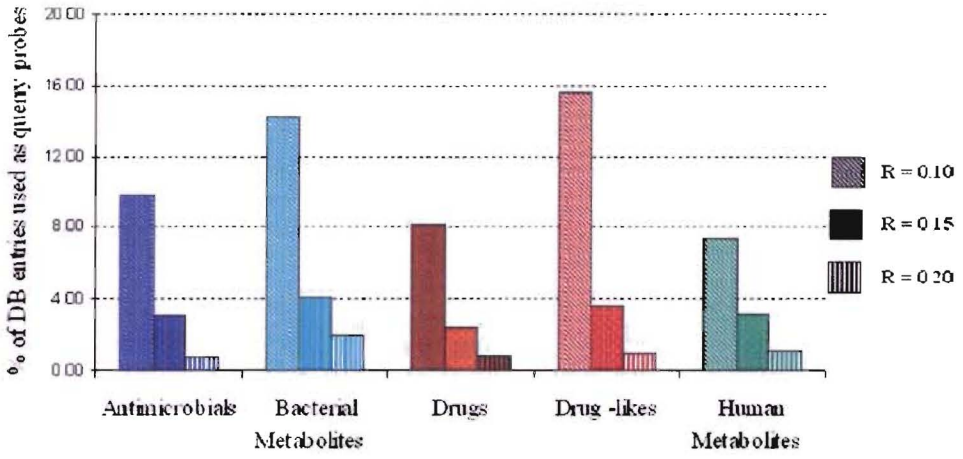
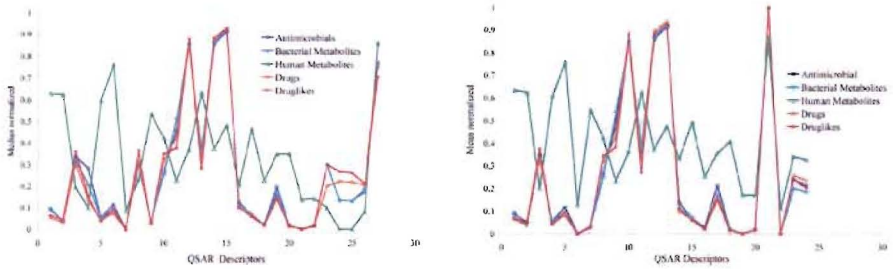


Figure 5.2: Histograms of $P_{1/2}$ values -fractions of cluster entries required to retrieve 50% members of the corresponding cluster from a large molecular database using the k-nn approach. The values have been identified for the searches with varying R parameters.

the number of k-nn probes of a certain radius that are required for identification of 50% of active entries. Thus, we have computed the corresponding parameters $P_{1/2}$ for five k-nn models with search radius values of 0.10, 0.15, and 0.20. The established numbers of probes required to identify 50% of each group are featured in Figure 5.2, where they are normalized by the size of the corresponding activity group. Thus, it required only 81 probes (or 7% of the total number of entries) with $R = 0.10$ radius to identify 552 human metabolites (50% of the total number) from the mixed pool of more than 2 million chemical structures. This illustrates that human metabolite substances are clustered very tightly in multidimensional descriptors space. The grouping becomes less profound for conventional drugs, followed by antimicrobials, bacterial metabolites, and, finally, by the group of druglikes which required more than 15% of actives to be used as probes to locate 50% of the group (see Figure 5.2).

The results of cross-recognition analysis also confirmed uncharacteristic QSAR behavior of human metabolites: the ANN model trained to recognize them in the mixed set of compounds did not produce any false positive predictions. The later may reflect the fact that QSAR descriptors computed for human metabolites follow different trends when compared to drugs, inactive chemicals, antimicrobials, and



(a) (b)

Figure 5.3: (a) Median values for selected "inductive" and conventional QSAR descriptors (normalized) calculated independently within studied sets of chemical substances. (b) Averaged values of selected "inductive" and conventional QSAR descriptors (normalized) calculated independently within studied sets of chemical substances.

bacterial metabolites. To illustrate this point we plotted median and mean values of inductive and 2D-QSAR descriptors that have been computed independently for the studied groups of chemicals (see Figure 5.3(a) and Figure 5.3(b)). The charts clearly demonstrate that descriptors computed for human metabolites appear differently.

To summarize the results of the above-described experiments it is possible to conclude that groups of antimicrobial compounds, conventional drugs, druglike chemicals, and bacterial and human metabolites form distinctive and relatively compact clusters in chemical space where the dimensions are defined by inductive and conventional QSAR descriptors. Such clustering allows rather accurate recognition of these types of biological activity using various statistical and machine-learning techniques that include methods of artificial neural networks, k-nearest neighbors, linear discriminative analysis, and multiple linear regression. The QSAR separation of antimicrobials, drugs, nondrugs, and metabolites with these approaches demonstrates a certain degree of similarity between the members of these activity classes resulting in their cross-recognition by the corresponding QSAR models. On another hand, the group of human metabolites demonstrated rather distinctive behavior com- pared to all other studied types of chemicals, with the corresponding cluster of entries being the most compact and completely separated from other groups in the descriptors space. Thus, the results of the comparative QSAR analysis allow categorizing human metabolites as a distinctive class of chemical structures and raises questions

about structural determinants of their unusual QSAR properties.

Chapter 6

Data Structures for k-nn Classification

Given the "optimal" distance measure which is computed using the linear programming described in chapter 5, it is desirable to construct an efficient data structure for performing k-nn search. Another common way of performing k-nn search is through range queries where all the compounds within a certain search radius is returned. K-nn search can be performed by starting with a small range query and increasing it iteratively until all k neighbors of the query is returned. In this chapter, we show how to construct an efficient data structure for performing range queries under any given metric distance measure and provide some experimental results.

Vantage Point(VP) Trees and its extension Space Covering Vantage Point(SCVP) Trees are desribed in chapter 2 for performing range queries. In the original SCVP tree construction, the vantage points in each level are chosen randomly until all search space is covered [63]. Clearly, it is desirable to minimize the number of vantage points that cover the search space. With fewer vantage points picked at each level, a better space utilization can be achieved, implying that more levels of the tree can be fitted in the available memory.

We first prove that the problem of minimizing the number of vantage points at each level is an NP-hard problem. However, we show how to approximate the minimum number of vantage points and thus obtain the optimum allocation of available memory through a simple polynomial time algorithm. The resulting data

structure, which we call the deterministic multiple vantage point tree (DMVP tree), when built in full, is guaranteed to have $O(\log \ell)$ levels, where ℓ is the size of the data set. If the maximum number of children of an internal node at level i is c_i, the query time guaranteed by our data structure is $O(\sum_{i=1}^{\log \ell} c_i$. Because c_i is typically a small constant, the query time is only $O(\log \ell)$, a significant improvement over linear/brute force search.

Due to redundant representation of data items, the memory usage of the DMVP tree can be super-polynomial. In case the full version of the DMVP tree requires more memory than available, lower levels of the DMVP trees could be cut out. In this case, when the search routine reaches the final level built, the pruning in the respective subspace can be achieved by linear search. We also show how to obtain the optimum cut so as to minimize the expected query performance.

Our data structure is not only interesting for classification purposes; similarity search among small molecules under various notions of similarity is of independent interest. To the best of our knowledge, this is the first application of an efficient similarity search data structure to small molecule data collections. In particular, all known k-nn classifiers employ brute force search, which is not scalable with the growth in the size of compound databases (e.g. PubChem).

We demonstrate that the DMVP tree performs very well in practice, achieving fast classification and similarity search. We compare the performance of our data structure against brute force search in terms of the number of comparisons between descriptor arrays that we need to perform under the weighted Minkowski distance. We also demonstrate how well our classifier performs against available alternatives in terms of running time.

6.1 Efficient data structures for k-nn search

Typical similarity search methods for large collections of data elements usually perform iterative partitioning of the data set into smaller subsets so as to perform efficient querying by pruning - which is achieved at each iteration by checking out to which partition the query falls into [69, 74]. The pruning strategy can be made

particularly effective on data collections where similarity is measured with respect to a metric distance. The partitions in such a metric space are usually achieved with respect to simply defined planar cuts; given a query element, it is quite simple to check to which side of the planar cut it falls into.

Given a set of data elements $X = \{X_1, \ldots, X_\ell\}$ in a metric space with distance D, similarity search for a query element Y can be posed in two flavors. (1) Range query: retrieve all items whose distance to Y is at most some user defined R. (2) k-nn query: retrieve the $k \geq 1$ items whose distances to Y are as small as possible.

One particularly efficient similarity search tool for performing range queries is the Vantage Point (VP) trees [69, 74]. Traditionally, a vantage point tree is defined as a binary tree that recursively partitions a data set into two equal size subsets according to a randomly selected vantage point X_v as follows. Let M is the median distance among the distances of the data elements to X_v. The *inner partition* consists of the elements Y such that $D(X_v, Y) < M$ and the *outer partition* consists of the elements Z such that $D(X_v, Z) \geq M$.

For a given query element Y, the set of data elements X_i for which $D(Y, X_i) \leq R$ for the search radius R can be computed as follows. Let X_v be the vantage point chosen for the entire data set and let M be the median distance among the distances of the data elements to X_v. If $D(X_v, Y) + R \geq M$ then recursively search the *outer partition*. If $D(X_v, Y) - R < M$ then recursively search the *inner partition*. If both conditions are satisfied then both partitions must be searched. The correctness of the search routine follows from the triangle inequality.

A natural extension to the traditional vantage point trees is what we call the *Space Covering VP trees* (SCVP Trees) first described by Sahinalp et al [63]. At each level of the SCVP trees, multiple vantage points are chosen so as to increase the chance of inclusion of the query region in one of the inner partition of the vantage points. The original SCVP trees chose the vantage points at each level randomly. Although this approach can perform quite well for certain data collections, it can also result in poor space utilization.

Clearly it is desirable to *cover* the entire data collection by the fewest number of (inner partitions of) vantage points. However, the problem of minimizing the number

of vantage points for this purpose turns out to be an NP-hard problem under all distance measures of interest (i.e. weighted Minkowski distance of any order p, wL_p); this is proven below. Nevertheless it is possible to approximate the minimum number of vantage points in any metric space through a simple polynomial time algorithm as we show later. As a result we obtain a data structure that deterministically picks the vantage points (whose inner partitions cover the entire data set) which results in almost optimal redundancy; we call this data structure *Deterministic Multiple Vantage Point* tree (DMVP tree).

We start with showing that the optimal vantage point selection problem, which we call OVPS problem, is NP-hard for any weighted Minkowski distance of order p, namely wL_p.

Theorem 3 *OVPS problem under the weighted Minkowski distance of any order p is NP-hard.*

Proof: We establish the NP-hardness of the OVPS problem under L_p through a reduction from the Dominating Set Problem which is known to be NP-hard. The decision version of the Dominating Set problem is as follows: Given a graph $G(V, E)$ and an integer k decide whether there exists a subset V' of vertices V such that every vertex in $V - V'$ has a neighbor in V'. The decision version of the OVPS problem in L_p is as follows: Given a set S of points in L_p, a radius r, and an integer k, decide whether there exists k (vantage) points such that the distance between each point in the set and at least one of the k points is less than r.

From an instance of the Dominating Set problem we first construct a $|V|$ dimensional space S where each vertex V_i is mapped to a point X_i in S as follows.

$$X_i[j] = \begin{cases} 1 & if\ i = j \\ -\epsilon & if\ (V_i, V_j) \notin E \\ 0 & if\ (V_i, V_j) \in E \end{cases}$$

One can calculate upper and lower bounds for the L_p distance between two vectors X_i and X_h as follows.

$$L_p(X_i, X_h)^p = \sum_{j=1}^{|V|} w_i \cdot |X_i[j] - X_h[j]|^p.$$

$$= \begin{cases} a \geq 2(1+\epsilon)^p & if\ (V_i, V_h) \notin E \\ b \leq 2 + \epsilon^p(|V| - 2) & if\ (V_i, V_h) \in E \end{cases}$$

If for a given p one picks ϵ such that

$$\epsilon < \frac{2p}{(|V| - 2)}$$

then

$$\binom{p}{p-1}\epsilon^{p-1} > \epsilon^p \frac{(|V| - 2)}{2}$$

which implies that

$$1 + \binom{p}{1}\epsilon + \ldots + \binom{p}{p-1}\epsilon^{p-1} + \binom{p}{p}\epsilon^p > 1 + \epsilon^p \frac{(|V| - 2)}{2}$$

and thus

$$2(1+\epsilon)^p > 2 + \epsilon^p(|V| - 2)$$

which implies that

$$a > b.$$

In other words, b, the distance between any two vectors whose corresponding vertices are connected (by an edge) is less than a, the distance between any two vertices which are not connected. We now simply pick r so that $a > r > b$.

We now show that G has a dominating set of size k if and only if there exists k vantage points for which the distance between each point in the data set S and at least one of the vantage points is at most r.

Given G, and a dominating set D of size k, we show that the k points in S that correspond to the k vertices in D, *cover* the entire set S. For any vertex $V_i \notin D$, there must exist a neighboring vertex $V_h \in D$. But if V_i and V_h are neighbors then by the above argument $L_p(X_i, X_h) < r$, i.e. X_i is in the radius-r-neighborhood of

the vantage point X_h.

Given S, and k vantage points whose radius-r-neighborhoods cover all points in S, we show that the k vertices in G that correspond to the k vantage points form a dominating set. For any point X_i which is not a vantage point, there must exist a vantage point X_h s.t. $wL_o(X_i, X_h) < r$. But this implies that V_i and V_h must be neighbors in G, i.e. V_i must have a neighbor which is in the dominating set.

The generalization of the proof to wL_p is not difficult and is not given here. ∎

Corollary 4 *OVPS problem under Tanimoto distance is NP-hard.*

Proof: The Tanimoto distance is no more than L_1 on binary vectors normalized by the number of dimensions (which is a constant). ∎

An $O(\log \ell)$ approximation to the optimal vantage point selection

The variant of the OVPS problem for which we establish NP-hardness assumes a fixed radius r for each neighborhood around a vantage point. One can think of two natural variants of the OVPS problem: (1) each neighborhood includes a fixed number of points (e.g. $\ell/2$ points as per the original VP Tree construction), (2) each neighborhood has at least ℓ/k and at most ℓ/k' points for some $k \geq k'$. It is not difficult to show that these variants are NP-hard as well.

In the remainder of this section we focus on variant (2) of the OVPS problem and describe a polynomial time $O(\log \ell)$ approximation algorithm for solving it. Such a solution will also imply an $O(\log \ell)$ approximation algorithm for variant (1) by setting $k = k'$. The approximation algorithm is achieved by reducing the OVPS problem to the weighted set cover problem as follows.

Consider each point X_i in S. We construct the following ℓ sets for X_i named $X_i^1, X_i^2, \ldots, X_i^\ell$. X_i^1 consists of only X_i. X_i^2 consists of X_i and its nearest neighbor. In general, X_i^j consists of X_i and its $j-1$ nearest neighbors. Let the cost of X_i^j be j.

Now given sets X_i^j, for all $1 \leq i \leq \ell$ and $k \leq j \leq k'$, each with cost j, if we can compute the minimum cost collection of sets such that each $X_h \in S$ is

in at least one such set, we would get a solution to the variant (2) of the OVPS problem. This problem is equivalent to the weighted set cover problem for which a simple greedy algorithm provides an $O(\log \ell)$ approximation (e.g. [13]). The greedy algorithm works iteratively: each iteration simply picks a set where the cost-per-uncovered-element is minimum possible. The algorithm terminates when all elements are covered.

Optimal fitting of the multiple vantage point tree in the memory

Although the deterministic multiple vantage point tree improves the memory usage of the randomized space covering vantage point tree, it is still possible that the tree may not fit in the main memory. If this is indeed the case, we try to place a connected subtree (which includes the root) to the memory. The search again is performed starting with the root. When an internal node whose children are not represented in the memory is reached, the search is done in a brute force manner on the set of points represented by that node.

Clearly it is of interest to obtain the *best* subtree for optimizing the query performance of the data structure. For that we use the following $0-1$ programming formulation.

Given a Multiple Vantage Point tree T and a node i, let S_i be the number of points in the neighborhood represented by i. During a search, when a node j is reached, its children $i, i+1, \ldots$ are considered for further search in linear order; i.e. we first check whether the query fits in the neighborhood of i, then we check $i+1$ and so on until a suitable vantage point $i+h$ is found. Let S'_{i+h} be the number of points in the neighborhood represented by node $i+h$ which are not in the neighborhoods represented by $i, i+1, \ldots, i+h-1$.

Our $0-1$ programming formulation sets the *probability* that node $i+h$ is reached during a search to S'_{i+h}/ℓ. If the children of the node $i+h$ are not placed in the memory, i.e. if node $i+h$ is on the *cut-set*, the time needed for performing a search on the neighborhood represented by this node is S_{i+h}. Thus the expected contribution

of node $i+h$ to the query time is $S_{i+h} \cdot S'_{i+h}/\ell$.

Let b_i be a binary variable, which takes the value 1 if vertex i is in the cut-set and is 0 otherwise. Our goal is to minimize the expected running time of the brute-force search performed for each query; i.e. our objective function is $f(T) = \sum_{\forall i} b_i S_i S'_i$ subject to the following constraints.
For any pair of consecutive sibling nodes i and $i+1$, we must have $b_i = b_{i+1}$.
We should not exceed the memory M dedicated to the cut-set; thus $\sum_{\forall i} b_i S_i \leq M$.
Finally, at least one node in every path from the root to a leaf in T must include one vertex in the cut-set. Thus for any such path P we have $\sum_{i \in P} b_i = 1$.

A $0-1$ assignment to b_i's that minimize the objective function will minimize the expected query time while fitting the data structure in the main memory.

6.2 Experimental Results

We use the same dataset with the same separation of training and testing set in the previous section for evaluating the efficiency of our data structure. Our similarity search data structure for computing the nearest neighbor of the query compound is quite efficient, especially when compared to brute force search. We tested our data structure under the wL_1 distance computed for each of the five bioactivities, on both of the data sets. The crucial parameter that determines the performance of our data structure is the pruning it achieves for any given query compound. Thus we determine the percentage of compounds pruned in the second training data set (the first training data set enriched with 20000 drug like compounds), averaged over all compounds in the test data set. On a 32GB Sun Fire V40Z server (with 2.4 Ghz AMD 64bit Opteron processor) the respective pruning ratios are as follows. We achieved (i) 84.4% pruning for being antibiotic, (ii) 84.5% pruning for being bacterial metabolite, (iii) 86.1% pruning for being human metabolite, (iv) 81.7% pruning for being drug, and (v) 81% pruning for being drug-like. This is significant improvement over brute force search.

As a result our k-nn classifier turns out to be very fast. On the first data set, the running time of our k-nn classifier averaged over all 4346 compounds (training+test

data sets) and all five bioactivities is 0.3 milliseconds on the above server. In contrast the ANN classifier requires 39.7 milliseconds on the same data set. On the second data set (which simply has additional 20000 compounds in the data structure) the running time of our k-nn classifier increases only to 1.3 milliseconds (again averaged over the 4346 compounds from the first data set and five bioactivities), still 30 times better than the ANN trained over a much smaller set.

Notice that our classifier is faster, thanks to the DMVP tree data structure which improves the existing vantage point tree data structures in multiple ways. It provides a deterministic selection of the optimal vantage points in each level as well as providing the optimal cut of the tree so as to fit it in the available memory. Our data structure can be applied to any metric distance including the wL_p distance for any p and the Tanimoto distance. It performs very well in practice, achieving fast similarity search and classification.

Chapter 7

Conclusion and Discussion

In this thesis we show how to apply combinatorial optimization methods, in particular dynamic programming and linear programming, to help solve two important problems in computational molecular biology: (1) structural prediction of RNA molecules and (2) functional prediction of small biological compounds.

We first describe the RNA secondary structure prediction problem and introduce the notion of *energy density* which improves the accuracy of the available methods significantly. Because the notion of energy density is non-linear, the standard dynamic programming approaches that has been used in the available total free energy minimization methods are updated. The end result, which is described in this thesis, can perfectly capture the secondary structure of many non-coding RNAs which have been difficult to even approximate with alternative methods.

One key application of RNA structure prediction is determination of interactions between two RNA sequences (e.g. an mRNA and a regulatory RNA). We formulate the RNA-RNA interaction prediction problem as a combinatorial optimization problem and show how to solve it again via dynamic programming. Because the complexity of the algorithm to solve the most involved formulation of the problem is very high, we also describe some heuristic shortcuts, which, in practice, are highly accurate.

The second set of problems we tackle are related to functionality of small chemical molecules. In particular we focus on structural similarity search among small chemical molecules, a standard approach used for in-silico drug discovery. It is possible

to use structural similarity to deduce the bioactivities of new compounds provided that the notion of similarity reflects the bioactivity in question and we have good data structures to perform structural similarity search efficiently.

We show how to computationally design the "optimal" weighted Minkowski distance wL_p for maximizing the discrimination between active and inactive compounds with respect to a bioactivity of interest. We also describe how to construct an iterative pruning based data structure for performing "nearest neighbor" search for a query compound with respect to the weighted L_p distance computated.

Bibliography

[1] G. W. Adamson, J. Cowell, M. F. Lynch, A. H. W. McLure, W. G. Town, and A. M. Yapp. Strategic considerations in the design of a screening system for substructure searches of chemical structure files. *J. Chem. Doc*, 13:153–157, 1973.

[2] T. Akutsu. Dynamic programming algorithms for rna secondary structure prediction with pseudoknots. *Dicr. Appl. Math.*, 104(1-3):45–62, Aug 2000.

[3] C. Alkan, E. Karakoc, J. H. Nadeau, S. C. Sahinalp, and K. Zhang. Rna-rna interaction prediction and antisense rna target search. In *Proc. Ninth Annual International Conference on Research in Computational Molecular Biology*, pages 152–171, Cambridge, MA, USA, May 14-18 2005.

[4] C. Alkan, E. Karakoc, S. C. Sahinalp, P. Unrau, A. Ebhardt, K. Zhang, and J. Buhler. Rna secondary structure prediction via energy density minimization. In *Proc. Tenth Annual International Conference on Research in Computational Molecular Biology*, pages 130–142, Venice, ITALY, Apr 2-5 2006.

[5] M. Andronescu, R. Aguirre-Hernandes, A. Condon, and H. Hoos. Rnasoft: a suite of rna secondary structure prediction and design software tools. *Nucleic Acids Res.*, 31(13):3416–3422, 2003.

[6] M. Andronescu, A. Condon, H.H. Hoos, D.H. Mathews, and K.P. Murphy. Efficient parameter estimation for rna secondary structure prediction. In *15th Annual International Conference on Intelligent Systems for Molecular Biology, ISMB*, Vienna, AUSTRIA, Jul 21-25 2007.

[7] A. N. Arslan, O. Egecioglu, and P. A. Pevzner. A new approach to sequence comparison: Normalized sequence alignment. In *Proc. Fifth Annual International Conference on Research in Computational Molecular Biology*, pages 2–11, Montreal, CANADA, Apr 22-25 2001. ACM.

[8] V. Bafna, H. Tang, and S. Zhang. Consensus folding of unaligned rna sequences revisited. In *Proc. Ninth Annual International Conference on Research in Computational Molecular Biology*, volume 3500, pages 172–187, Cambridge, MA, USA, May 14-18 2005. LNBI.

[9] R. D. Brown. Descriptors for diversity analysis. *Persp. Drug Discovery Des.*, 7(8):31–49, 1997.

[10] E. Check. Firm sets sights on gene silencing to protect vision. *Nature*, 430(7002):819, 2004.

[11] X. Chen and C. H. Reynolds. Performance of similarity measures in 2d fragment-based similarity searching: Comparison of structural descriptors and similarity coefficients. *J. Chem. Inf. & Comp. Sci.*, 42:1407–1414, 2002.

[12] A. Cherkasov. Inductive descriptors. 10 successful years in qsar. *Curr. Computer-Aided Drug Des.*, 1:21–42, 2005.

[13] V. Chvatal. A greedy heuristic for the set covering problem. *Math. of Operations Research*, 4:233–235, 1979.

[14] J. M. Claverie. Fewer genes, more noncoding rna. *Science*, 309(5740):1529–1530, 2005.

[15] G. Collins, S. Le, and K. Zhang. A new algorithm for computing similarity between rna structures. In *Proc. 5th Joint Conference on Information Science*, volume 2, pages 761–765, Atlantic City, NJ, USA, Feb 27 - Mar 3 2000. JCIS.

[16] A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant. Classifying rna pseudoknotted structures. *Theor. Comput. Sci.*, 320(1):35–50, 2004.

[17] Jennifer Couzin. Breakthrough of the year: Small rnas make big splash. *Science*, 298(5602):2296–2297, 20 December 2002.

[18] R. D. Cramer, J. D. Bunce, and D. E. Patterson. Crossvalidation, bootstrapping, and partial least squares compared with multiple regression in conventional qsar studies. *Quant. Struct.-Act. Relat.*, 7:18–25, 1988.

[19] E. Davydov and S. Batzoglou. A computational model for rna multiple structural alignment. In *Proc. Symp. on Combinatorial Pattern Matching*, volume 3103, pages 254–269. LNBI, 2004.

[20] C.B. Do, D.A. Woods, and S. Batzoglou. Contrafold: Rna secondary structure prediction without energy-based models. *Bioinformatics*, 22(14):e90–e98, 2006.

[21] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

[22] A. Fire, S. Q. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391:806–811, 1998.

[23] S.M. Freier, R. Kierzek, J.A. Jaeger, N. Sugimoto, M.H. Caruthers, T. Neilson, and D.H. Turner. Improved free energy parameters for predictions of rna duplex stability. *Proc. Natl. Acad. Sci.*, 83:9373–9377, 1986.

[24] L. Le Gallic, D. Sgouras, G. Beal, and G. Mavrothalassitis. Transcriptional repressor erf is a ras/mitogen-activated protein kinase target that regulates cellular proliferation. *Molecular and Cellular Biology*, 19(6):4121–4133, 1999.

[25] P. Geladi and B. R. Kowalski. Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[26] A. C. Good, S. S. So, and W. G. Richards. Structure -activity relationships from molecular similarity matrices. *J. Medicinal Chemistry*, 36:433–438, 1993.

[27] J. Gorodkin, L. Heyer, and G. Stormo. Finding the most significant common sequence and structure motifs in a set of rna sequences. *Nucl. Acids Res.*, 25(18):3724–3732, 1997.

[28] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. Eddy. Rfam: an rna family database. *Nucl. Acids Res.*, 31:439–441, 2003.

[29] I. Hofacker, M. Fekete, and P. Stadler. Secondary structure prediction for aligned rna sequences. *J. Mol. Biol.*, 319:1059–1066, 2002.

[30] J. J. Irwin and B. K. Shoichet. Zinc- -a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model*, 45:177–182, 2005.

[31] P. Itskowitz and A. Tropsha. Kappa nearest neighbors qsar modeling as a variational problem: theory and applications. *J. Chem. Inf. Model.*, 45(3):777–85, 2005.

[32] Y. Ji, X. Xu, and G. D. Stormo. A graph theoretical approach for predicting common rna secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, 20(10):1591–1602, 2004.

[33] E. Karakoc, A. Cherkasov, and S. C. Sahinalp. Distance based algorithms for small biomolecule classification and structural similarity search. In *ISMB 06, 14th Annual International conference on Intelligent Systems*, Fortaleza, BRAZIL, Aug 6-10 2006.

[34] C. H. Kim and I. Tinoco Jr. A retroviral rna kissing complex containing only two G-C base pairs. *Proc. Nat. Acad. Sci. USA*, 97(17):9396–9401, 2000.

[35] B. Knudsen and J. Hein. Rna secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6):446–454, 1999.

[36] B. Knudsen and J. Hein. Pfold: Rna secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, 31(13):3423–3428, 2003.

[37] F. A. Kolb, H. M. Engdahl, J. G. Slagter-Jager, B. Ehresmann, C. Ehresmann, E. Westhof, E. G. H. Wagner, and P. Romby. Progression of a loop-loop complex to a four-way junction is crucial for the activity of a regulatory antisense rna. *EMBO J.*, 19(21):5905–5915, 2000.

[38] F. A. Kolb, C. Malmgren, E. Westhof, C. Ehresmann, B. Ehresmann, E. G. H. Wagner, and P. Romby. An unusual structure formed by antisense-target rna binding involves an extended kissing complex with a four-way junction and a side-by-side helical alignment. *RNA*, 6(3):311–324, Mar 2000.

[39] M. Lagos-Quintana, R. Rauhut, W. Lendeckel, and T. Tuschl. Identification of novel genes coding for small expressed rnas. *Science*, 294(5543):853–858, 2001.

[40] N. C. Lau, L. P. Lim, E. G. Weinstein, and D. P. Bartel. An abundant class of tiny rnas with probable regulatory roles in caenorhabditis elegans. *Science*, 294(5543):858–862, 2001.

[41] David L. Lewis, James E. Hagstrom, Aaron G. Loomis, Jon A. Wolff, and Hans Herweijer. Efficient delivery of sirna for inhibition of gene expression in postnatal mice. *Nature Genet.*, 32(1):107–108, Sep 2002.

[42] G. Lin, B. Ma, and K. Zhang. Edit distance between two rna structures. In *Proc. Fifth Annual International Conference on Research in Computational Molecular Biology*, pages 211–220, Montreal, CANADA, Apr 22-25 2001. ACM.

[43] D. J. Livingstone. *Data analysis for chemists. Applications to QSAR and chemical product design*, page 239. Oxford Univ. Press, 1995.

[44] R. B. Lyngso, M. Zuker, and C. N. S. Pedersen. Fast evaluation of internal loops in rna secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999.

[45] B. Ma, L. Wang, and K. Zhang. Computing similarity between rna structures. *Theo. Comp. Sci.*, 276(1-2):111–132, 2002.

[46] MACCS. Maccs ii manual.

[47] G. M. Maggiora and M. A. Johnson. *Concepts and Applications of Molecular Similarity*. Wiley, New York, 1990.

[48] C. Malmgren, E. Gerhart, H. Wagner, C. Ehresmann, B. Ehresmann, and P. Romby. Antisense rna control of plasmid r1 replication. *The Journal of Biological Chemistry*, 272(19):12508–12512, 1997.

[49] D. Mathews, J. Sabina, M. Zuker, and D. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. *J. Mol. Biol.*, 288(5):911–940, 1999.

[50] D. Mathews and D. Turner. Dynalign: an algorithm for finding the secondary structure common to two rna sequences. *J. Mol. Biol.*, 317(2):191–203, 2002.

[51] J. S. Mattick and M. J. Gagen. The evolution of controlled multitasked gene networks: The role of introns and other noncoding rnas in the development of complex organisms. *Mol. Biol. Evol.*, 18(9):1611–1630, 2001.

[52] M. T. McManus and P. A. Sharp. Gene silencing in mammals by small interfering rnas. *Nat. Rev. Genet.*, 3(10):737–747, 2002.

[53] Eric G. Moss. Rna interference: It's a small rna world. *Curr. Biol.*, 11(19):R772–R775, 2001.

[54] Eric G. Moss. Micrornas: Hidden in the genome. *Curr. Biol.*, 12(4):R138–R140, 2002.

[55] C. Notredame, E. A. O'Brien, and D. G. Higgins. RAGA: Rna sequence alignment by genetic algorithm. *Nucleic Acids Res.*, 25(22):4570–4580, 1997.

[56] C. D. Novina and P. A. Sharp. The rnai revolution. *Nature*, 430:161–164, 2004.

[57] R. Nussinov and A. Jacobson. Fast algorithm for predicting the secondary structure of single stranded rna. *Proc. Natl. Acad. Sci. USA*, 77(11):6309–6313, 1980.

[58] Dmitri D. Pervouchine. IRIS: Intermolecular rna interaction search. In *Proc. Int. Conf. Genome Informatics*, volume 15, pages 92–101, Yokohama, Japan, Dec 19-21 2004. Universal Academy Press.

[59] N. Peyret and J. SantaLucia. HyTher^{TM} version 1.0. at http://ozone2.chem.wayne.edu/Hyther/hythermenu.html. Wayne State University.

[60] N. Peyret, P. A. Seneviratne, H. T. Allawi, and J. SantaLucia. Nearest-neighbor thermodynamics and NMR of DNA sequences with internal A.A, C.C, G.G, and T.T mismatches. *Biochemistry*, 38(12):3468–3477, 1999.

[61] B. J. Reinhart, F. J. Slack, M. Basson, A. E. Pasquinelli, J. C. Bettinger, A. E. Rougvie, H. R. Horvitz, and G. Ruvkun. The 21-nucleotide let-7 rna regulates developmental timing in Caenorhabditis elegans. *Nature*, 403(6772):901–906, 2000.

[62] E. Rivas and S. R. Eddy. A dynamic programming algorithm for rna structure prediction including pseudoknots. *J. Mol. Biol.*, 285(5):2053–2068, 1999.

[63] S. C. Sahinalp, M. Tasan, J. Macker, and Z. M. Ozsoyoglu. Distance-based indexing for string proximity search. *Proc. IEEE Int. Conf. on Data Eng.*, 19:135–138, 2003.

[64] D. Sankoff. Simultaneous solution of the rna folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45:810–825, 1985.

[65] Science. Mapping rna form & function. *Science*, 309(5740), Sep 2005.

[66] Erwei Song, Sang-Kyung Lee, Jie Wang, Nedim Ince, Nengtai Ouyang, Jun Min, Jisheng Chen, Premlate Shankar, and Judy Lieberman. Rna interference targeting fas protects mice from fulminant hepatitis. *Nature Med.*, 9(3):347–351, Mar 2003.

[67] J. Thompson, D. Higgins, and T. Gibson. Clustal-w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucl. Acids Res.*, 22:4673–4680, 1994.

[68] I. Tinoco, O. Uhlenbeck, and M. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230(5293):362–367, 1971.

[69] J. K. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Inf. Proc. Lett.*, 4:175–179, 1991.

[70] E. G. H. Wagner and K. Flardh. Antisense rnas everywhere? *Trends Genet.*, 18(5):223–226, 2002.

[71] NCBI web site. at http://www.ncbi.nlm.nih.gov.

[72] E. Westhof and V. Fritsch. Rna folding: beyond watson-crick pairs. *Structure*, 8:55–65, 2000.

[73] P. Willett, J. M. Banard, and G. M. Downs. Chemical similarity searching. *J. Chem. Inf. & Comp. Sci.*, 38(6):983–996, 1998.

[74] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. *Proc. ACM-SIAM Symp. on Discr. Alg.*, 1:311–321, 1993.

[75] V. V. Zernov, K. V. Balakin, A. A. Ivaschenko, N. P. Savchuk, and I. V. Pletnev. Drug discovery using support vector machines. the case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *J. Chem. Inf. & Comp. Sci.*, 43(6):2048–2056, 2003.

[76] Kaizhong Zhang. `RNA_align` tool. at http://www.csd.uwo.ca/faculty/kzhang/rna/. University of Western Ontario.

[77] W. Zheng and A. Tropsha. Novel variable selection quantitative structure-property relationship approach based on the k-nearest neighbor principle. *J. Chem. Inf. & Comp. Sci.*, 40(185), 2000.

[78] M. Zuker. On finding all suboptimal foldings of an rna molecule. *Science*, 244(4900):48–52, 1989.

[79] M. Zuker and P. Stiegler. Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, 9(1):133–148, 1981.

[80] J. Zupan and J. Gasteiger. *Neural Networks in Chemistry and Drug Design.* Wiley, New York, 2 edition, 1999.