UNIVERSITY OF SOUTHAMPTON

Functional Nucleic Acids as Substrate for Information Processing

by

Effirul I. Ramlan

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
Department of Electronics and Computer Science

June 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Effirul I. Ramlan

Information processing applications driven by self-assembly and conformation dynamics of nucleic acids are possible. These underlying paradigms (self-assembly and conformation dynamics) are essential for natural information processors as illustrated by proteins. A key advantage in utilising nucleic acids as information processors is the availability of computational tools to support the design process. This provides us with a platform to develop an integrated environment in which an orchestration of molecular building blocks can be realised. Strict arbitrary control over the design of these computational nucleic acids is not feasible. The microphysical behaviour of these molecular materials must be taken into consideration during the design phase. This thesis investigated, to what extent the construction of molecular building blocks for a particular purpose is possible with the support of a software environment. In this work we developed a computational protocol that functions on a multi-molecular level, which enable us to directly incorporate the dynamic characteristics of nucleic acids molecules. To allow the implementation of this computational protocol, we developed a designer that able to solve the nucleic acids inverse prediction problem, not only in the multi-stable states level, but also include the interactions among molecules that occur in each meta-stable state. The realisation of our computational protocol are evaluated by generating computational nucleic acids units that resembles synthetic RNA devices that have been successfully implemented in the laboratory. Furthermore, we demonstrated the feasibility of the protocol to design various types of computational units. The accuracy and diversity of the generated candidates are significantly better than the best candidates produced by conventional designers. With the computational protocol, the design of nucleic acid information processor using a network of interconnecting nucleic acids is now feasible.

Contents

List of Figures

List of Tables

Acknowledgements

It has taken three years of undivided attention and dedicated work to produce this dissertation. In that three years, without the guidance, teaching and support of Dr. Klaus-Peter Zauner, I cannot imagine whether the completion of this research would ever be possible. To his guidance, teaching and support, I would like to express my sincerest gratitude. It is an honour to have the supervision and the privilege to learn from Dr. Klaus-Peter Zauner.

A special thanks to my examiners, Prof. Andrew Adamatzky and Dr. Srinandan Dasmahapatra for their evaluation of this work.

Through the hardship and difficulties, I would like to thank my wife with her support and patience. It is not easy to start our life as a family in a foreign place. However, you have managed to make it worked and patiently supported me through thick and thin. Also, my love to Aaqilah, whom always make me smiles after long hours of working. With her, things have not always been easy, but every minute of taking care of her has always been a great joy and pleasure.

I would like to thank my colleagues in the SENSe group for discussion and help in any area of my research. Last but not least, my family especially my parents, whom without them none of this would be possible. They have allowed me to choose my own path and supported me in any endeavours I have taken thus far. I am forever in debt to the love that they have given me.

“No birds soars too high, if he soars with his own wings”
- Sir William Blake

To my parents, my wife Hasniza and Aaqilah

Chapter 1

Introduction

The work in this chapter is adapted from the paper titled “Nucleic Acid Enzymes: The Fusion of Self-assembly and Conformational Computing“ which was first presented in the Unconventional Computing 2007 in Bristol and later appeared in the International Journal of Unconventional Computing 2009 (Ramlan and Zauner, 2009).

1.1 Biomolecular Computing Paradigms

With the feature size of solid-state devices approaching nanometer scale, molecules are coming increasingly into focus as an alternative material substrate for the implementation of information processing devices. A wide range of approaches to utilising molecules in computing are under consideration. The area of *molecular electronics* investigates possibilities for implementing with organic materials the architectures from silicon-electronics (cf. Petty et al., 1995). Polymer semiconductors and single-molecule transistors are typical research goals. In *chemical computing* excitable chemical reaction systems with diffusive coupling are investigated for their potential as massively parallel processing media (cf. Adamatzky et al., 2005). And *biomolecular computing* is concerned with the use of macromolecules and supramolecular systems and in many cases attempts to exploit mechanisms found in nature.

A prominent difference between solid-state materials and macromolecular materials is the large range of properties found in molecules. Biomolecules are mainly composed from only six (C, H, O, N, S, P) out of the 91 naturally occurring chemical elements. The number of possible compounds that could in principle be formed from these six atoms is very large. Even though there are many restrictions on how the atoms can be combined, stable macromolecules comprising hundreds or thousands of atoms can be formed. An application that is conceivable for nucleic acids computers but not plausible for conventional machines can be designed. The potential of biomolecules as a computing substrate in artificial devices has been investigated for over three decades (Zauner, 2005b).

Macromolecules occurring in organisms are typically formed from a set of building block molecules. These building blocks link through covalent bonds originating at specific atoms, but can be combined in arbitrary order. The twenty commonly occurring amino acids form such a set of building blocks. Linear polymers from up to a few hundred of these amino acids linked in arbitrary sequential order constitute an important class of biomacromolecules, the proteins.

Another set of building blocks found in nature are the nucleotides, which are combined, again in arbitrary order, to long nucleic acid molecules. The exact linear sequence of the building blocks may have a relatively small influence on the properties of the complete macromolecule, as is the case with the deoxyribonucleic acids (DNA), the carriers of genetic information in the cell. But the exact sequence can also be crucial to the properties of the macromolecule, as is typical for proteins. Both cases have practical advantages. The former is ideally suited for representing information, because the physical properties of the macromolecule are largely independent of its information content. The latter case gives rise to the diverse specificity and large range of material properties that is the basis of the tremendous variety of organisms seen in nature.

Two phenomena are key to the interaction and function of macromolecules: self-assembly and conformational dynamics. Both play also an important role for molecular information processing in nature and each serves as a paradigm for man-made molecular computing schemes. Figure 1.1 illustrates these paradigms.

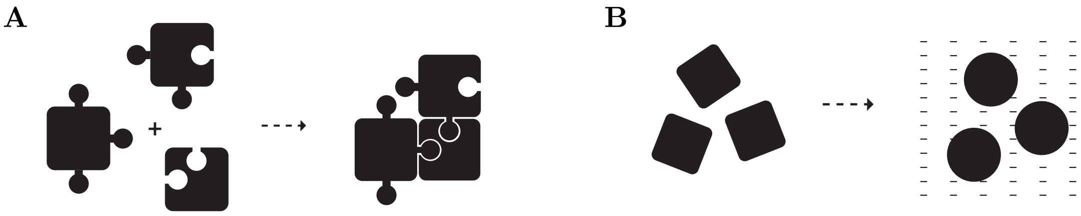


FIGURE 1.1: Cartoon of the two basic biomolecular computing paradigms. In *self-assembly computing* (A) information is encoded by molecular shapes. Molecules with complementary shape form supra-molecular clusters through non-covalent binding. Thus shape-encoded input is mapped into features of the cluster as output. In *conformational computing* (B) the physicochemical environment of a macromolecule serves as input signal. Intramolecular dynamics maps this milieu information into a change in conformational state.

Atoms attached through covalent bonds in a molecule can exert weak, short-range attractive forces (van-der-Waals interaction, hydrogen-bonds) on atoms in other molecules. If two macromolecules have complementary surfaces, i.e., surfaces that allow for close proximity of a large number of suitable atom-pairs, then the additive effect of the weak attractive forces results in a stable binding of the complementary molecules. In other words, the potential energy will dominate entropy even at high temperature. Molecules, on the one hand, are large enough to have specific shape features. On the other hand, they are small enough to be moved around by thermal motion and therefore can explore each others shapes by diffusion.

The self-assembly paradigm (Fig. 1.1A) effectively converts a symbolic pattern recognition problem into a free-energy minimisation process (Conrad, 1989, 1992, 1993). The self-assembly paradigm can conveniently be implemented with deoxyribonucleic acid (DNA) because it is relatively easy to predict the information among oligonucleotides from thermodynamic data and computer simulations. Over the past decade a number of experimental realisations of self-assembly computing have been reported (e.g., Adleman, 1994; Mao et al., 2000; Winfree, 2000). A drawback of self-assembly computing is that the random search of molecules for complementary partners by means of Brownian motion does not scale well to large reaction volumes.

In conformational computing (Fig. 1.1B) one attempts to exploit the shape changes that large macromolecules can undergo in response to their environment. The freedom of atoms in a molecule to rotate around single covalent bonds equips molecules with considerable flexibility. Proteins in particular have a distinctive agility that is core to their folding from a linear amino acid chain to a compact functional or structural component. This flexibility, however, does not terminate with the folding. The physicochemical milieu in which the macromolecule is embedded modulates the transition probabilities among the molecule's conformational states. Different conformational states commonly result in altered functional activity. A few experimental implementations that make use of the conformational dynamics have been reported (e.g., Hampp, 2000; Zauner and Conrad, 2001; Baron et al., 2006). In the conformational paradigm much of the computation is an intramolecular process and state changes can therefore be fast. However, a problem with this approach is that in practise the conformational effects are at least hard and often impossible to predict. The practical implementations so far rely on molecules occurring in nature or genetically engineered variants of these molecules (Hampp, 2000).

In nature's molecular information processing infrastructure both self-assembly and conformational dynamics play an important role. Typically both occur in combination. A protein may undergo a conformational change and as a consequence of this its shape becomes complementary to a region on another macromolecule thus leading to self-assembly. Conversely, a molecule that participates in self-assembly experiences a significant change in its environment as a result of the binding to another molecule and this change can give rise to an altered conformation. In combination self-assembly and conformational switching are a powerful set of primitives on which the entire molecular machinery of cells is built. It would be desirable to combine the self-assembly and the conformational paradigm also for artificial molecular computing schemes. In nature proteins are the key components that integrate self-assembly and conformational switching. Unfortunately, both phenomena are notoriously difficult to predict for proteins. However, an intriguing alternative has been experimentally demonstrated in form of nucleic acid enzymes, i.e., DNA (Stojanovic et al., 2002) or RNA (Penchovsky and Breaker, 2005) molecules with catalytic activity.

1.2 Computational Nucleic Acids Enzymes

Organisms have powerful and enviably efficient information processing capabilities. To a large extent these capabilities are conferred by macromolecules and their specific properties. The existence of these natural information processing architectures demonstrate that computing based on physical substrates that are radically different from silicon is feasible. Early suggestions for implementing a molecular computer with DNA followed the encoding principle of genetic information (cf. Liberman, 1979). This would require the formation and cleavage of numerous covalent bonds for their operation and thus require specific sets of enzymes. Major progress in the application of nucleotides for information processing came about two decades later with Adleman's insight that random oligonucleotides could be the basic tokens for information processing (Adleman, 1994). His method employed enzymes only to stabilise (through covalent bonds) the products of a self-assembly process (hybridisation of partially complementary oligonucleotides) but not in the information processing itself, and accordingly did not require enzymes with sequence specificity. This implementation by Adleman spawned the idea of building nucleic acids (DNA) computers as an alternate computing means that possess greater computational power then the conventional machines. However this view has changed as currently, nucleic acids computers are being designed and engineered to function inside a living cell (Rinaudo et al., 2007; Beisel et al., 2008; Win and Smolke, 2008; Isaacs et al., 2006).

The discovery of short RNAs' (which is 21-25 nt in length) role in regulating gene expression (Couzin, 2002) has sparked a strong interest in RNA molecules. In RNA interference, a short interfering RNA molecule (siRNA) forms base pair with a target region in mRNA, allowing a protein complex called RISC (RNA-induced silencing complex) to attach and cleaves the target region (Hannon, 2002; Petersen et al., 2006). Another short RNAs called microRNA (miRNA), generated from an enzyme named Dicer that cleaved non-coding RNAs (i.e., RNA that do not code protein), binds imperfectly with the target region in mRNA (forming a bulge) to prevent the translation machinery from accessing this target region (Zamore and Haley, 2005). In addition to these short RNAs, another complex folded RNA domain called riboswitches also play a role in gene regulation. Riboswitches sense the presence of specific metabolites and harness their conformational switching to activate the gene-control mechanisms in preventing the production of protein (Mandal and Breaker, 2004; Nudler and Mironov, 2004). The ability of these RNA molecules provides an interesting application scenario for molecular computing.

In nature proteins appear to play the preeminent role as molecular computing substrate. At the present state of technology, however, two other classes of biomolecules are more amenable to applications in man-made information processing architectures: deoxyribonucleic acids (DNA) and ribonucleic acids (RNA). The former offers a more limited conformational flexibility and concomitant less functionality, while the latter is less stable and requires more careful laboratory techniques. In general, the existing body of work on using nucleic acids for information processing can be grouped under three concepts:

Covalent Concept: Early proposals for the use of DNA in computing were inspired by the discovery of DNA's role in the storage of inheritable information and the astounding information density that can be achieved with molecular encoded data. All of these concepts require the formation and cleavage of specific covalent bonds which would require custom-designed proteins and are for this reason not practical (cf., e.g., Liberman, 1979).

Complement Concept: In Adleman (1994) the influential suggestion to use arbitrary nucleotide sequences and the hybridisation with their complementary sequences instead of covalently linked individual bases was made and a practical demonstration of this approach was given. This idea moved the burden of recognising tokens of information from proteins (which up to now cannot be designed for purpose) to the self-assembly of short nucleotide sequences which can be designed with desired self-assembly properties and can be synthesised with ease.

Conformational Concept: Whereas in the above two concepts the conformational flexibility of the nucleic acids is irrelevant or even undesirable, more recently computing concepts that exploit the change in conformation a nucleic acid undergoes upon hybridising with another nucleic acid molecule have been developed (Stojanovic and Stefanovic, 2003b; Penchovsky and Breaker, 2005).

Thus far, information processing systems constructed with nucleic acids components have shown promising prospects (Shapiro and Gil, 2008). The fusion of the two concepts, complementary (self-assembly) and conformational has been demonstrated in *in-vitro* environment and both concepts employ a relatively similar method of implementation. The concept of allosteric ribozymes is illustrated in Fig. 1.2. Input signals are encoded as small molecules of DNA strands which effect a computing machinery for processing that combines both functional nucleic acids (ribozymes for RNA and deoxyribozyme for DNA) and receptor units for input signal detection. Key to this approach is the possibility to control the activity of a ribozyme or deoxyribozyme with oligonucleotides as input. Such allosterically controlled nucleic acid enzymes have been investigated as sequence specific biosensors, where they have the advantage over molecular beacons that they catalytically amplify the recognition event (Kuwabara et al., 2000). Within certain constraints, the base sequence for the binding site of the control oligonucleotide (labelled OBS in the figures) can be chosen independently of the sequence on which the nucleic acid enzyme will act. It is therefore possible to have an oligonucleotide sequence start (or stop) the production of another, largely independent, oligonucleotide sequence. Moreover, it is possible to engineer nucleic acid enzymes to be controlled by more than one oligonucleotide.

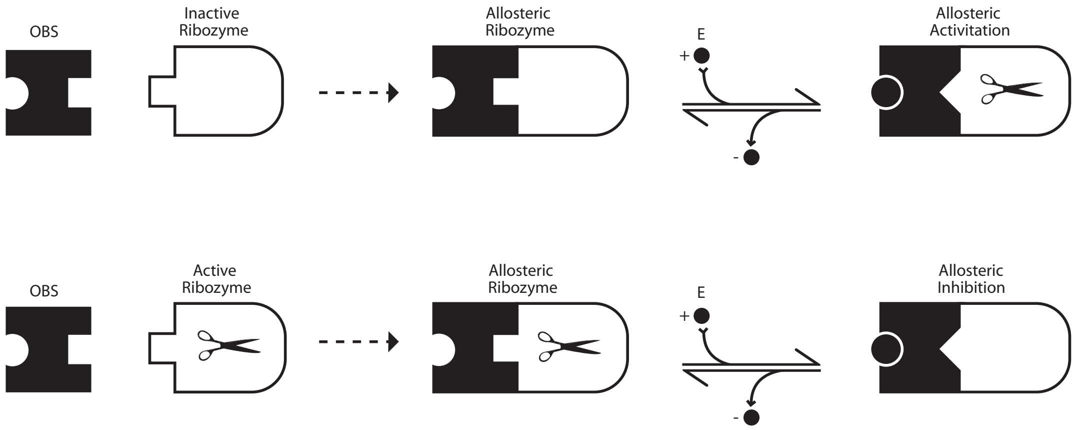


Figure 1.2: Allosterically activated ribozyme (top) and allosterically inhibited ribozyme (bottom) (Soukup and Breaker, 1999; Silverman, 2003). The allosteric ribozyme is composed of two components (left of the dashed arrow), a oligonucleotide binding site (OBS) and a ribozyme part. The two components are covalently bound and from a single nucleic acid molecule (centre). Upon binding an effector oligonucleotide (E) the conformation of the binding site changes and affects the conformation of the ribozyme component. The latter conformational change will activate (top) or inhibit (bottom) the catalytic activity of the ribozyme part. The same scheme can also be realised with deoxyribozymes. The scissors symbolically represents the cleavage reaction of the ribozyme.

Following this approach, one is likely to construct simpler information processing units that can be integrated into a network, where output from one unit can be used as input for another corresponding unit. The *in-vitro* demonstrations of such networks have been shown by Stojanovic et al. (2003) and Penchovsky and Breaker (2005). The nucleic acid constructs are employed to solve simple arithmetic operations (Stojanovic and Stefanovic, 2003a; Lederman et al., 2006) and capable of handling task that requires the integration of several different types of molecular gates with a common set of input and substrates molecule (Stojanovic and Stefanovic, 2003b; Pei et al., 2006). Thus far, the construction of a network comprising more than 100 nucleic acid molecular gates has been reported (Macdonald et al., 2006), suggesting that, the development of highly regulated molecular networks able to support complex decision-making criteria is feasible.

Computational nucleic acids are constructed as modular tuneable units, where different components of one system can be substituted with alternative parts as demonstrated in proof-of-concept models of simple computational units (Stojanovic and Stefanovic, 2003b; Penchovsky and Breaker, 2005) and in devices integrated within living cells (Rinaudo et al., 2007; Beisel et al., 2008; Win and Smolke, 2008). The tuning of the computational units (i.e., ribozyme and the receptor sites) is achieved by mutating bases in certain regions of the nucleic acids guided by energetic information that can be calculated from the sequence to structure mapping of the molecules. In a similar manner, one can allocate sequence constraints that can enhance sequence specificity for any particular part of the computational units. In designing sets of nucleic acids for information processing, one typically has a desired molecule conformation and additional local constraints specific to certain regions of the molecules. For example, a binding site for an effector molecule (i.e., a nucleic acid that will affect the activity of a functional nucleic acid) may be required to be complementary to a sequence released in a preceding step. For nucleic acids, basic folding tools (sequence to structure) and sequence generators (sequence from structure) are available to handle the design requirement mentioned here.

The application of allosterically controlled nucleic acids in bioimmersive computation has the potential to open up interesting possibilities. Smart drugs that can sense the internal state of cell and intervene in the intracellular regulatory mechanisms may come within reach (Benenson et al., 2004) and engineered molecular control mechanisms that can be integrated into cells would be a powerful tool for life-science research (Simpson, 2004). Before the potential of these long-term aims can be realised many obstacles in the laboratory need to be tackled and much better computational design procedures are required. A crucial issue will be the prediction of the interactions within complex mixtures of molecules. At present folding simulators for multiple interacting RNA strands are at their infancy and simulation tools capable of predicting DNA-RNA interactions do not exist. There is a need for a general methodology and supporting computational tools to create purpose-designed sets of interacting computational nucleic acids. This *in-silico*-first approach will enable designers to specify the physiological conditions plus additional constraints that should aid in the construction of well-defined computational units, and reduce the cost and time required in the laboratory.

1.3 Thesis Outline

Nucleic acids are an attractive computing substrate for three reasons. Firstly, they support both basic paradigms of molecular computing: self-assembly and conformational switching. Secondly, the intramolecular and intermolecular hybridisation of nucleic acids can be predicted reasonably well with existing computational tools. Thirdly, nucleic acids play a very important role for memory (Dietrich and Been, 2001) and control (Blencowe and Khanna, 2007) in every living cell. Ribonucleic acids are challenging to work with in the laboratory, however. Yet, these challenges are at present more manageable than the computational challenge one would face if one would attempt to design information processing components with proteins.

As discussed in the previous section, computational nucleic acids are constructed as modular tuneable units, where any parts of the system can be substituted, to either increase likelihood of structure folding into a desired confirmation or improve sequence specificity (i.e., hybridisation or self-assembly between two molecules). Computer aided design of a nucleic acids information processor and its experimental validation has been demonstrated by Penchovsky and Breaker (2005). This work highlights the value of prediction tools in generating a pool of well-defined candidate molecules to reduce the complexity and effort of direct implementation in the laboratory. Despite the success, this work only partially exploits the computational prediction tools available and the methodology implemented is too specific for the general purpose-design of sets of molecules for information processing. In this thesis we develop a computational protocol to support potentially with constraints on the structure or the sequence the design of molecules capable of performing information tasks. However, before we could arrive at this, a set of computational tools needs to be developed. These tools should be able to predict the secondary structure formation of multiple interacting nucleic acid molecules as well as be capable of designing sequences that conform to the target structure of these interacting molecules. The development of these tools will be addressed in this thesis accordingly.

We present in Chapter 2, a general overview of nucleic acids and discuss more thoroughly the implementation of functional nucleic acids as computational substrate through the combination of both the self-assembly and conformational dynamics paradigms. In Chapter 2, we also review the computational biology of nucleic acids by describing the essential prediction tools that are already available. As of now, the majority of the tools described are not intended for the construction of computational nucleic acids. They have been developed specifically to gain understanding of naturally occurring DNA and RNA molecules. Despite the different motivation, we identify certain tools that are suitable for the design of computational units and described their algorithms in detail.

As a first step we extended the standard notation for RNA to a string representation for both DNA and RNA that is simple yet sufficient enough to depict constraint assignments and to describe binding among molecules (Chapter 3). The commonly used string representation lacks these features and is generally quite lengthy for describing the structures of nucleic acids. While this notation is convenient, it does not show the secondary structure in a more readily understandable form. We have therefore also developed a drawing tool that rendered nucleic acids structures. This tool is capable of rendering the intermolecular binding reactions that occur between two or more molecules, a feature currently non-existence in any other nucleic acids drawing tool. The ability to illustrate intermolecular binding among molecules allows us to show, for instance, the conformational change undertaken by a functional nucleic acid unit triggered by an oligonucleotide (short nucleic acid) binding to a receptor region of the unit. Our string representation is able to express these computational phases. Using this representation as input, our drawing tool translates this string into a conventional structure representation to get better understanding.

In order to develop a general computational design procedure, it is appropriate that we focus on the structure to sequence mapping of nucleic acids (known as "inverse prediction" in the computational biology field). Chapter 4 investigates the problem of generating a sequence that plausibly folds into a desired structure by comparing the performance of current sequence generator algorithms within a defined design space. This design space is restricted to a parameter envelope that includes all known computational nucleic acid units. Our main objective in Chapter 4 is to find the sequence generator best suited to generate sequences in this design search space. From there, we investigate the possibility of improving this generator in order to develop a custom sequence generator that performs efficiently in the specified space. Chapter 4 serves as a foundation for the development of a multiple conformations sequence generator for interacting molecules.

We go on to develop an entirely new algorithm focused on the problem of designing a nucleic acid sequence for a given set of structures representing a molecule in different conformational states. Furthermore, we extended the problem to includes multiple molecule interactions with multiple conformational states. To evaluate the performance of our new algorithm, we conduct a comparison study against the one existing multiple-conformation algorithm. We validate our multiple molecule multiple state algorithm against nucleic acids molecules described in the literature that provide cases of multiple molecules interacting with multiple conformational states because no other algorithm exist that could be used for comparison. Through the development of this new algorithm, we have provided the basis for a general approach to the construction of computational nucleic acids. The effectiveness of this methodology and specifically the newly developed algorithm is investigated in the Chapter 6. In this chapter we constructed all two-input molecular gates to analyse the efficiency of our design methodology. We show the significant difference our new approach accomplished against two existing computational procedures.

Finally, in Chapter 7 a summary of the work conducted is presented first. This is followed by critical discussions on the effectiveness of the newly developed algorithm in designing nucleic acid sequences that conform to multi-stable conformations of interacting molecules, the ability of the new algorithm in directing the generation of sequences according to a desired free energy profile. In the discussion, we highlight some of the flaws of the newly developed algorithm, specifically regarding its processing time and accuracy of the solution. Furthermore, we discuss our computational procedure and demonstrate its ability to design a few basic nucleic acid information processing units. Chapter 7 also discusses the problematic issue of validating computational tools without wet-laboratory experimentation, and suggests future directions.

Chapter 2

Research Background

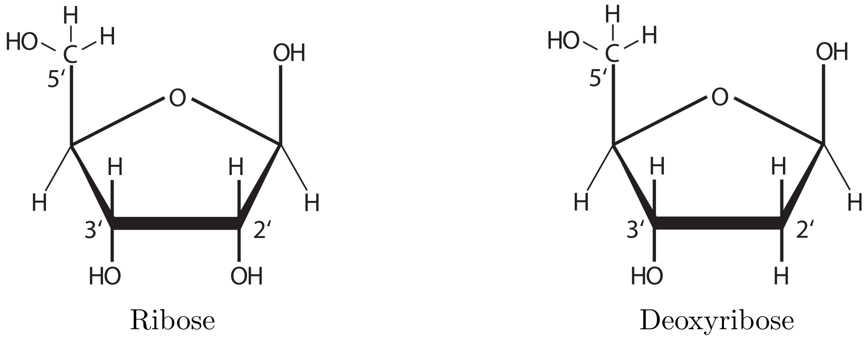
The work in this chapter is adapted from the paper titled “Nucleic Acid Enzymes: The Fusion of Self-assembly and Conformational Computing“ which was first presented in the Unconventional Computing 2007 in Bristol and later appeared in the International Journal of Unconventional Computing 2009 (Ramlan and Zauner, 2009).

2.1 Properties of Nucleic Acids

Nucleic acids are macromolecules that play an important role as information carriers in cells. Two types occur, ribonucleic acids (RNA) and deoxyribonucleic acids (DNA), which are named after the structure of a sugar component always present in these molecules (Fig. 2.1). RNA and DNA are typically long linear polymers that consist of a large number of monomers taken from a set of four different nucleotides. The sequential order in which these nucleotides are interlinked in the nucleic acid molecule can represent information.

The two types of nucleic acids play different roles. RNA has the task of transmitting information within the cell, while DNA transmits information from generation to generation. The genetic information is encoded in a dimer of two complementary nucleotide chains (’single-stranded’ DNA) which upon self-assembly assumes the well known double-helical structure (’double-stranded’ DNA). DNA is well suited as carrier of genetic information because of its energy degeneracy with respect to the sequential order of the nucleotides. The properties of RNA molecules are more dependent on the sequence of nucleotides and as a consequence RNA takes on additional roles in the cell aside from representing information.

Each of the monomer units that make up nucleic acids consists of a sugar moiety, a phosphate group, and a base. The sugar component of the monomers in RNA molecules is ribose, hence the name *ribonucleic acid*. Correspondingly, DNA is named *deoxyribonucleic acid* after its sugar component deoxyribose. Figure 2.1 shows the chemical structures of both sugar components.



Ribose

Deoxyribose

FIGURE 2.1: The sugar compounds that form the backbone of RNA (left) and DNA (right). Note the lack of the oxygen at the $2'$ carbon in the right panel as compared to the ribose (left). The name *deoxy*nucleic acid refers to the absence of these oxygen atoms in the backbone of DNA (Bloomfield et al., 2000).

The hydroxyl group (-OH) at the $2'$-carbon in ribose is not present in deoxyribose (Berg et al., 2003). The consequence of this structural difference is twofold. DNA is considerably more stable against hydrolysis and forms more compact double strands, while RNA has more conformational flexibility (Bloomfield et al., 2000). The flexibility of macromolecules to change their three-dimensional shape, i.e., their *conformation*, while maintaining the covalent bonds among atoms, i.e., their *configuration* unchanged, is the basis of conformational computing. The chemical stability of DNA and the structural flexibility of RNA are both desirable properties for the molecular computing based on nucleic acid enzymes. Which type of nucleic acid is preferred for the implementation of a particular molecular component will often depend on the trade off between stability and flexibility.

Within either RNA or DNA the sugar moieties and the phosphate groups of all monomers are identical. The base, which forms the third component of each monomer, provides the variety requisite for representing information in a sequence of monomers. Each monomer unit carries one of four possible bases. In RNA these are adenine, guanine, cytosine, and uracil, abbreviated as A, G, C, and U. The first three of these bases also occur in DNA, but instead of uracil DNA contains thymine (T). This difference is thought to be of use for DNA repair mechanisms that actively maintain the integrity of a cell's genetic information, but is of no relevance within the context of the present thesis.

Of crucial importance for the interaction of nucleic acid molecules is the *complementarity of bases*. The base of a nucleotide can form weak bonds, called hydrogen-bonds, with another nucleotide that carries a complementary base. Hydrogen bonds occur between a hydrogen atom bound to an electronegative atom, and another electronegative atom. They are roughly 20× weaker than a covalent bond. Among the four possible bases that can occur in a nucleotide, T or U can bind to A with two hydrogen bonds and G can bind to C forming three such bonds . Two nucleotide strands with complementary base sequence will form a dimer that is held together by the additive effect of the hydrogen bonds that can be formed between the complementary bases. This process is called *hybridisation.* The direction of the sequence has to be taken into account if complementarity is considered. The two strands that form a double helix are intertwined running in opposite direction. To indicate the orientation of a single stranded nucleic acid, its ends are named after the unbound carbon atom in the sugar moiety as $5'$ at one end and $3'$ at the other. As a convention, the notation of nucleic acid sequences is written from left to right in $5'$ to $3'$ direction, i.e., ATTGC always stands for $5'$–ATTGC–$3'$ (Berg et al., 2003). In the following figures a diamond symbol ($\diamond$) indicates the $3'$-end of a strand. If a nucleic acid has a sequence that is complementary to itself, then it can fold back onto itself and form an intramolecular double-helix (SantaLucia and Hicks, 2004). Partial intramolecular hybridisation can result in a complex three-dimensional structure of the molecule. In some instances the three-dimensional structure confers functionality such as a specific catalytic activity.

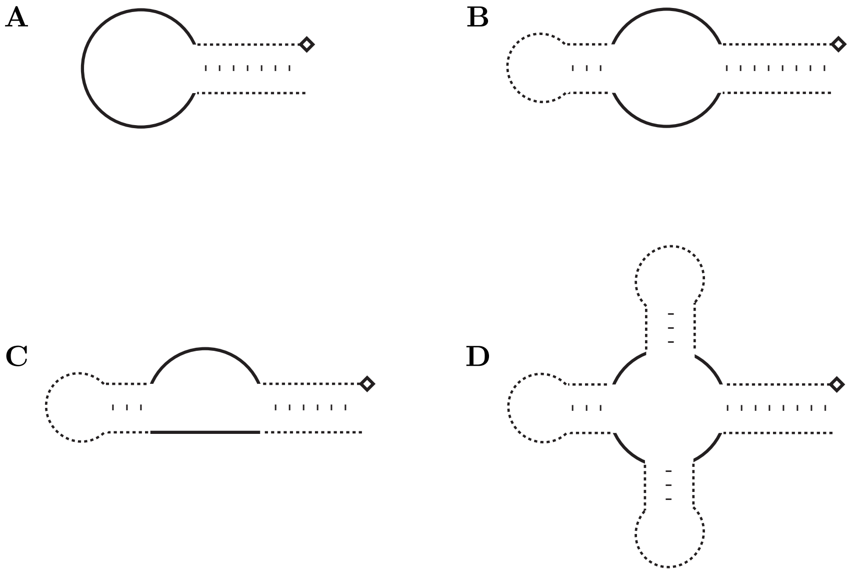


Figure 2.2: Classification of RNA loop motifs; the named motif is shown with solid lines. Hairpin loop (A), internal loop (B), bulge (C), multi-branch loop (D; a four-way junction is shown). A hairpin loop with the adjacent stem is referred together as stem loop. After (Tinoco and Bustamante, 1999).

As mentioned above, RNA is more flexible than DNA and as a consequence it forms intramolecular base-pairs more readily. A single stranded RNA molecule can bind to itself in several regions with the unbound segments present as loops between bound segments or dangling ends. The loops can be grouped into four classes illustrated in Fig. 2.2. Due to its higher flexibility, in addition to the pairing of complementary bases (A-U/U-A, C-G/G-C), the 'wobble pairing' of G-U (and, reverse oriented, U-G) through two hydrogen-bonds also contributes to the structural variability exhibited by RNA molecules (Varani and McClain, 2000).

For a given RNA sequences ('primary structure'), there is often a diverse set of secondary structures it can fold into. Which structure is favoured will depend on the environment of the molecule, for example, the presence or absence of other molecules or ions. Conversely, a diverse set of sequence configurations can yield a particular secondary structure (Draper, 1996; Zuker, 1989). Subsequently, interactions among secondary structure motifs lead to the formation of a tertiary structure which in some cases entails functionality. Determining the secondary structure of RNA sequences becomes an integral part before predicting the tertiary structure as helices formed in secondary phase tends to be stronger than the tertiary interactions that connect the element of RNA loop motifs (Crothers et al., 1974; Banerjee et al., 1993; Tinoco and Bustamante, 1999). Such functional RNA molecules are discussed in the next section.

2.2 Functional Nucleic Acids

Biological catalysis was thought to be synonymous with catalytically active protein, i.e., enzymes, until RNA molecules with catalytic capability were discovered (Altman, 1990). These *ribozymes*, as the RNA enzymes are also called, led to the hypothesis that precursors of the cell may have relied exclusively on RNA for both transmission of genetic information and metabolism, tasks which are in present cells relegated to DNA and protein, respectively (i.e., the "RNA world" hypothesis (Gilbert, 1986)). Although it appears unlikely that DNA has any catalytic function in nature, it is possible to produce DNA enzymes in the laboratory (Breaker, 1997).

Ribozymes can be categorised according to size and catalytic activity (Doudna and Cech, 2002; Symons, 1992). The three classes of ribozyme are small catalytic RNAs, group I and II introns, and Ribonuclease P (RNase P). Small catalytic RNAs range in size from 40–160 nt (nucleotides) and are self-cleaving molecules. Group I introns are self-splicing RNA molecules over 700 nt in length while group II introns are more than 1500 nt long. These self-splicing RNAs are found in unicellular organisms. RNase P is over 500 nt long, occurs in all cells, and is required for the production of transfer RNA (tRNA), a key component of the cellular machinery for protein synthesis (Altman, 1990; Doudna and Cech, 2002).

Small catalytic RNAs are the most attractive with regard to molecular computing applications. The group of small catalytic RNA comprises hammerhead ribozymes (Birikh et al., 1997; Scott, 1999), hairpin ribozymes (Doudna and Cech, 2002), the hepatitis delta virus (HDV) ribozyme (Ferré-D'Amaré et al., 1998), and the Neurospora Varkud Satellite (VS) ribozyme (Lafontaine et al., 2001). Each of these ribozymes has a distinct structure. Nevertheless, all of them catalyse the same reaction. They cleave the phosphodiester bond in RNA, generating a 5′-product with a 2′, 3′-cyclic phosphate terminus and a 3′-product with a 5′-hydroxyl terminus. It is thought that the 2′ hydroxyl group of the ribose moiety of RNA participates in the catalysis (Fedor and Williamson, 2005), however DNA can also act as a catalyst as will be discussed later in this section.

Most of the known natural occurring ribozymes catalyse intramolecular (also called *in-cis*) reactions in which the ribozyme cuts and detaches from part of its own sequence (Forster and Symons, 1987). However, some ribozymes have been successfully modified to split other nucleic acids. To avoid ambiguity, we will use the term *ribozyme core* to refer to the catalytically active RNA molecule in intermolecular (*in-trans*) reactions. Such a reaction is illustrated in Fig. 2.3. The figure shows the sequence of reaction steps in the catalytic cycle of a hammerhead ribozyme. For brevity the release of both products is shown as a single step, however, the products are likely to dissociate from the ribozyme core one after another (Long and Uhlenbeck, 1993). The turnover rate of small ribozymes is typically about 1 cleavage per minute (Wedekind and McKay, 1999; Doudna and Cech, 2002).

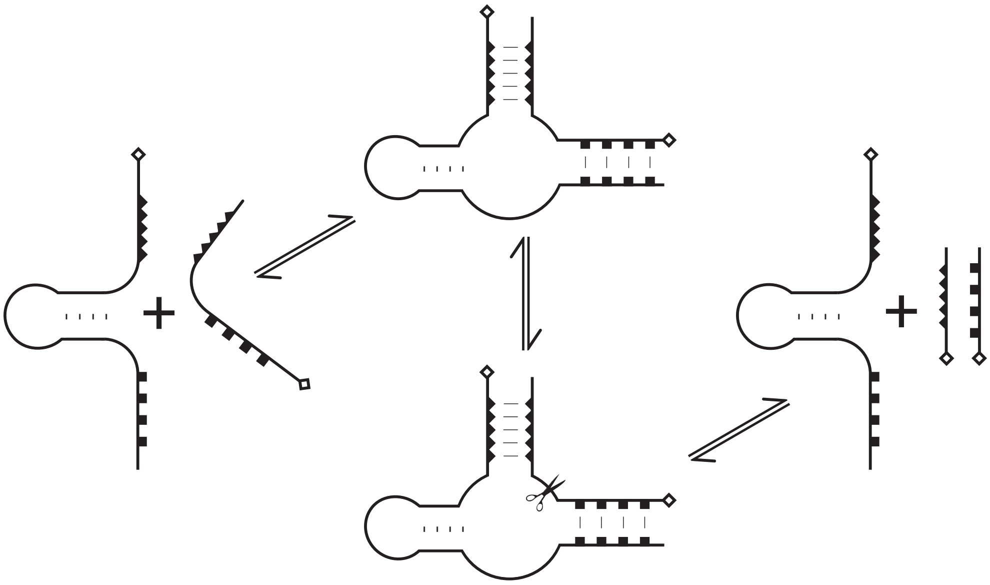


FIGURE 2.3: Splitting of an RNA molecule catalysed by another RNA molecule (Long and Uhlenbeck, 1993). The catalytic RNA binds a substrate RNA molecule if it is complementary to the two hybridisation regions indicated by squares and triangles (left). At the location indicated by scissors the substrate molecule is cut (centre). After the two reaction products dissociate from the catalytic RNA, the latter is ready for another reaction cycle (right).

Among the small catalytic RNAs the Neurospora Varkud Satellite (VS) ribozyme has the largest core with a length of with 150 nt. It is followed by HDV ribozyme with a core of at least 90 nt. In both the tertiary structure appears to play an important role for their catalytic function. Even shorter cores have been found in ribozymes of plant viroids and virusiods undergoing site-specific, self-catalysed cleavage as part of the replication process (Forster and Symons, 1987; Symons, 1997). Hairpin ribozymes can have cores as short as 70 nt, although in nature they are part of a four-way junction (cf. Fig. 2.2) (Walter and Burke, 1998). The smallest known natural ribozyme cores are of the hammerhead type and can be as short as 40 nucleotides Forster and Symons (1987). Still, smaller ribozyme cores have been engineered. So called minizymes, derived from hammerhead ribozymes can be as short as 22 nt, but the reduced size comes at a cost in catalytic efficiency (McCall et al., 1992).

For some of the ribozyme cores it is feasible to control their catalytic activity. This property is key to the application of ribozymes in molecular computing. In order to understand the mechanisms of controlling the activity of the ribozymes it is useful to consider their secondary structure. The secondary structure that emerges from an RNA sequence is composed from the motifs in Fig. 2.2 and possibly dangling single-stranded ends. The different types of ribozymes are distinguished by their characteristic combination of loops and helices (Lilley, 1999).

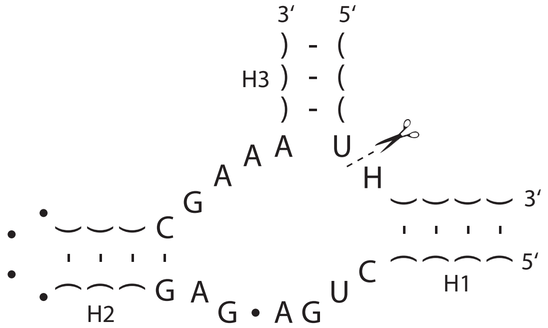


Figure 2.4: Minimal functional structure of hammerhead ribozyme. Three helical stems (H1, H2, H3) emanate from a junction on the ribozyme core (Hertel et al., 1994; Symons, 1997). In nature, always either helix H1 or H3 is terminated by a hairpin loop which results in intramolecular catalysis. Hammerhead ribozymes that catalyse the *in-trans* reaction, as depicted in the figure, can be made synthetically (Birikh et al., 1997). The core region has a specific sequence for all known active structures and is therefore termed 'conserved'. Conserved bases are specified explicitly, with H representing any one of {A, C, U}. A dot (•) stands for any base that will not cause hybridisation in this position; correspondingly two parentheses connected by a dash indicate an arbitrary pair of complementary bases. Hammerhead ribozyme cleaves the substrate strand that binds to form H1 and H3 as symbolically represented by the scissor and dashed lines.

The secondary structure of a hammerhead ribozyme is depicted in Fig. 2.4. Hammerhead ribozymes require the presence of a metal ion (typically Mg^{2+}) to be catalytically active (Doudna and Cech, 2002). A ribozyme with a different structure, the so called hairpin ribozyme, is shown in Fig. 2.5. Within the context of molecular computing, the two ribozymes illustrated in Figs. 2.4 and 2.5 appear suitable as the enzymatic core for the construction of allosterically controlled ribozymes. A discussion of allosterically controlled nucleic acids enzymes as a means for information processing is presented in the following section.

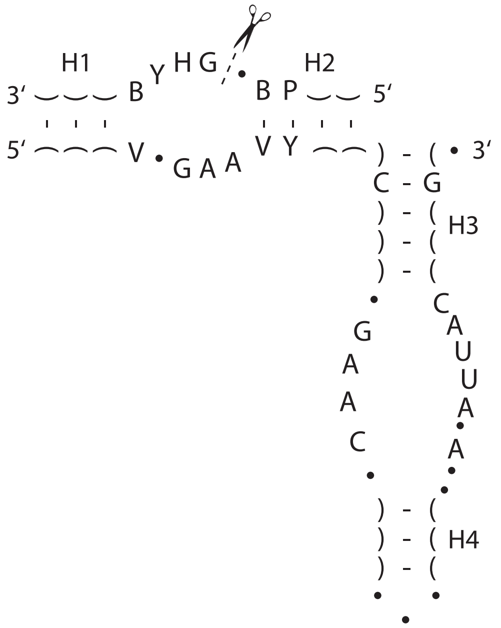


Figure 2.5: Minimal functional structure of hairpin ribozyme. Two domains are distinguished each of which contains two helices separated by a conserved internal loop. One domain is drawn horizontally and includes the substrate which binds to the core by forming helix H1 and helix H2. The other domain is drawn vertically and contains the helices H3 and H4 (Fedor, 2000; Walter and Burke, 1998; Porschke et al., 1999). The following notation is used for the conserved region: Y∈{C, U}, V∈{A, C, G}, B∈{C, G, U}, and H∈{A, C, U}. See Fig. 2.4 for the explanation of dots and parenthesis.

It is generally believed that the conformational flexibility of RNA is important for the catalytic process itself (Doherty and Doudna, 2000; Hohng et al., 2004). The conformational flexibility of RNA gives also rise to a large variety of secondary structures. The secondary structure consists of single stranded regions alternating with double stranded regions where stretches of the RNA molecule binds to itself (cf. Fig. 2.4). On the other hand, one can arrive at secondary structure conformation through a self-assembly process involving multiple molecules. Figure 2.5 depicts three distinct RNA strands that intermolecularly bind to form a unique secondary structure. The secondary structure motifs interact and form the three dimensional tertiary structure of the RNA molecules.

The conformational flexibility of RNA supports a diverse set of functional roles (Nagai and Mattaj, 1994). The structural variety of RNA and its concomitant functional diversity make RNA also a suitable medium for directed in-vitro evolution (Joyce, 1992). This technique is based on the possibility to copy RNA molecules with aid of protein enzymes. Errors in the copy process yield a population of RNA molecules with slightly varied sequences. Repetitive application of this error-prone replication process will lead to an evolution of the population of molecules. In the absence of other selection pressures, the evolution would favour molecules that are most efficiently reproduced by the participating protein enzymes. However, a selection step can be introduced to assert

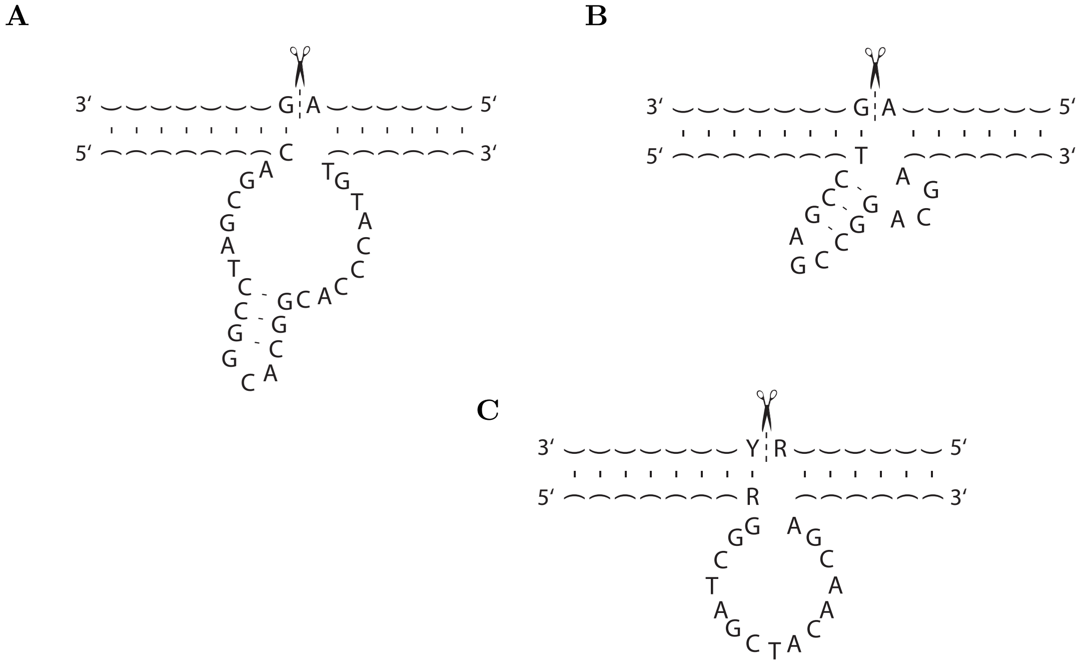


Figure 2.6: Secondary structures of three deoxyribozymes. Panel A shows a deoxyribozyme that resembles the general structure of the hammerhead ribozymes (cf. Fig. 2.4). It is characterised by a specific ('conserved') region of 15 nt connected to a stem-loop and is capable of cleaving substrates that contain a G-A-joint. A deoxyribozyme with a different structure, but also applicable only to substrates sequences with a G-A-joint is shown in panel B. The deoxyribozymes in panels A and B both have been applied for information processing (Stojanovic et al., 2002). Panel C shows a deoxyribozyme that is less constrained in the substrate junction it will cleave (Santoro and Joyce, 1997). The notation for the binding region is: Y∈{C, T} and R∈{A, G}.

evolutionary pressure in another direction. The molecules could for example be selected by their binding capabilities towards a particular substrate molecule (Famulok and Szostak, 1992). A number of ribozymes have been produced through directed evolution (Breaker and Joyce, 1994b; Chapman and Szostak, 1994). The majority of them possess a ribozyme core that does not resemble any of those found in nature (Tang and Breaker, 2000). Directed evolution provides a technique to enrich the repertoire of RNA structures amenable to molecular computing applications.

Directed evolution can also be applied to DNA and, rather surprisingly yields DNA molecules with enzymatic activity, so called deoxyribozymes (Breaker and Joyce, 1994a, 1995; Santoro and Joyce, 1997; Joyce, 2004). DNA is best known as a memory molecule inscribed with information crucial for the production of macromolecular components in cells. The properties that make DNA suitable for this function are its stability, and reliable hybridisation, but also the fact that DNA forms a double-helical structure largely independent of the sequence of bases as long as the two strands that hybridise are complementary. These properties together with the absence of DNA enzymes in nature had let to the view that DNA is not flexible enough to act as a catalyst. It is now, however, well established that DNA does have the structural flexibility to support a range of secondary and tertiary structures (Seeman, 2003) and can form a diverse set of tertiary structures with a potential to function as catalysts (Breaker, 1997). Secondary structures of three deoxyribozymes developed through the process of in-vitro selection are depicted in Fig. 2.6. For the deoxyribozymes shown in panels B and C of Fig. 2.6 it was found that their catalytic reaction rates are comparable to those of ribozymes (Emilsson and Breaker, 2002). As mentioned above, hammerhead ribozymes require the presence of metal ions to be catalytically active. The deoxyribozymes also require metal ions. The first deoxyribozyme was designed in the presence of Pb^{2+} as co-factor (Breaker and Joyce, 1994a) and the deoxyribozymes shown in Fig. 2.6 all require Mg^{2+}.

From an application perspective the use of DNA has the advantage over RNA that DNA molecules are generally more stable. Furthermore, the DNA-DNA-binding is more reliable and results in higher specificity. Given these practical advantages of DNA and the fact that DNA enzymes do not occur in nature it is of particular interest that recently a ribozyme was successfully converted into a deoxyribozyme by means of directed evolution (Paul et al., 2006). A DNA sequence that corresponded (apart from the T for U substitution) to a known ribozyme which catalyses a covalent bonding between two RNA oligonucleotides was found to be inactive. However, after acquiring suitable mutations during directed evolution, a deoxyribozyme that also catalyses a covalent bonding between two RNA oligonucleotides—though at a lower efficiency and different bond location—was arrived at.

In combination the capability of self-assembly through hybridisation of complementary sequences and the conformational flexibility to form sequence dependent spatial structures with catalytic activity make nucleic acids an attractive material for molecular computing.

2.3 Nucleic Acid Enzymes as Computing Substrate

From the time it became apparent that nucleic acid polymers carry the genetic information in their base sequence, its astounding information density was recognised. Since then, the implementation of nucleic acids as information processing substrate has grown from covalent concept introduced by (Liberman, 1979) to exploitation of complementary base pairing binding by (Adleman, 1994) and currently, the combination of both, self-assembly (base-pairing binding) and the conformational flexibility of nucleic acids enzymes (Stojanovic and Stefanovic, 2003b; Penchovsky and Breaker, 2005). In the meantime, the objective of biomolecular computing has also shifted from constructing alternate computing machines to replace conventional computers in high complexity tasks to the development of biomolecular computing units that can function inside a living cell (Shapiro and Gil, 2008). Earlier in section 1.2, we discussed the concept of allosteric nucleic acids enzymes that combine both the self-assembly and the conformational dynamics paradigm for constructing computational nucleic acids units. This fusion of both paradigms has been demonstrated *in-vitro* by Stojanovic et al. (2002); Stojanovic and Stefanovic (2003a,b); Penchovsky and Breaker (2005).

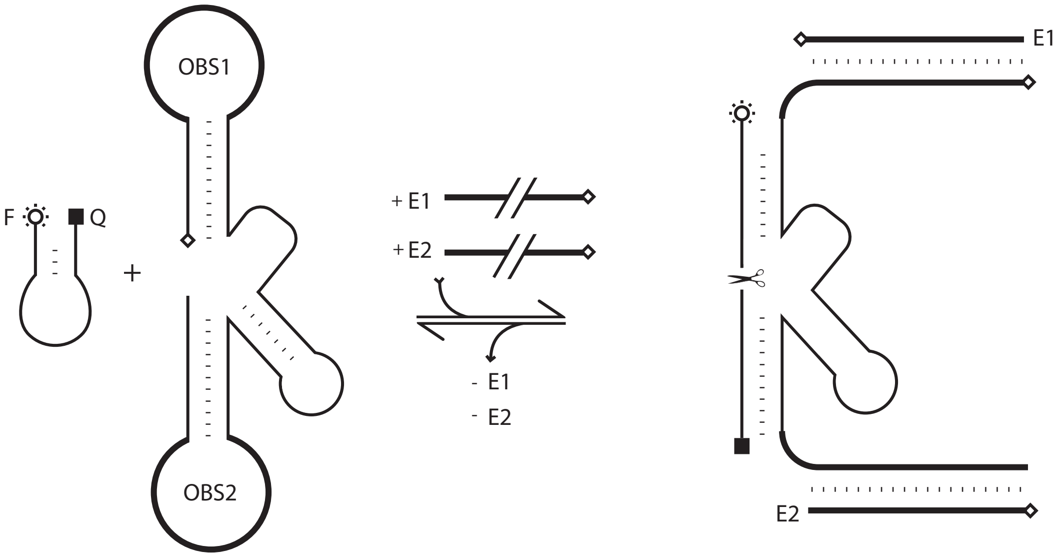


Figure 2.7: Deoxyribozyme acting as a logic AND gate after (Stojanovic et al., 2002). The molecule is designed in such a way that it can self-hybridise to block its own substrate binding site. This self-hybridisation is weaker than the binding of the effector molecules (E1, E2) to their oligonucleotide binding sites (OBS1, OBS2). Only in the presence of both effector molecules is the substrate binding site accessible and accordingly the deoxyribozyme catalytically active. By supplying a molecular beacon (far left) (Stojanovic et al., 2001) as substrate the output of the gate can be determined optically. If the deoxyribozyme is catalytically active it will cleave the beacon molecule, thus separate the quencher (Q) from the fluorophore (F), and consequently give rise to a fluorescence signal.

In general, an allosterically controlled nucleic acid enzyme comprises of two separate entities, a nucleic acid enzyme and a receptor unit, coupled together to form a single molecule. Allosteric nucleic acid enzymes can be purposely designed to have multiple stable conformational states. As an example, in the initial state, the combination of the nucleic acid enzyme and receptor unit folds into a confirmation that disrupt catalytic activity by binding the conserved bases of the enzymatic-core with complementary bases allocated in the molecule. The introduction of another smaller molecule then triggers a conformational change that leads to a folding which releases the conserved bases from base-pairing and thus activates the catalytic core. There are various control strategies that can be applied in designing these molecules. For instance, the molecule shown in Fig. 2.7 was designed by adding an allosteric control to the deoxyribozyme shown in Fig. 2.6B (Stojanovic et al., 2002). It is inactive unless two effector molecules with specific base sequences are present. The behaviour of the molecule can be interpreted as an AND logic gate. We note, however, that the possibility to catalyse the production of oligonucleotides as output signal with a base sequence independent of the sequences that serve as input signals (effector molecules) allows for applications other than logic AND operations.

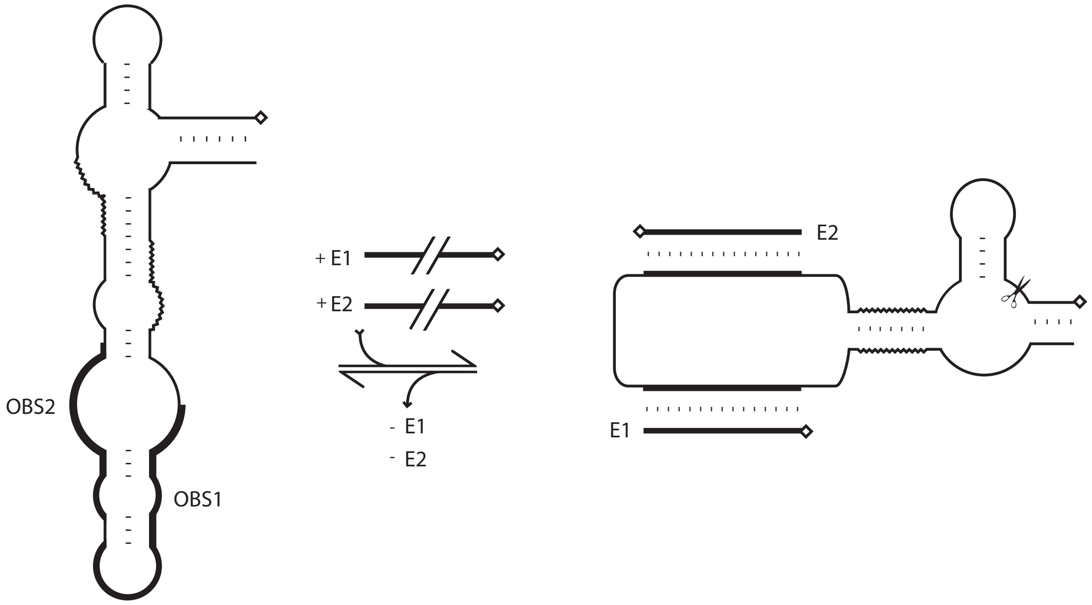


FIGURE 2.8: Two-input molecular switch based on allosterically controlled hammerhead ribozyme after (Penchovsky and Breaker, 2005). In the absence of effector molecules (E1, E2) the inactive conformation (left) is more stable. Upon binding of the effector molecules to their corresponding oligonucleotide binding sites (OBS1, OBS2) the ribozyme changes into a catalytically active conformation (right). Crucial for the formation of the hammerhead conformation is the correct self-hybridisation in the helix II region shown by crinkled lines in both conformations. The oligonucleotide binding sites are indicated by bold lines in both conformations.

A hammerhead ribozyme requiring the presence of two specific oligonucleotides for it to become active is shown in Fig. 2.8. While the deoxyribozyme gate in Fig. 2.7 is inactivated by blocking the substrate binding site, the ribozyme in Fig. 2.8 is controlled by a different mechanism. In the absence of effector oligonucleotides the molecule will self-hybridise to form a structure that is not a ribozyme. Hybridisation with the effector molecules overcomes the self-hybridisation of the inactive conformation and the molecule changes into a structure with a hammerhead ribozyme component. A comparison of the the multi-branch loop on the far right of Fig. 2.8 with the structural requirements of a hammerhead ribozyme depicted in Fig. 2.4 reveals how the straightening of the oligonucleotide binding sites upon hybridisation with two DNA effector molecules induces catalytic activity.

The conformational dynamics of RNA molecules allows for a relatively straightforward design of allosteric control structures into known ribozymes along the line of the concept represented in Fig. 1.2. Accordingly hammerhead ribozymes have been engineered with a wide variety of effector molecules (Soukup and Breaker, 1999). One strategy is to add effector binding sites at the crucially important helix II of the hammerhead structure (cf. Fig. 2.4). Due to the conformational flexibility of RNA it is then likely that an effector molecule binding to the ribozyme will affect the helix II conformation and thus disrupt the catalytic function.

For the application of ribozymes as signal processing components RNA structures that can be controlled with nucleic acid oligonucleotides as effector molecules are of particular interest. This is the case because the controlling oligonucleotide may conceivably be the product of a reaction catalysed by another ribozyme and therefore enable the implementation of small molecular control networks. Different approaches to controlling a hammerhead ribozyme by means of oligonucleotide effectors are illustrated in Fig. 2.9. All four strategies have been demonstrated in experiments (Porta and Lizardi, 1995; Burke et al., 2002; Komatsu et al., 2000; Wang et al., 2002). The first three (A–C) follow a common design philosophy. Starting from the basic hammerhead ribozyme structure shown in Fig. 2.4 an RNA sequence is engineered that does not fulfil the requirements for a hammerhead ribozyme, but can overcome this deficiency by hybridising with an effector oligonucleotide. This is explained further below.

The earliest implementation of an engineered allosteric control mechanism in a ribozyme (Porta and Lizardi, 1995) is based on an RNA molecule that can form a hammerhead ribozyme, but has a preferred secondary structure that does not resemble the hammerhead motif and shows no catalytic activity (right side in Fig. 2.9A). The self-hybridisation that stabilises the preferred conformation (left side in Fig. 2.9A.) can be overcome by a suitable effector molecule, the binding of which is energetically more favourable than the self-hybridisation. Upon binding the effector molecule the RNA sequence folds into an active hammerhead conformation. This control strategy is the one that has been used in the molecular switch shown in Fig. 2.8 (Penchovsky and Breaker, 2005) and is the one most commonly implemented.

The hammerhead motif of the ribozyme in Figure 2.9B is inactivated by self-hybridisation between the 3′-end of the ribozyme and its conserved junction region (Burke et al., 2002). Between the region of the ribozyme participating in helix III and the region near the 3′-end that is complementary to part of the conserved core is an effector binding site. The binding of an oligonucleotide effector to the binding region is energetically favoured over the self-hybridisation in the core region. Accordingly the binding of the effector releases the hybridisation of the core and activates the hammerhead structure. As mentioned earlier, the helix II is a necessary part of the hammerhead motif and its stability is important for the enzymatic activity of hammerhead ribozymes (Birikh et al., 1997). Fig. 2.9C shows a control strategy based on an RNA sequence that contains the essential components of a hammerhead motif short of the complementary regions that could form the helix II. Binding of the effector induce a pseudo-half-knot structure that together with the helix formed between the effector strand and the ribozyme apparently forms a

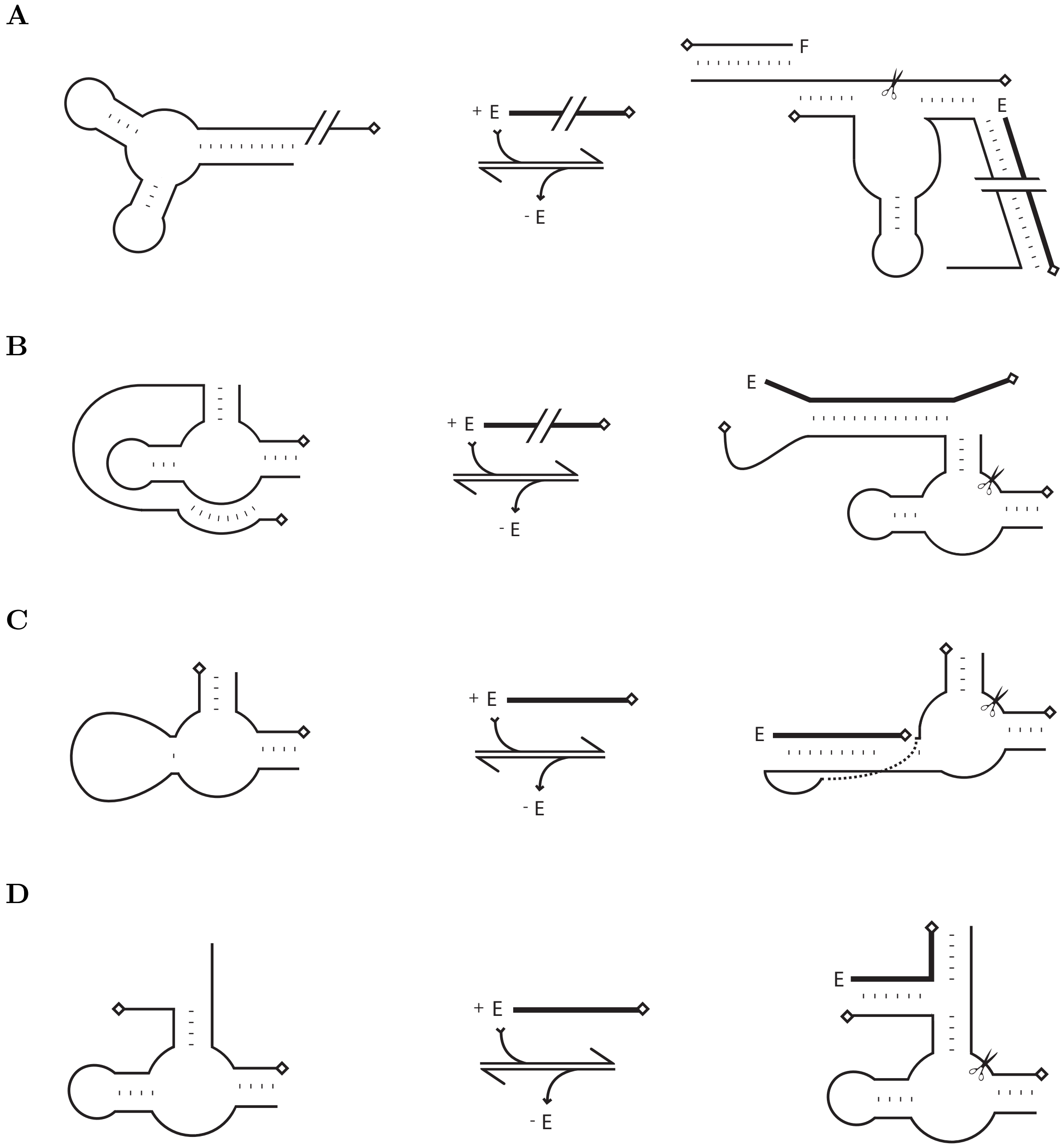


Figure 2.9: Four different strategies to control a hammerhead ribozyme. In all cases the ribozyme is active only in the presence of an oligonucleotide effector (E). Panel A: Formation of hammerhead structure upon binding of effector (Porta and Lizardi, 1995); a DNA facilitator strand (F) enhances the binding of substrate to ribozyme (Goodchild, 1992). Panel B: Effector releases conserved core junction from hybridisation (Burke et al., 2002). Panel C: Effector enables formation of helix II (Komatsu et al., 2000). Panel D: Effector supports binding of substrate (Wang et al., 2002).

pseudo-stem capable of activating the ribozyme (Komatsu et al., 2000). In contrast to the three allosteric ribozymes just described, the ribozyme shown in Fig. 2.9D is always in a catalytically active state and can cleave a sequence that will bind fully to form helix I and helix III (cf. Fig. 2.4 for helix positions). However, the catalytic activity with regard to substrate sequences that bind only partially in the helix III region can be controlled by an effector molecule (Wang et al., 2002). The effector binds to the dangling 3′-end of the ribozyme and the dangling 5′-end of a suitable substrate. It thus facilitates the binding between the ribozyme and a substrate that would not be cleaved without the effector. Note, that the effector sequence in this case will influence which substrates the ribozyme acts upon.

The combination of molecular motifs found in nature, molecules developed through directed evolution, and rational design decisions have led to a set of allosterically controlled functional nucleic acids suitable as components of simple molecular information processors. From the diverse family of catalytically active nucleic acids that have been found (Famulok and Szostak, 1992; Ellington and Szostak, 1990; Tang and Breaker, 2000) it appears likely that the set of available components will grow. These facts coupled with the different control mechanisms presented here allow for the design of a variety of computational units. Once a structural design scheme has been chosen, we would require a set of tools that enable us to translate this design scheme into a set of nucleic acid sequences. In the next section, we review computational tools that can support this step.

2.4 The Computational Biology of Ribonucleic Acids

Deoxyribonucleic acids (DNA) is the carrier of genetic information and is most commonly found to be in a three-dimensional double helical conformation (duplex). In the flow of genetic information, Ribonucleic acid (RNA) acts as intermediary, providing copies of DNA sequence information in form of messenger RNAs (mRNA) and plays a dual role in the translation of mRNA into protein (tRNA, ribosomes) (Alberts et al., 2002). Ribonucleic acids also play diverse regulatory roles in cell (Couzin, 2002) as well as having the ability to act as catalytic agent (Cech, 1987). RNA molecules carry out their functions as single strands, where self-complementary nucleic acid sequence regions dictate the folding into the secondary structure conformation and have therefore much greater structural heterogeneity. Single stranded DNA molecules can also fold back upon themselves, but with less structural diversity due to steric constraints (SantaLucia and Hicks, 2004).

Compared to the folding of protein, the amount of energy released in the formation of RNA secondary structure is much larger than the energy required during the formation of its tertiary structure. In fact, on its own, RNA secondary structure is quite informative. Long helices that are present in the secondary phase are most likely to be retained in its tertiary structure. This therefore provide a reasonable prediction of the basic form and relative positioning for some elements in the tertiary structure (Higgs, 1995). Tools such as *mfold* and *DINAMelt* (Zuker et al., 1991; Zuker, 2003; Markham and Zuker, 2005), the *Vienna RNA Package* (Hofacker et al., 1994), *RNAstructure* (Mathews et al., 2004), *RNAsoft* (Andronescu et al., 2003) and *RNAshapes* (Steffen et al., 2006) are some of the most common and well-known in the prediction of RNA secondary structure.

The folding of RNA into its tertiary structure, can be described hierarchically, from sequence configuration, the self-complementary folding through many weak hydrogen bonds based following canonical base-pairing rules (C-G, A-U and G-U) forming its secondary structure. The various secondary structure elements (loops) then undergo conformational changes that lead to more complex tertiary loops (Tinoco and Bustamante, 1999). Tertiary interactions are assumed to only change the weakest secondary structure elements such as shifting of base pairs in a relatively unstable helix and hybridisation between secondary structure loop elements (Higgs, 2000). The dominance of the secondary structure is largely based on the formation of helices (base-pairs) following the canonical complementary rules. Initially, this led to the development of optimisation algorithms that simply search for the RNA secondary structure with the highest number of base-pairs. Subsequently the algorithm for predicting secondary structures have been extended to include thermodynamic and kinetic information. Although the discussion of this section revolved mostly around ribonucleic acids (RNA), the substitution of thermodynamic parameters will make the algorithm suitable for deoxyribonucleic acids (DNA) (Hofacker et al., 1994; SantaLucia and Hicks, 2004).

Secondary structure prediction of single RNA molecule is a classical problem in computational biology. This area of research has raised many interesting questions and propelled the emergence of various computational tools (Reeder et al., 2006). Among these questions are the inverse prediction problem: given an RNA structure, find a sequences that will fold into the desired conformation. For RNA molecules, the number of possible sequence always exceeds the number of structures. Thus, for a particular conformation of an RNA molecule there are always many sequences that confer to it (Schuster et al., 1994; Schultes et al., 1998; Schuster, 2006). Random walks and stochastic approaches have been implemented to solve the inverse prediction problem. Most commonly, these approaches rely on folding prediction as a point of reference during the optimisation process. We discuss at length the algorithms available for the inverse prediction, as well as the construction of a variant algorithm that is more suited to the task of designing computational molecules in Chapter 4.

Secondary structure folding tools generally produce the equilibrium conformation of an RNA molecule, together with the minimum free energy (MFE). However, upon closer inspection, there can be a large number of suboptimal folding with free energy similar to the free energy of the native conformation (Zuker, 1989; Wuchty et al., 1999). This led to the development of secondary structure prediction tools that generate a set of suboptimal conformations which reside within a certain range from the minimum free energy of the equilibrium structure. The suboptimal structures are useful in determining how well-defined the lowest free energy structure is, and at the same time highlight weak base-pairing positions that are present in a structure.

More recently, secondary structure prediction has been extended to includes interacting molecules (Andronescu, 2003; Dimitrov and Zuker, 2004; Mückstein et al., 2006; Andronescu et al., 2005; Bernhart et al., 2006). The co-folding of two interacting RNA molecules is an important process for the function of regulatory RNAs (Nelson et al., 2003). In the context of constructing nucleic acids for information processing, (e.g., Fig. 2.7 and 2.8), the hybridisation between an effector molecule and the receptor site of an allosterically controlled nucleic acid is an example of intermolecular binding between RNA and RNA or DNA and RNA. The secondary structure prediction for interacting RNA molecules also quantifies the probability of homo-dimer formation (Dimitrov and Zuker, 2004). The relative probability of inter-molecular and intra-molecular binding is critical for designing a set of interacting nucleic acids for the task of information processing. To derive this network of nucleic acid units, it is important that the base pairing within each molecule binds stronger than any possible intermolecular binding involving other molecules in the solution. At the current state, such tasks would be infeasible if we apply the co-folding or multi-folding prediction tools directly. However, using the measurement of RNA-RNA and DNA-DNA binding along with the calculation of homo-dimers formation and probability biased base assignment, we have developed a sequence designer that is able to fulfil the task of designing such set of interacting nucleic acid. In Chapter 5, we discuss the development of the designer.

The properties of RNA molecules can be calculated using what we called utility programs. These programs usually requires less processing power then the secondary structure or inverse prediction programs. The characteristics of RNA that are taken into account include the free energy estimates of RNA in a given conformation, the distance value between two molecules, melting temperature of the molecules and the kinetics of folding of the molecule into a conformation. These utility tools play an important role in the construction of computational nucleic acids (Penchovsky and Breaker, 2005), acting as filters to prune out sequences under consideration that are regarded as unworkable in the laboratory.

Despite the various tools that are available, there are issues for which no computational prediction method exist. Some are directly associated with our aim of developing a general methodology for the construction of nucleic acid networks for information processing. One that stands out is the co-folding prediction of RNA-DNA duplexes. As with the basic structure prediction, the availability of thermodynamics information is crucial and thus, the absence of thermodynamics measurements for RNA-DNA intermolecular binding affect any attempt to develop a program for predicting RNA-DNA duplexes. Predicting RNA-DNA interaction is important in constructing computational nucleic acids as exemplified in the gate depicted in Fig. 2.8 where DNA effectors are required to form base-pairs with the binding site region that is a part of an allosteric RNA enzyme in order to inflict conformational change. The presence of this RNA-DNA co-folding molecules can greatly help the design process, but without any advancement in finding the thermodynamic contribution between RNA and DNA pairing, all we can use are RNA-RNA and DNA-DNA co-folding simulators.

For some RNA molecules, there can exists a number of alternative confirmations. Each conformation may be associated with a different function for these RNA molecules. This characteristic is exemplified by SV11 (a relatively small molecule that is replicated by Qβ replicase (Biebricher and Luce, 1992)) that exists in two different conformations. In its meta-stable state, SV11 served as a template for the replication process, but in its native state SV11 functionality as replication template vanishes. In nature, the capability of RNA molecules to form multiple (meta) stable conformations with different functions can be observed in RNA switches or Riboswitches (Nudler and Mironov, 2004; Winkler and Breaker, 2005; Nudler, 2006), which are responsible for regulating a number of biological processes. The issue of designing RNA sequences that can fold into prescribed alternative conformations has been addressed theoretically by Flamm et al. (2001). The schematic of computational nucleic acids illustrated in Fig. 2.7 and 2.8 demonstrates the needs of design tools for multi-stable sequences as both examples show, at least two different conformations, i.e., with and without effector molecules. The design of RNA sequences that can fold into two or more meta-stable states can speed up the candidate generation phase for the construction of nucleic acid capable of performing information processing task. In Chapter 4 we survey the current state of progress for sequence designers and evaluate the performance of each algorithm against the design space relevant for nucleic acids computing. Later in Chapter 5 we develop an algorithm specifically for multi-stable conformation sequence design and extend this algorithm to include interactions among molecules with multi-stable conformation.

2.4.1 RNA secondary structure prediction

With the assumption that the native conformation of RNA molecules in equilibrium is the one with the lowest free energy (MFE), an initial aim for the secondary structure prediction tools is to determine a conformation with a base pairing combination that yields to the minimum free energy. Based on the canonical base pairing rules, hydrogen bonds are formed between C-G, A-U pairs and also between the less stable G-U pairs. The occurrence of an isolated base pair is rare due to its instability, hence helices normally comprise of two or more adjacent base pairs. The stability of helices are enhanced by the attractive stacking interactions between these successive base pairs. The free energy of the helix is described by the nearest neighbour model (i.e., free energy is assigned to each two consecutive base pairs) (Zuker et al., 1999).

The RNA secondary structure can be predicted using thermodynamic, kinetic, or comparison methods. It has been established that for many known RNA sequences with a length of less than 500 nucleotides, 73% of the secondary structure can be predicted accurately (Mathews et al., 1999b, 2006). The structural prediction accuracy is given as sensitivity and is calculated as follows,

$$\text{Sensitivity} = \frac{\text{number of correctly predicted base pairs}}{\text{number of natural occurring base pairs}} \times 100$$

This suggest that the functional folds for RNA molecules can be largely determined by thermodynamics information. A survey conducted by Higgs (1995) confirms that for the complete tRNA database, 85% of the cloverleaf conformations were correctly predicted. However a survey conducted for longer sequences by Zuker and Jacobson (1995) only produced a mean of 49% for a sample of 15 SSU rRNAs. Konings and Gutell (1995) studied a larger sample of SSU rRNAs and only found between 10% - 80% with a mean of 46% correctly predicted base pairs. Morgan and Higgs (1996) studied a selection of long RNAs, including SSU and LSU rRNAs and RNase P, and achieved a mean accuracy of 55%. Refinement of the thermodynamic parameters has been explored in order to improve the accuracy. Recent attempt in parameter tuning by Andronescu et al. (2007) managed to enhance the accuracy of the prediction by 2% to 16% for tRNA, RNase P RNA and ribosomal RNA.

One major drawback of the thermodynamics based method is that it may get stuck in local energy minima. With the assumption that the minimum free energy structure would represents the native conformation of a RNA molecule, whenever a candidate structure is stuck in a local energy minima, the stacking energies of RNA helices can be too large to overcome compared to the thermal energy (kT), thus making it difficult for the conformation to dissociate and shift into a conformation that likely to arrive to the native state. Therefore, the kinetic aspect of determining RNA secondary structure focuses on the structure will form easiest instead of the one having the minimum free energy. The prediction accuracy of the kinetic methods has been surveyed by Morgan and Higgs (1996). They concluded that, similar to the thermodynamic method, the accuracy of the prediction decreases as the length of the sequences increased. The survey reported that more than 90% correctly predicted base pairs were found for structures shorter than 50 nucleotides. For structures with a length of 100 nucleotides, the accuracy decreases to around 80%, and as the structure length increases to 200 nucleotides, the accuracy decreases to 55%. Coincidently, this later value resembles the predicted accuracy produced by the thermodynamic method discussed earlier. Although the kinetic approach produced results similar to the thermodynamic approach, the kinetic simulation generates folding pathways that can indicate the plausibility of a structure forming (Higgs, 2000).

The comparison method for predicting secondary structure relies on finding a structure common to multiple homologous sequences. With a sufficiently large set of aligned sequences, the method searches for positions that have compensatory mutations (Higgs, 2000), i.e, changes at one position are correlated with changes at another position. For instance, if we have several sequences with G and C bases at position 5 and 9, and we have bases A and U at the same position in other sequences, then there is a strong evidence that a base pair is present at position 5 and 9. The assumption is that molecules with the same function in different species share a common structure and therefore it is possible to find base pairing patterns that are present if one has a reasonably diversified set of sequences. The comparison method is intended to overcome the two major flaws inherent in the thermodynamic approach, which are reliance on thermodynamic parameters estimation and the assumption that the native conformation is at equilibrium. In order for the comparison method to work there must be a reasonable amount of variation between the sequences so that the base pair pattern can be identified. However, if the variation is too much, then generating a reliable sequence alignment is impossible (Gutell et al., 2002). It has been reported that secondary structure prediction with the comparison method yields more reliable structures compared to both thermodynamic and kinetic methods (Higgs, 2000), but aside from acquiring multiple set of well-aligned sequences, there is no information available regarding alternative conformations, its folding pathways or even its thermodynamic stability: factors that are crucial for the construction of computational nucleic acids. The algorithm also requires significant manual input and intuition which increases the complexity of the design methodology.

2.4.1.1 The Terminology of Ribonucleic Acids *in silico*

A secondary structure can be depicted as a list of base pairs present in a particular nucleic acids sequence. There are two important notations for the formal definition of secondary structure; the nucleotide alphabet (bases) denoted as $\mathcal{A}$, where $\mathcal{A} = \{\alpha_1, \ldots, \alpha_4\} = $ {A,U,C,G} in natural RNA molecules, and the set of canonical base-pairs denoted by $\mathcal{B}$, where $\mathcal{B} = \{\beta_1, \ldots, \beta_6\}$ with $\beta_n = \alpha_i\alpha_j$. The set of permissible base pair combinations for secondary structure prediction is given as $\mathcal{B} =${GC,CG,AU,UA,GU,UG}. Non-canonical base pairs (Bloomfield et al., 2000) (i.e., base pairs that are neither Watson-Crick nor Wobble) such as A·G, C·U, G·G, U·U, A·A, C·C are found in ribosomal RNA (rRNA), tRNA and Polynucleotides (among others) and are not considered in the secondary structure prediction.

An RNA sequence is defined as a string of nucleotides ($\mathcal{X}$), where $\mathcal{X} = \{x_1x_2 \ldots x_n\}; x \in \mathcal{A}$ and each bases are numbered from 1 to n. RNA secondary structure as a set S of base pairs (i, j) must satisfy the following constraints:

(i) i and j are complementary, $(i, j) \in \mathcal{B}$ (belonging to either Watson-crick or wobble pairings),

(ii) $i < j$ and $|i - j| \geq 4$, since there must usually be at least three unpaired bases to form hairpin loop (Zuker et al., 1999), Let the bases in position k and l form another allowed base pair, we can then assume that (k, l) is compatible with the pairing of (i, j) if $i \leq k$ and:

1. $i = k$ if and only if $j = l$, and
2. $k < j$ implies $i < k < l < j$

The first condition simply states that each nucleotide can either form a base pair or exist as unpaired base. The enforcement of the second condition omits pseudoknots structures (cf. Fig. 2.10). There is an on-going debate in the field of computational biology regarding the omission of the pseudoknot as the majority of the community regards it as one of the tertiary structure motifs rather than secondary structure. The presence of pseudoknots plays a major role in RNA self-splicing, translational autoregulation, and ribosomal frameshifting (t. Dam et al., 1992). However, little is known about the sequence dependence and thermodynamic stability of pseudoknots structures. Accordingly, it is excluded by the majority of the secondary structure prediction tools. There are a few algorithms that include pseudoknots in secondary structure prediction (Rivas and Eddy, 1999; Dirks and Pierce, 2003) using thermodynamic approximation and kinetic method, albeit at the cost of increased algorithm complexity. For our purpose we have excluded pseudoknots due to the lack of thermodynamic data for this motif. The formation of a pseudoknot influences the stability of RNA molecules (Puglisi et al., 1991) and with only energy estimation to aid in the secondary structure prediction, the accuracy of our computational tool would be compromised. In shorter RNA sequences, it is rare to find the pseudoknot motif. With the design space in this study restricted to 200 nt and the overhead (in terms of processing time) of predicting secondary structure with pseudoknots (Dirks and Pierce, 2003), we decided to omit the pseudoknot motif in our design of nucleic acid units.

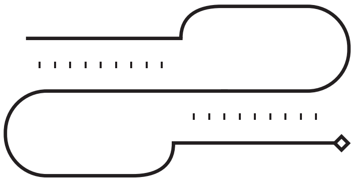


FIGURE 2.10: The secondary structure representation of a simple RNA pseudoknot motif. As illustrated, the formation of a pseudoknot occurs when a single strand on either side of a hairpin folds back and forms base pairs with the loop creating a two loops and two helices.

For a typical RNA molecule (cf. Fig. 2.11), approximately half of its conformation consists of multiple stretches of unpaired bases that can be identified as secondary structure loop motifs (see Fig. 2.2). In addition to hairpin loop, internal loop, bulges and multibranch loop, helices are also classified as secondary structure loop motifs. Helices are also known as stacked-loop. A stacked-loop is defined as a loop with size zero.

The definitions for the remaining motifs are as follows; the roman numbers correspond to the labels in Fig. 2.11:

I. A Multi-branched loop is defined as a loop with three or more closing base pairs creating a junction. The size of the multi-loop motif is determined by the number of branches it possesses.

II. A Hairpin loop is formed when a single-stranded RNA makes a sharp U-turn and folds back to itself. Normally a hairpin loops will have one closing base pair in either Watson-Crick complementary or wobble pairs and a set of unpaired bases. The size of the loop will be measured by the number of unpaired bases.

III. An Internal loop is created when helices or stacked loops are disrupted by the presence of unpaired bases. This leads to a loop that has two closing base pairs and an unpaired region. Internal loop can either be asymmetric or symmetric, where an asymmetric loop can be identified by having an odd number of unpaired bases.

IV. A bulge loop is derived from an internal loop which contains no unpaired base on one side of the strands.

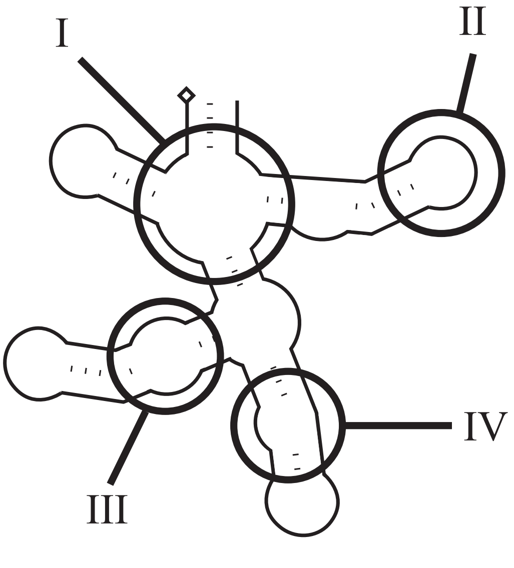


Figure 2.11: A typical RNA secondary structure uniquely decomposed into loops. The motifs shown are: Multibranch loop (I), Hairpin loop (II), Internal loop (III) and Bulges (IV).

Absent from Fig. 2.11 is the dangling end motif that is located at the end of the strand, either before the start of the first helix or right after the last helix. Such a dangling region is often used as sticky-end intended for inter-molecular self-assembly (Winfree, 2000; Mao et al., 2000).

RNA secondary structures can be depicted using a variety of representations, as shown in Fig. 2.12. The simplest way of representing a secondary structure is through the dot-bracket or parenthesis notation (Hofacker et al., 1994), where a pair of brackets "(" and ")" is used to annotate two paired bases and a dot "." to annotate an unpaired base (Fig. 2.12 A). Using the standard dot-bracket notation, one can easily determine all the base positions that will form base pairs as the position of brackets in the notation directly correspond to its RNA sequences. The majority of the computational tools adopted the dot-bracket notation to represent RNA molecules for both input and output channels. No matter how complex the conformation, any pseudoknots-free RNA molecules can be expressed. However as the length of the RNA molecule increases, finding matching brackets inside the notation becomes cumbersome. In the next chapter, we introduce an extended dot-bracket notation capable of expressing multi-conformational molecules as well as tackling the issue of finding matching brackets for long RNA molecules. To understand the structure of a conformation, a pictorial representation is required. Representing RNA molecule as a planar graph, constructed from the various secondary structure loop motifs often a more readable representation (Fig. 2.12 B). As depicted in Fig. 2.11, this more conventional representation shows a two-dimensional view of the structure in which the loop motifs, helices and dangling ends of the molecule can be descend easily. Secondary structure only tell us about the possible base pairing of a RNA sequence and the two-dimensional representation should not be construed as providing the relative positioning of the elements in three dimensions. In reality, the orientations, coiling and rotations of nucleic acids molecules present a more complex three dimensional views; an example is shown in Fig. 2.13.

A (((((((.....(((((......)))))....((((....))))............)))))))

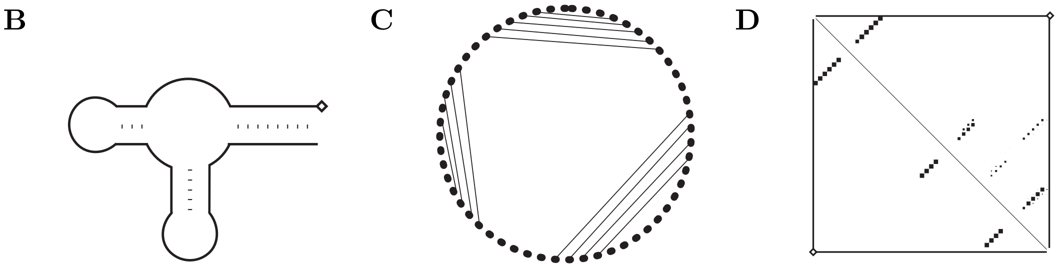


Figure 2.12: Representations of RNA secondary structure. The standard dot-bracket or string notation (A) The planar graph representation constructed from combination of different loop motifs (B) The outer-planar representation and (C) The dot-plot diagram (D) introduced by (Hofacker et al., 1994).

Alternatively, secondary structure can also be visualised as an outer-planar graph, where the nodes of the graph are nucleotides of the RNA molecule, $i \in \{1, 2, \ldots, n\}$ consecutively numbered in a circle and edges that connect the two nodes representing base pairs

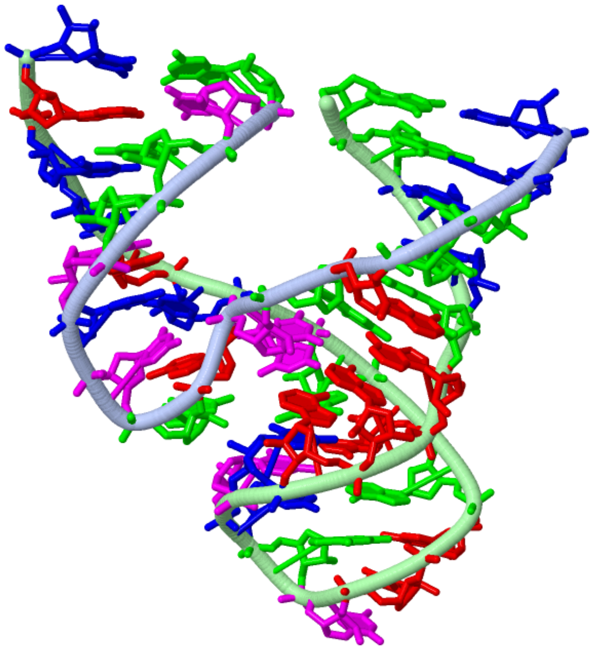


FIGURE 2.13: Three-dimensional representation of a hammerhead ribozyme The still image of the unmodified hammerhead RNA crystal structure (Scott et al., 1996) is rendered by *Jmol* (Sühnel, 2008) from the Jena Library of Biological Macromolecules (Reichert et al., 2000; Reichert and Sühnel, 2002).

(Fig. 2.12 C). An outer-planar representation can easily exhibits pseudoknot structure, which results in intersections of edges in the graph. It is also useful for visualising multiple conformations of RNA molecules, to which we will return in Chapter 5. Another common representation method for visualising RNA molecules is the dot-plot matrix. The dot-plot matrix shows the equilibrium base pairing probabilities as calculated using a partition function (McCaskill, 1990). McCaskill (1990) introduced a similar representation to the dot-plot diagram called “box matrix”. The equilibrium frequency of a base pair (i, j) is represented by a square of area in row i (labelled at the top and bottom end of the graph) and column j (labelled on both sides of the graph). The upper right triangle shows the base pairing probabilities, while the lower left triangle shows base pairs for the minimum free energy structure.

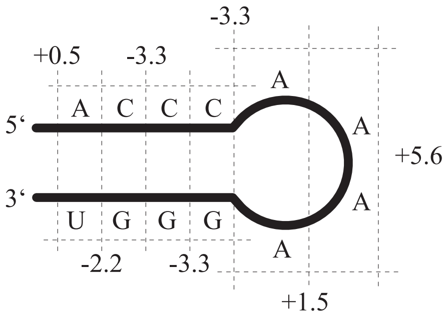
2.4.1.2 Thermodynamics of Ribonucleic acids

As discussed earlier, various types of unpaired regions occurs in between helices. These regions are known as hairpin loops (which connect two sides of a single helix), bulges and internal or interior loops (connecting two helices) and multi-branched loops or junctions (connecting three or more helices) (cf. Fig. 2.11). The thermodynamic parameters for the formation of the loop elements and helices have been determined by monitoring the unfolding of oligonucleotide model systems (Xia et al., 1998; Mathews et al., 1999b, 2004). Several techniques of measuring thermodynamic parameters have been reviewed in (Jaeger et al., 1989). The thermodynamic parameters for these loop motifs also account for free energy penalties due to unfavourable entropy associated with constraining conformational freedom whenever a base pair is formed at the loop ends (Mathews et al., 2006).

A free energy is assigned to each of the helices, loop motifs and dangling ends. By adding these free energy terms in a nearest-neighbour model (NNM), one arrives at a reasonable approximation of the complete free energy value for a particular RNA conformation. The energy of the structure can be written as:

$$E(S) = \sum_{\text{loops } L \text{ in } S} \epsilon(L) + \epsilon(L_{ext}), \tag{2.1}$$

where L_{ext} refer to the energy contribution of dangling ends. A helix here is treated as a loop element (called stacked loop) with zero length. Figure 2.14 shows an example of a nearest-neighbour model free energy calculation for a stem-loop structure.



$$\begin{aligned}\Delta G^{\circ 37} &= 0.5\ \frac{\text{kcal}}{\text{mol}} + \text{-}2.2\ \frac{\text{kcal}}{\text{mol}} + \text{-}3.3\ \frac{\text{kcal}}{\text{mol}} + \text{-}3.3\ \frac{\text{kcal}}{\text{mol}} \\ & \quad \text{-}3.3\ \frac{\text{kcal}}{\text{mol}} + \text{-}1.5\ \frac{\text{kcal}}{\text{mol}} + 5.6\ \frac{\text{kcal}}{\text{mol}} \\ &= \text{-}4.5\ \frac{\text{kcal}}{\text{mol}}\end{aligned}$$

FIGURE 2.14: An example of nearest-neighbour free energy calculation for an RNA stem-loop structure. The complete free energy estimate of the molecule is calculated by adding all free energy terms assigned to each secondary structure motifs based on the nearest-neighbour model (Zuker and Stiegler, 1981). Parameters for the calculation are measured experimentally (Jaeger et al., 1989; Lyngsø et al., 1999b; Mathews et al., 1999b; SantaLucia and Hicks, 2004) under the physiological environment of 37°C and 1 M NaCl concentration.

Generally, thermodynamic parameters for the loop motifs are known with lower accuracy compared to parameters for helices (SantaLucia and Turner, 1997). In addition, there are certain aspects such as the lack of multi-branch thermodynamic data and the imprecise approximation of dangling ends that can influence the free energy estimates of a structure. There are free energies available for helices subject to the base pairing of the next adjacent helix (i.e., a pair of G-C that is stacked to another G-C pairing is estimated to have a lower free energy value compared to a G-C pair next to an A-U pair). The free energy of the loop motifs, is calculated based on the number of unpaired bases of the loop, with adjustments for special circumstances specific to each loop motif. For instance, an asymmetry penalty is imposed on interior loops, or a free energy bonus is available for tetraloops (hairpin loop of size four). Zuker et al. (1999) provides a comprehensive guide for the nearest-neighbour model (NNM).

2.4.1.3 Dynamic Programming Algorithm

The thermodynamics model of secondary structure prediction suggests that the most stable RNA structure, is equivalent to the one with the minimum free energy (cf. Section. 2.4). The MFE structure can be obtained using the nearest-neighbour free energy calculation model presented in Section 2.4.1.2. In principle, one could construct a simple algorithm that explicitly generates all possible secondary structures that can be formed using a simplified free energy estimate assigned to each base-pair (set of $\mathcal{B}$). However, as the length of the molecule increases, the number of possible structures increases exponentially (1.8^N, where N is the number of nucleotides in the sequence and given an equal probability of assigning base A, C, G and U in the sequence) (Zuker and Sankoff, 1984), resulting in exponential complexity of the algorithm.

Using dynamic programming, the problem of predicting RNA secondary structure becomes tractable with algorithm that scale scaled in the time order of N^3 (Lyngsø et al., 1999a). This is possible as dynamic programming works in stages, thus allowing the calculation to be written as a recursive relation that breaks down the structure in smaller sub units. Waterman and Smith were first to realise that a dynamic programming algorithm can be used to predict RNA secondary structure, echoing their scheme to solve the problem of sequence alignment (Waterman, 1978; Waterman and Smith, 1978). The first dynamic programming solution using a simplified free energy estimation called “maximum-matching model” was suggested by (Nussinov and Jacobson, 1980). In this model, the free energy of each pair in a helix is assigned -1 unit with no penalties associated with loops. Hence, the predicted conformation with this model is the one with the maximum number of paired bases.

To illustrate the dynamic programming method that were implemented by Nussinov and Jacobson (1980), in the description here for simplicity, arbitrary integer values are used. We assign each base pair a different energetic value: -3 units for C-G/G-C, -2 units for

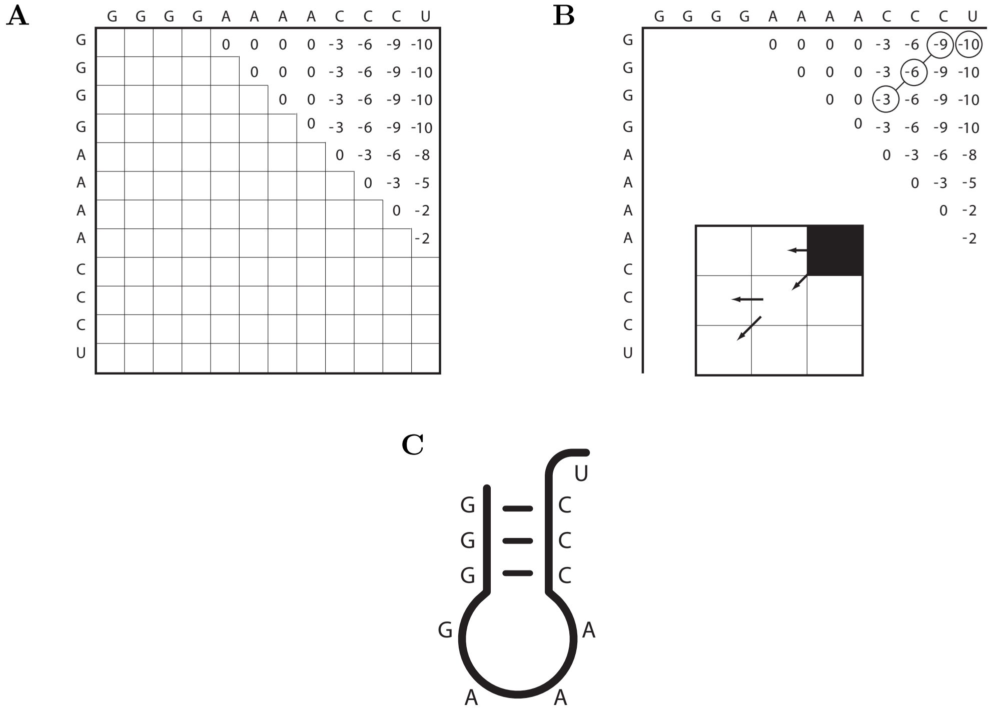
A-U/U-A and -1 unit for the wobble pair G-U/U-G. Similar to the maximum-matching model, no penalties are associated with unpaired bases. Let ϵ_{ij} be the free energy of bases i and j binding and $E(i, j)$ be the minimum free energy over all possible folding on the region, $R_{i,j} = r_i \dots r_j$, where i and j are paired, then the recursion to find the structure with the lowest free energy value can be written as (for $j - i \geq 4$):

$$E(i,j) = \min \begin{cases} E(i+1, j), \\ E(i, j-1), \\ \epsilon(i,j) + E(i+1, j-1), \\ \min_{i<k<l} \;\; E(i,k) + E(k+1, j) \end{cases} \tag{2.2}$$

For cases where $j - i < 4$, we assigned a free energy contribution of zero because folding is not possible for hairpin loops with less than 3 unbound bases due to steric constraints as discussed in Section 2.4.1.1. In general, dynamic programming algorithms are divided into two steps. In the first step, called "fill", the lowest conformational free energy is calculated recursively starting from the smaller fragments, moving on to larger and larger fragments, until the whole sequence is completed. At each stage, a pointer that refer to the lowest $E(i, j)$ is stored. The second step called "traceback" is then conducted by backtracking through this array of pointers to obtained the structure with the lowest $E(1, n)$ (as illustrated in Fig. 2.15).

In the "fill" step, for each fragment of $R_{i,j}$, there are only four possible ways in which a structure with a nested base pair can be constructed; r_i can only be either paired or unpaired. If r_i is unpaired, then the minimal free energy $E(i + 1, j)$ holds since no penalty is assigned for an unpaired base. The same holds if r_j is unpaired, and the free energy is given by $E(i, j - 1)$. If r_i is paired, then it is likely to be paired with r_j, in which case the free energy of $\epsilon(i, j)$ is added to the minimal free energy of remaining base pair $E(i + 1, j - 1)$. If r_i pairs with some other bases r_k, the minimal energy is contributed into two disjoint folding (a bifurcation) on $R_{i,k}$ and $R_{k+1,j}$, with energy of $E(i, k) + E(k + 1, j)$. As depicted in Fig. 2.15 A, we tabulate the minimum free energy $E(i, j)$ in a triangular matrix starting from $E(1, 1)$ until $E(1, n)$. Once the "fill" step is completed, the second stage of tracing the optimal path is conducted in order to obtain the structure corresponding to $E(1, N)$ (cf. Fig. 2.15B). Starting from the top right hand of the matrix ($E(1, n)$), we then move to the next optimal point in either horizontal or diagonal direction until no more traceback fragments $R_{i,j}$ remain.

Unlike the simplified energy model presented above, RNA secondary structure prediction tools such as *mfold* (Zuker et al., 1991; Zuker, 2003) and *RNAfold* from Vienna RNA package (Hofacker et al., 1994) used a more complicated energy model during the "fill" phase. The calculation of free energy contribution is based on the nearest-neighbour model has been discussed above in Section 2.4.1.2. Energy minimisation algorithms tends to be somewhat complicated as the recursion relations used in these programs



A

	G	G	G	G	A	A	A	A	C	C	C	U
G					0	0	0	0	-3	-6	-9	-10
G						0	0	0	-3	-6	-9	-10
G							0	0	-3	-6	-9	-10
G								0	-3	-6	-9	-10
A									0	-3	-6	-8
A										0	-3	-5
A											0	-2
A												-2
C												
C												
C												
U												

B

	G	G	G	G	A	A	A	A	C	C	C	U
G					0	0	0	0	-3	-6	-9	-10
G						0	0	0	-3	-6	-9	-10
G							0	0	-3	-6	-9	-10
G								0	-3	-6	-9	-10
A									0	-3	-6	-8
A										0	-3	-5
A											0	-2
A												-2
C												
C												
C												
U												

FIGURE 2.15: The "fill" and "trace" steps of dynamic programming. The "fill" step recursively calculates the free energy of folding based on Eq.2.2 (A). The "traceback" step uses the free energy precomputed during the "fill" step to obtain the optimal folding. The inserted grid with horizontal and diagonal arrows represents the backtracking directions undertaken by the "traceback" step (B). Using the "traceback" step described in (Eddy, 2004; Mathews et al., 2006), the backtracking algorithm picks either a direct or diagonal path based on the free energy of $E(i, j)$. If similar energy values are recorded between both direct and diagonal paths, a random selection is made. For a given sequence of `GGGGAAAACCCU`, three base pairs of (G-C)–(G-C)–(G-C) were identified by the algorithm, the optimum folding is predicted in (C).

have to calculate penalties and bonuses specific for each loop motif with many special cases to be considered. In an attempt to reduce computation time, free energy estimates for the loop motifs are typically restricted to a maximum size of 30 nucleotides (Zuker et al., 1999). Despite the more complex energy models, the underlying concept of the algorithms remains similar to the one described above. The most common secondary structure prediction algorithms scale $O(n^3)$ in time and $O(n^2)$ in storage (Mathews et al., 2004; Lyngsø et al., 1999a), where O refers to the order of the calculation and n is the length of the molecule.

The energy minimisation discussed earlier yields only the single structure with minimal free energy. There are two major drawbacks of having just a single estimated native state. An RNA molecule will not always stay in its native conformation, but bounces between many alternative conformations within similar free energy value. Another drawback is the inaccuracies of the thermodynamic energy model. Every time we have several alternative structures within a close proximity to the minimum free energy, then choosing a particular structure becomes arbitrary. To compensate for these issues, McCaskill

(1990) introduced a statistical thermodynamics approach for RNA secondary structure prediction. Following McCaskill (1990), the partition function, Z, is the sum of all the Boltzmann factors given as,

$$Z = e^{-\Delta G^{\circ}/RT} \tag{2.3}$$

where R is the gas constant (1.987 $\frac{cal}{Kmol}$) and T is the temperature in kelvin and ΔG° is the standard free energy difference between folded and unfolded states). The probability of any structure is given by its Boltzmann factor divided by Z. The partition function of all foldings on $R_{i,j}$ is denoted as $Z(i,j)$ and the restricted partition function of only those folded conformations that contain the base pair r_i and r_j is denoted as $Z'(i,j)$. For $j - i < 4$, we let $Z(i,j) = 1$ and $Z'(i,j) = 0$. The partition function $Z(i,j)$ for bases r_i to r_j in $R(i,j)$ can be obtained using the following,

$$Z(i,j) = Z(i+1,j) + Z'(i,j) + \sum_{i<k<j} Z'(i,k)Z(k+1,j) \tag{2.4}$$

$$Z'(i,j) = e^{-\epsilon(i,j)/RT} Z(i+1,j-1) \tag{2.5}$$

For any base pair, r_i and r_j, $1 \leq i < j \leq n$, the product of $Z'(i,j)Z'(j,i+n)$ is the partition function for all possible foldings that contain this base pair, assuming that the loop-dependent energy rules (Section 2.4.1.2) are used. From this, one can calculate the probability that bases i and j are paired in the complete equilibrium ensemble of structures:

$$\mathcal{P}(i,j) = \frac{Z'(i,j)Z'(j,i+n)}{Z(1,n)} \tag{2.6}$$

The pair binding probabilities $\mathcal{P}(i,j)$ provides important information for the consideration of equilibrium structure alternatives to the MFE structure. From here, one can assess the accuracy of the thermodynamic parameters as well as determine the flexibility of these structures towards kinetic changes, i.e, opening and shifting of base pairs. The dot-plot representation implemented in the *Vienna RNA package* (Hofacker et al., 1994) gives a graphical representation of these base pairing probabilities. The dot-plot notation also allows for identification of variable and non-variable regions, similar to the "box matrix" display discussed in (McCaskill, 1990). The partition function approach is available through the *RNAfold* program in the *Vienna RNA package* and *RNAstructure* (Mathews, 2004). From the partition function algorithm, it is also possible to calculate the heat capacity that can be used to predict the 'melting curve' of an individual sequence. The calculation of heat capacity can be obtained from the formula (McCaskill, 1990):

$$C_p = -T\frac{\partial^2 G}{\partial T^2} \qquad \text{and} \qquad \Delta G = -kT\,ln\,Z \tag{2.7}$$

Computationally, the 'melting curve' calculation is provided in the *Vienna RNA package* through the *RNAheat* program. 'Melting curve' calculation allows one to identify the estimated temperature at which the structure completely unfolds. This is useful in validating whether a structure can be preserved within a certain range of temperature. The standard model of the energy parameters were measured at body temperature (37°C), therefore in order to give allowance for parameters error, one can devise a validation test to check whether the conformation of an RNA structure can be preserved within a certain range. For instance, the molecular gates of (Penchovsky and Breaker, 2005) were developed to retain their conformations within 20–40°C.

A dynamic programming algorithm capable of predicting secondary structure with a large class of pseudoknots has been reported in (Rivas and Eddy, 1999). Only the most complex pseudoknot topologies are omitted, while most biological relevant pseudoknots can be found by this algorithm. The algorithm presented by Rivas and Eddy (1999) also allows coaxial stacking between helices at junction to be included in the free energy calculation. Although experimentally significant (Higgs, 2000), this additional energy contribution has been excluded from the conventional dynamic programming algorithms. With addition of pseudoknot prediction and the coaxial stacking energy, the algorithm now scales $O(n^6)$ in time and $O(n^4)$ in storage, and is currently impractical for sequences of more than 100 nucleotides. (Mathews, 2005). Various attempts (Dirks and Pierce, 2003; Condon et al., 2004) were devised to improve the scaling of the algorithm, but this was only achieved on the expense of reducing the complexity the of structures that can be predicted.

2.4.1.4 Suboptimal Folding Algorithm

As pointed out in Section 2.4.1.3, the main weakness of many dynamic programming algorithms using the thermodynamic approach is that by design, they yield only a single solution. There could be a large number of alternative folds that are within close proximity of the minimum free energy value, and are classified into a well-defined ensemble that is kinetically interchangeable. This leads to the understanding that although the MFE structure is shown to be formally correct, in practise the MFE structure may undergo a simple kinetic motion and change into one of the well-defined ensemble (Mathews, 2005). This phenomena is relevant especially in RNA switches (Riboswitches) where conformational switching between two highly stable states has been reported (Schultes and Bartel, 2000; Flamm et al., 2001; Avihoo and Barash, 2006).

Suboptimal folding algorithms capable of predicting alternative folding that within close proximity to the MFE structure are therefore desirable. Suboptimal folding algorithms can be used to determine the high probability base pairing regions in RNA secondary structure folding and can indicate a plausible folding trajectory, followed by an RNA molecule to arrive at its native conformation. Using circular RNA viroids as reference, Zuker (1989) derived a dynamic programming algorithm that doubled the "fill" step calculation to produce a matrix consisting of free energy of possible folds on an elongated RNA sequences (i.e., doubling the original sequence). In circular RNAs, the choice of an origin is arbitrary (i.e, there is no $5'$ or $3'$ end). For circular molecule of $r_1, r_2, \ldots, r_n$, a base pair of r_i and r_j divides the structure into two components, the "included fragments" from r_i to r_j and the "excluded fragments" from r_j back to r_i. The "fill" step (previously discussed in Section 2.4.1.3) is applied to the elongated sequence and doubles both the in term of CPU time and the storage capacity required. Free energy calculation for the "fill" step is now the sum of the energies of both fragments $E(i,j) + E(j,i)$. Accordingly, the "traceback" procedure then identifies all base pairs for which the $E(i,j) + E(j,i)$ is within a particular percentage value to the MFE value (Zuker, 1989).

Alternatively, the recursive nature of dynamic programming allows for a less resource heavy approach. For instance, by allowing the "traceback" procedure to begin on any fragments, we could easily arrive at alternative folding that are within close proximity to the MFE value. However, such a naive approach only produces a fraction of the possible suboptimal folding. For a sequence of length n, at most $n(n-1)/2$ suboptimal structures are produced (Wuchty et al., 1999). A more appropriate method to generate a complete set of suboptimal structures was developed by Wuchty et al. (1999) based on the structure counting procedure for the maximum-matching model proposed in (Waterman and Byers, 1985). The algorithm implements a suboptimal "traceback", where at each stage in the "traceback" procedure, any partial structures that are within a certain energy range from the minimum free energy value $E(1,n)$ are kept for further refinements. This criterion is presented as, $E_\varphi \leq E(1,n) + \delta$, where φ represents partial structures that are kept for further refinement and δ represent the level of sub-optimality. Iteratively this "traceback" procedure produces a "branching tree" with a depth of δ. The parameters are introduced to limits the growth of the "branching tree" in order to reduce the amount of CPU time and memory required for the procedure. If $\delta = 0$, then the "traceback" procedure is equivalent to the conventional "traceback" step presented in Section 2.4.1.3 with additional alternative folds, if there exist any. Setting $\delta = \infty$ yields a complete suboptimal "traceback" equivalent to simulating the complete kinetic folding of the molecule. The algorithm is implemented in the *RNAsubopt* program from the *Vienna RNA package* (Hofacker et al., 1994).

2.4.1.5 Kinetic Folding Algorithm

Instead of relying solely on thermodynamics, kinetic folding algorithm assume that the secondary structure of an RNA molecule is best associated with structures that form easily rather than with the structure having the lowest free energy. This theory is parallel with the finding that for larger RNAs, only half of the structures were correctly predicted using the thermodynamic approach (Morgan and Higgs, 1996). Rather than assuming that insufficient energy parameters caused these inaccuracies, Morgan and Higgs (1996) concluded that the energy model is essentially correct but the molecules are prevented from reaching their native conformation by kinetic traps. Several studies of RNA folding kinetics (Treiber et al., 1998; Treiber and Williamson, 1999; Pan et al., 1999, 2000) show that the energy landscapes for large RNAs is rugged, and consequently these molecules can easily be trapped in meta-stable states that requires a longer timescale to unfold and refold itself into their native conformation.

The "hierarchical folding" (Brion and Westhof, 1997; Tinoco and Bustamante, 1999) used to describe the different phases in arriving at the tertiary structure of RNA molecule. Morgan and Higgs (1996) expanded this view, by hierarchically describing the formation of RNA secondary structure elements. Short helices are suggested to form first, followed by the formation of longer helices, which in turn happens much more slowly depending on presence of short helices fragments. Most long helices only form when previous short helices are already present, bridging the distance between two long fragments. Rearrangements and merging reactions occur to promote the formation of these longer helices. However, each rearrangement or merger would require a leap of activation energy to cross the energy barriers that differentiate between these different meta-stable states (Morgan and Higgs, 1998). As the size of the individual secondary structure elements increases, the energy barrier also grow and therefore the time required for structural reorganisations to occur also increases. This might leads to a kinetic trap, i.e., the energy barriers become too large to overcome in a biologically reasonable time (Morgan and Higgs, 1996). Following Higgs (2000), from the perspective of kinetic folding, a sequence ℓ specifies a set of conformations $S(\ell)$,

$$S(\ell) = \{S_0, S_1, \ldots, S_m\} \cup \{\mathbf{0}\} \tag{2.8}$$

where S_0 represents the MFE structure, $S_1 \ldots S_m$ represent suboptimal conformations of sequence ℓ ordered according to its free energy estimates and $\mathbf{0}$ denotes the unfolded strand. The secondary structure formation can then be described as a successive selection of elementary step based on some probabilistic model from a set of allowable kinetic moves. This results in a folding trajectory $\mathcal{T}$ comprised of a time-ordered series of structures $S(\ell)$, starting from the open chain $\mathbf{0}$ and ending with the MFE structure S_0,

$$\mathcal{T}(\ell) = \mathbf{0}, \ldots, S(t-1), S(t), S(t+1), \ldots, S_0; S(j) \in S(\ell) \tag{2.9}$$

Based on the kinetic folding assumption of Morgan and Higgs (1996), eventually all sequences will fold into its MFE structure. However, some folding require a longer time to break the energy barriers when stuck in a local optima. Computational algorithms developed for kinetic folding, there focus on secondary structures that form within a certain time-scale.

An earlier implementation of a kinetic folding algorithm is presented in (Abrahams et al., 1990). The algorithm sequentially adds helices to the structure in such a way that the free energy is lowered at each step. The next helix to be added is the one that lowers the free energy of the structure by the largest amount. In addition this kinetic approach can generate pseudoknot structures rather easily. In (Abrahams et al., 1990), base pair cross-linking (i.e., if i and j paired, then if there exists base pairing in k and l, then it is assumed that the base position must follows $i < j < k < l$) is not taken into account. Thus, the algorithm allow for any base pair to occur (e.g., (i, k) and (j, l) even though $i < j < k < l$) as long as the helix that is to be added to the structure contributed to the lowest free energy estimates.

Other implementations of the kinetic approach use Monte Carlo simulation (Higgs and Morgan, 1995; Schmitz and Steger, 1996; Fernández et al., 1999). These algorithms introduced two kinetic move sets, the addition of helices and the unzipping of existing helices. A reaction rate of formation is assigned to each possible base pairing in the molecule and similarly, removal rates are assigned to existing base pairs. One move set is then chosen based on a probability proportional to its rate. In this sense, faster reactions are more likely to occur compared to slower reactions, although all possible moves with non-zero probability can be selected. At each stage, these changes are reflected directly in the conformation. As suggested by (Higgs and Morgan, 1995), the reaction rates for each kinetic move set follow the Metropolis algorithm (Metropolis et al., 1953); The rate assigned to the unzipping of a helix that increases the free energy by an amount of ΔG is $r_1^{(-\Delta G/kT)}$, where r_1 is the rate at which helices that lower the free energy one added. The rate constant r_1 is chosen so that the timescale of the simulation is relatively close to the timescale observed in the experimental measurement of RNA folding.

Flamm et al. (2000) extended the model of Higgs and Morgan (1995) by introducing additional kinetic move sets to speed up the reorganisation of secondary structures. In addition of the normal formation and unzipping of base pairs, Flamm et al. (2000) introduced the base-pair "shifting" move (i.e., the combination of a base pair unzipping and a base pair addition in which one position remains unchanged) in order to imitate the "defect diffusion" mechanism, believed to be important in the dynamics of RNA folding (Pörschke, 1974). These shift-moves can be imagined as a propagation of a bulge loop along a stem. The Kawasaki's dynamics (Kawasaki, 1966) was chosen to replace the Metropolis assumption for assigning reaction rates. The Kawasaki's dynamics method is chosen because it favours downhill steps with larger free energy gain and accordingly yields shorter folding times. Flamm et al. (2000) concluded, however, that neither of the two models generates any qualitative difference in simulating the folding trajectory.

A genetic algorithm implementation of kinetic folding has been proposed by several authors (van Batenburg and Pleij, 1995; Benedetti and Morosetti, 1995; Shapiro et al., 2001). Generally, during the start stage, a population comprising alternative conformations for a given sequence is kept. The fitness of each structure is measured as the free energy estimate of the structure. Following the trend of the Monte Carlo approach, structures with lower free energy have a higher fitness and are kept for reproduction. The kinetic move sets comprises of mutation (i.e., addition and removal of helices) and recombination (i.e., a hybrid structure formation from the cross-over process of two parent structures). Similar to the Monte Carlo approach, the series of structures $S(\ell)$ that appears during the simulation is equivalent to a folding pathways. Although there is no quantitative measure of time, one can use the number of generations the algorithm requires to converge as a qualitative representation. In addition, there is also no guarantee that the population will converge to the MFE structure despite the inclusion of thermodynamics in its selection.

2.4.2 Secondary Structure Prediction of Interacting RNA molecules

Self-assembly between two or more RNA molecules are common in biology. An example is presented in Fig. 2.3 involving the self-assembly between a substrate and a ribozyme core strands. From the self-assembly of the two strands, two additional helices are formed (helix I and helix III of a hammerhead ribozyme) and a conformation of hammerhead ribozyme is now appears. The catalytic reactions of the ribozyme then cleaved the substrate strand (which is now the complementary region of helix I and helix III of the ribozyme). Apparently, the single molecule folding prediction algorithms discussed previously, can be generalised in a straightforward manner to solve the co-folding prediction of two RNA molecules (Andronescu, 2003; Dimitrov and Zuker, 2004; Mückstein et al., 2006; Andronescu et al., 2005; Bernhart et al., 2006). In the simplest approach, only the inter-molecular binding (i.e, base pairing across two molecules) are taken into consideration, omitting the intra-molecular pairing (self-folding) of each molecule. This approach is implemented in the programs *RNAhybrid* (Rehmsmeier et al., 2004), *UNAfold* from the *mfold server* (Dimitrov and Zuker, 2004), and *RNAduplex* from the *Vienna RNA package* (Hofacker et al., 1994). An earlier attempt that considers alternative folding of inter-molecular binding was suggested by (Mathews et al., 1999a), and was made available in *OligoWalk*. However, a biophysical more plausible model is the co-folding of two RNA molecules.

A co-folding prediction can be made by concatenating two RNA strands into one sequence. The idea is to apply different energy parameters for the loop that contain the linkage location of the two sequences (Andronescu et al., 2005). The programs *pairfold* from *RNAsoft* (Andronescu et al., 2003) and *RNAcofold* from the *Vienna RNA package* (Bernhart et al., 2006) implement this approach. For two sequences of RNA ℓ_1 and ℓ_2

(i.e., $\ell_i = l_i^1 \, l_i^2 \, \ldots \, l_i^n$), let $s = l_1^1 \, l_1^2 \, \ldots \, l_1^n \parallel l_2^1 \, l_2^2 \, \ldots \, l_2^n$ denote the sequence obtained from concatenating both ℓ_1 and ℓ_2, and as $\parallel$ the linkage location of ℓ_1 and ℓ_2. Then any loop motifs comprising the linkage location is classified as a special element and accordingly a different set of energy parameter is applied for the free energy calculation. If the linkage location is not part of the loop motif, the normal energy parameters are used in the free energy calculation.

Co-folding allows structures to have both inter and intra-molecular pairing (i.e., formation of both internal as well as external base pairs). Using dynamic programming algorithm, in the "fill" step, the free energy of these "special" loop motifs are calculated similar to an internal loop, but with an additional penalty being apply (Mathews et al., 1999a). In the "traceback" step, the complex with the minimum free energy (MFE) is chosen. For co-folding, this complex could either be two molecules self-assembled into a supra-molecular complex or two unbound molecules that undergo self-folding (i.e., homo-dimers). By extending the sub-optimal folding model of Wuchty et al. (1999), the *pairfold* program also generates sub-optimal conformations, whose free energies are within a specified distance from the minimum free energy complex. However, in *pairfold* the number of conformations generated can be controlled explicitly by the user instead of being predefined as in (Wuchty et al., 1999).

Andronescu et al. (2005) introduced an extended method of predicting secondary structure for multiple molecules using the program *multifold*. To work out the multi-folding problem, for a set of n RNA sequences $(\ell_1, \ell_2, \ell_3, \ldots, \ell_n)$, one needs to find the MFE structure for the concatenated sequences corresponding to the possible permutation of the set of N RNA sequences. For each permutation σ of $1, 2, \ldots, m$, let $\mathcal{S}_\sigma^i$ represents the concatenation of sequences $\ell_1, \ell_2, \ldots, \ell_m$ according to σ, where m denote the number of molecules ($m > 2$), for instance for $m = 3$, we have the following:

$$\begin{array}{ll} \sigma_1(1,2,3) & s_\sigma^1 = \ell_1\ell_2\ell_3 \\ \sigma_2(1,3,2) & s_\sigma^2 = \ell_1\ell_3\ell_2 \\ \vdots & \qquad \vdots \\ \sigma_6(3,2,1) & s_\sigma^6 = \ell_3\ell_2\ell_1 \end{array} \tag{2.10}$$

The MFE structure $\mathcal{S}_\sigma$, is then chosen from the collection of structures $\{\mathcal{S}_\sigma^1, \mathcal{S}_\sigma^2, \ldots, \mathcal{S}_\sigma^m\}$ generated based on the permutation of σ. The number of permutation grows exponentially with the number of sequences (n), therefore this approach is only suitable for a small set of sequences.

Another point to consider is the accuracy of *pairfold* in predicting the self-assembly between ribozymes and the RNA substrate strands. Seventeen pseudoknot-free ribozymes and RNA substrate complexes were chosen in the study, with total length of between 43 to 170 nt. According to Andronescu et al. (2005) the overall accuracy reported for predicting the self-assembly of the two molecules is 79%, with 87% correctly predicted base pairs compared to the reference structure. The overall percentage improved to 91% with the aid of 100 suboptimal structures. On the other hand, for multiple molecule prediction, the program *multifold* produced at least 93% overall accuracy for predicting five types of ribozymes that involve the self-assembly of three or four molecules. Furthermore, a dataset of DNA molecular automata from Benenson et al. (2004) were evaluated using *multifold* with similar results regarding the folding of multiple RNAs.

A common theme in constructing molecular processors involves the hybridisation of a short oligonucleotide to a much larger RNA molecules. As mentioned earlier in this section, algorithms that specifically handle prediction of inter-molecular binding between a pair of molecules have been presented in *RNAhybrid* (Rehmsmeier et al., 2004), *UNAfold* from the *mfold server* (Dimitrov and Zuker, 2004) and *RNAduplex* from the *Vienna RNA package* (Hofacker et al., 1994). These algorithms however, assume that the inter-molecular binding site for the two molecules is restricted to the exterior loop region, which is not always the case. In fact, there is no biophysical plausible reason to exclude any unpaired region of the target molecules from functioning as oligonucleotide binding site (OBS). Mückstein et al. (2006) proposed a co-folding algorithm that allows oligonucleotide binding with any unpaired region of the loop motifs (hairpin, interior and multibranch loop). The algorithm is conducted in two stages, where during the first stage, the partition function for the secondary structure of the larger RNA molecules is calculated with the constraints that a certain region in the molecule remains unpaired. Then, in the second stage, the energetics of RNA-RNA interaction is computed as,

$$\Delta G = \Delta G_u + \Delta G_h \tag{2.11}$$

in which the free energy of binding consists of ΔG_u the energy contribution necessary to open the binding site in a particular conformation and ΔG_h represents the contribution of energy gain due to the hybridisation that occurs at the binding site. The total interaction probability at a possible binding site is then obtained as the sum over all possible type of bindings. Due to the lack of thermodynamic measurements, the energy contribution of the loop motifs that might be altered by the RNA-RNA hybridisation is kept constant. The algorithm is implemented as the program *RNAup* included in the *Vienna RNA package*.

In this chapter, we have discussed a number of computational tools that cater for a wide variety of problems ranging from the prediction of RNA secondary structure and the design of sequence given an RNA molecule conformation, to prediction of suboptimal folds, co-folding, multi-folding, prediction of folding pathways, and so forth. Most of the computational tools available are directed towards the mapping of an RNA sequence to its structure. There is a lack of tools for the mapping from an RNA structure to its sequence. In particular there is no tool available for the design of sequences for RNA molecules with multi-stable states including multiple interacting molecules. Before we discuss on the implementation of a computer-aided design protocol (to design nucleic acid units for the task of information processing), the problem involving the design of sequences for interacting multi-stable RNA molecules need to be addressed. In Chapter 4 we discuss and compare several sequence designer algorithms and later in Chapter 5, we developed a sequence designer algorithm that addresses the problem of designing sequences for interacting multi-stable RNA molecules.

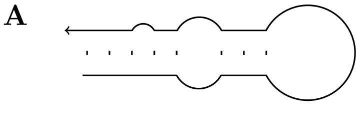
Chapter 3

An Extended Dot-Bracket-Notation for Functional Nucleic Acids

The material in this chapter is reprinted largely from the paper titled "An Extended Dot-Bracket-Notation for Function Nucleic Acids" was presented at International Workshop on Computing with Biomolecules (Ramlan and Zauner, 2008).

3.1 Representation of Nucleic Acid Structure

Within recent years nucleic acids of up to about 200 nucleotides in length have become a focus of interest for prototype implementations of molecular computing concepts. During the same period the importance of ribonucleic acids as components of the regulatory networks within living cells has increasingly been revealed. As we have discussed in Chapter 2, while the configuration of the nucleic acids is generally linear, they can adopt to a range of conformations. A widely used method to denote RNA secondary structure is the dot-bracket-notation or parenthesis format introduced by Hofacker et al. (1994). It uses matching parenthesis and dots to denote paired and free bases, respectively. Fig. 3.1 illustrates the notation for a short RNA sequence. The dot-bracket-notation



B `CCGAUAGAGGCGUGCGGUCAAGGUCCGG`
C `(((((...(((.....)))...)).)))`

FIGURE 3.1: A sample secondary structure of an RNA molecule is shown (A) together with the notation of its sequence (B) and its structure in dot-bracket form (C).

has the advantage that a string denoting the secondary structure of a nucleic acid is of the same length as the string denoting the nucleotide sequence with a single character for each nucleotide. The two strings can be aligned to show the secondary structure features along the nucleotide sequence (Fig. 3.1 B and C).

In molecular biology one typically has a given (discovered) sequence and is interested in its folding properties (cf. Section 2.4). The dot-bracket notation reflects this mode of operation. As indicated above, in molecular computing applications it is common that the secondary structure is of importance, but the detailed sequence that yields the structure is arbitrary over large stretches. In such a scenario the dot-bracket notation is often cumbersome, it leads to large expressions with information that is partially obscured for human readers, because it would require counting identical characters. While the sequence is typically arbitrary in most positions, as long as the structure of the molecule is preserved, nucleic acids with functional properties, such as catalytic activity, often require specific bases in a few positions. If in a few places the nucleotide sequence (i.e. the primary structure) is given, two strings are required: one to specify the structure in dot-bracket-notation and a second string to represent the type of the immutable nucleotides. Furthermore, for communicating structural features among humans a two-dimensional rendering of the one-dimensional dot-bracket notation is often desirable. It would be convenient if specific features in the sequence could be communicated to the rendering software.

We have extended the standard dot-bracket-notation for the convenience of human users as well as machine processing. The extensions allow for a more compact notation through the use of iterator operators and grouping symbols, provide for constraints placed on the nucleotides that may appear in a position, and facilitate the annotation of sequence regions and the graphical rendering of secondary structures. This extended dot-bracket-notation is suitable to denote functional nucleic acids where the primary characteristic is the secondary structure and not the sequence. In doing so we were aiming at:

- backward compatibility to the dot-bracket notation
- use of familiar and mnemonic conventions
- single character operators
- the possibility to describe sets of interacting molecules
- flexibility to chose expressions according to application
- support for rendering with and without colour

Achieving these aims comes at the price of giving up the equivalence of the length between the secondary structure specification and the sequence. On the other hand, the extended notation is often more compact and capable of describing, in a single string, a group of RNA molecules where each molecule varies in length, sequence, and conformation. For our purpose, the ability to use a single string to express sets of interacting molecules outweigh the length mismatch between sequence notation and our extended dot-bracket-notation.

3.2 Extended Dot-Bracket-Notation Syntax

The extended dot-bracket-notation introduces several new symbols which fall into four different categories:

Scoping symbols group sections of the notations. Square brackets "[ ]" are used for grouping a range of base positions. Curly braces delimit alphanumerical parameters for operators and also limit sets of constraints placed on the choice of bases permitted for a particular sequence position. Curly braces can optionally be used to delimit the numbers specifying repetitions.

Operator symbols associate a property with the preceding base position, or grouped range of positions. The properties are mostly used for graphical rendering of the the structure (`_,$,~,@`), but also to mark binding sites for inter-molecular binding (`+`). An operator symbol is always followed by a parameter.

Constraint symbol restrict the possible bases that may be present at the preceding base position. The two constraint symbols are a colon (`:`) which restricts the preceding position to be equal to the base or set of bases that follow it, and a hat (`^`) that restricts the preceding position to differ from the base or set of bases that follow it.

Special symbols are available to express features which cannot be expressed with the above elements. At present only the `%`-symbol is defined, it marks a cleavage-point where the RNA sequence may be hydrolysed.

An overview of the new symbols introduced in the extended dot-bracket-notation is provided in Tab. 3.1. With these enhancements the extended dot-bracket notation can carry a lot more information about an RNA structure than the standard notation while generally leading to a more compact description. However, the translation from the extended dot-bracket-notation to the standard notation is straightforward. This is important as the computational tools for nucleic acids secondary structure have adopted the dot-bracket-notation as their input–output channels (Higgs, 2000). A translation from the standard notation to the extended dot-bracket-notation can of course not make much use of the richer syntax of the extended notation. It is also generally not required, as the

Table 3.1: New symbols introduced in the extended dot-bracket-notation.

	Description	Usage	Comment
[ ]	Grouping of base positions	[.8]@{label A}	Eight unbound bases marked as “label A”.
{ }	Parameter delimiter		see example above
{ }	Set delimiter	.:{A,C}	A single unbound base that can be either A or C.
{ }	Repetition delimiter	A{10}	Always optional.
_	Line width	((([.5]_1)))	Stem-loop structure with bold loop
$	Colour	(3[.2]$1(3.4)3)3	A bulge in red.
~	Line decoration	.24~1(3.3)3	Binding site marked as crinkled line.
@	Annotation marker		See first row.
+	Multi-molecule binding	(24+1(3.3)3	Sticky end of 24 bases, will bind to site marked 1 on other molecule.
:	Base assignment	)):A	Two binding bases, the second one of which is A; See also set delimiter.
^	Base exclusion	(((..^U.)))	Stem loop where the central base in the loop is not a uracil.
%	Clevage point	(((..%..(((	Between bases, i.e., not a base position.

standard notation is a valid subset of the extended notation. Nevertheless, such a translation may be useful to arrive at shorter representations as shown for small examples in Tab. 3.2.

Table 3.2: The extended dot-bracket-notation allows for run-length encoding to achieve a compact representation.

Standard notation	Extended notation	Part of Fig.
...(((((....))))).))))))))	.3(5.4)5.)8	3.4A
(((....))).....((((((((	(3.4)3.5(8	3.4B
(((((....))))).)))))))).	(5.4)5.)8.	3.3A
..)).)))........	.2)2.)3.8	3.3C

Any symbol that can occupy a base position in a sequence (i.e., .,(,),A,U,G,C,T,...) may be followed by a positive integer value n to denote n repetitions of the symbol. This run-length notation is particularly convenient for manually entering secondary structure descriptions. The downside is that the run-length notation can obscure structural motives which may be recognised more readily in the standard notation. A considered use of the repetition parameter will maximise readability, whether by reducing the length of the representation or by deliberately breaking runs of parenthesis into sections that match. For example, (3.2(3.4)6 represents a stem-loop with bulge. The same structure could also be written as (3.2(3.4)3)3. The latter is longer, but preferable nevertheless, because the base-pairing of the two helices is emphasised by breaking the run of six closing parenthesis into two groups of three closing parenthesis each. This example also illustrates that in the extended dot-bracket-notation there is no unique string to describe a given structure. The equivalence of two structures denoted in the extended form, however, can be established by translating both into the standard form, which is easily accomplished.

Specifications for individual base positions, repetitions of these, as well as groups (marked by square brackets) of individual positions and repetitions can be arguments for operators. The operator follows its argument and precedes its parameters. More than one operator/parameter combination may follow an argument and all will be applied to the argument. Table 3.3 provides a few examples of operator use—some of them taken from the structures rendered in Figs. 3.4 and 3.3. Note that in all cases the argument itself, which precedes the operator, is not shown.

Table 3.3: Sample usage of operators.

Notation	Description	Fig.
`~2@{shift region 1}`	Apply decoration type 2 to the preceding region and label it as “shift region 1”.	3.4A
`+3~2`	The preceding argument binds externally with another molecule at the region marked “3” and is drawn with decoration type 2 (“cross”).	3.4C
`+2_1@{OBS2+EFF2}`	The argument (not shown) binds with another molecule at the region marked “2”, draw the binding region in bold (line thickness 1) and label it as “OBS2+EFF2”	3.3C
`+1_1${Red}@{node001}`	The preceding argument binds externally with another molecule at the region marked “1”, render the argument with line thickness 1 in red and label the region as “node001”	-
`~1_2${blue}`	Combination of drawing parameters applied to the preceding argument resulting in a strong bold (thickness 2) crinkled line (type 1) in blue colour.	-
`+{SITE1}~1_1@{match}${red}`	The preceding argument binds externally with another molecule at the region marked as “SITE1”. This region is rendered using a crinkled line with the thickness value 1, and coloured in red. The region is labelled “match”.	-

The acceptable parameters that follow an operator and their semantics are not specified by the notation. A rendering program, for example, may accept a predefined colour number, an explicit colour name, or a hexadecimal RGB value. The corresponding operator with parameters would be `$1`, `${red}`, and `${#FF0000}`. An overview of the extended notation is provided in Fig. 3.2. For clarity, three of the non-terminal symbols occurring in the syntax graph are not shown in Fig. 3.2. The non-terminal *digit* stands for a single digit in the range from 0–9. The non-terminal *alpha* stands for a single

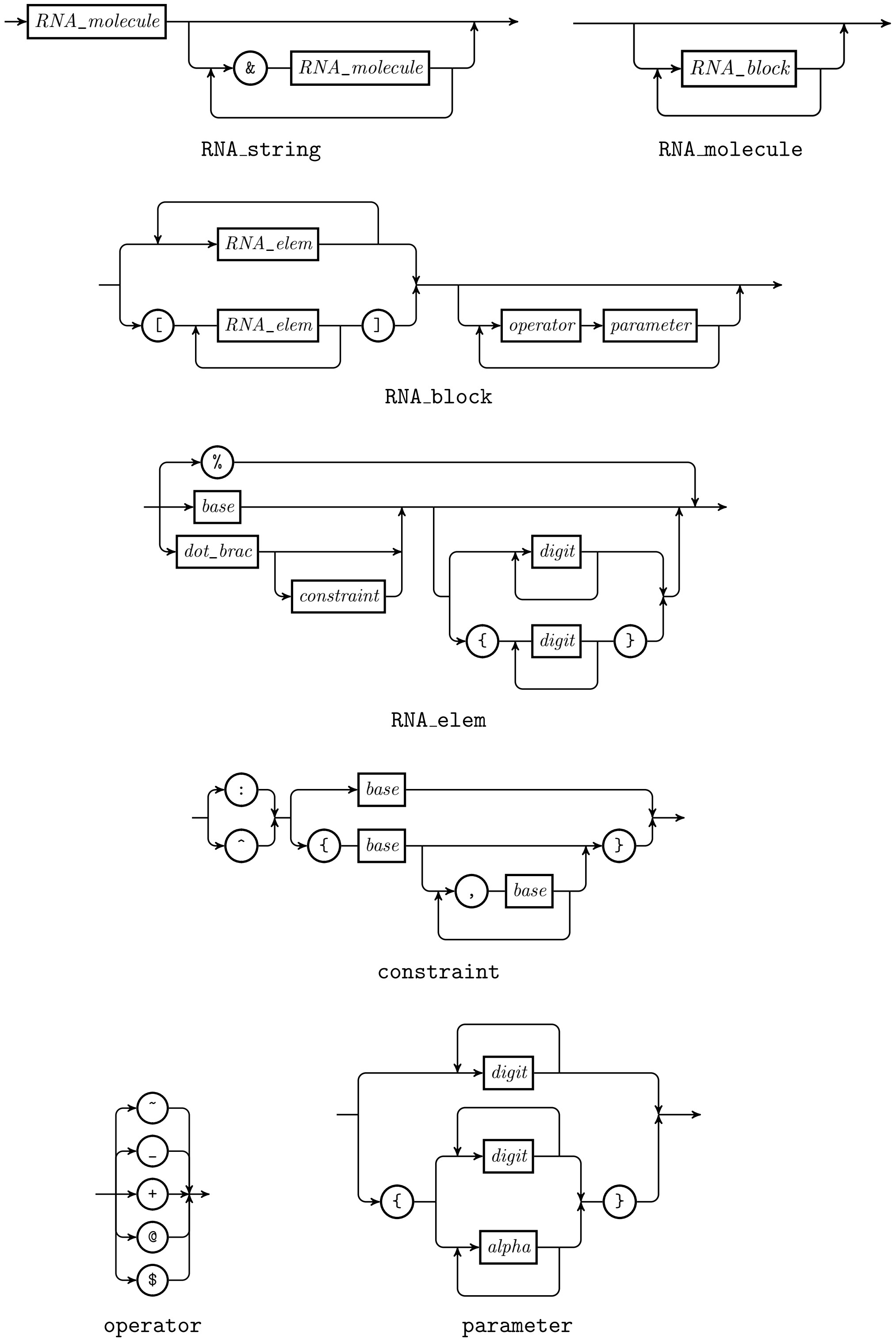


Figure 3.2: Syntax graph for the extended dot-bracket notation.

character from either the range a–z, or A–Z, or a dash (-), underline (_), or space ( ). The non-terminal *base* stands for any one of (A,U,G,C,T,X,N) in upper or lower case.

As can be seen in Fig. 3.2, a single string in the extended dot-bracket-notation can denote more than one molecule (cf. *RNA_string* in Fig. 3.2). The limitations in expressing interactions among multiple molecules in the standard notation was one of the factors motivating the extension described in this chapter. An example with a pair of molecules that bind in two different regions of equal length will illustrate the difficulty of using the standard notation in such cases:

```
((((((.....((((((......))))))...((((....))))...(((((( &
)))))).........(((((....((((....))))....)))))...))))))
```

The two lines represent two different molecules, separated by the `&`-symbol. The regions in which the two molecules will bind to each other are underlined. The dot-bracket-notation is not able to express in which combination the binding regions will bind. In larger molecules the situation can easily be more ambiguous with numerous plausible locations for intermolecular binding. In the extended dot-bracket-notation, the `+` operator can indicate the matching regions. Accordingly, the two molecules shown above can be represented as:

```
(6+{B1}.....((((((......))))))...((((....))))...(6+{B2} &
)6+{B1}........(((((....((((....))))....)))))...)6+{B2}
```

A more relevant example for such an ambiguous binding situation can be seen in Fig. 3.3D, where the two binding sites in the AND gate have the same length. The AND gate uses two effector molecules as input signals and, in its active state, is a three-molecule supramolecular complex. The alphanumeric marking of binding sites in the extended dot-bracket-notation enables the description of interactions among several molecules. Note in the examples above, how the standard notation and the extended notation can be mixed to highlight particular features of a molecule or set of molecules.

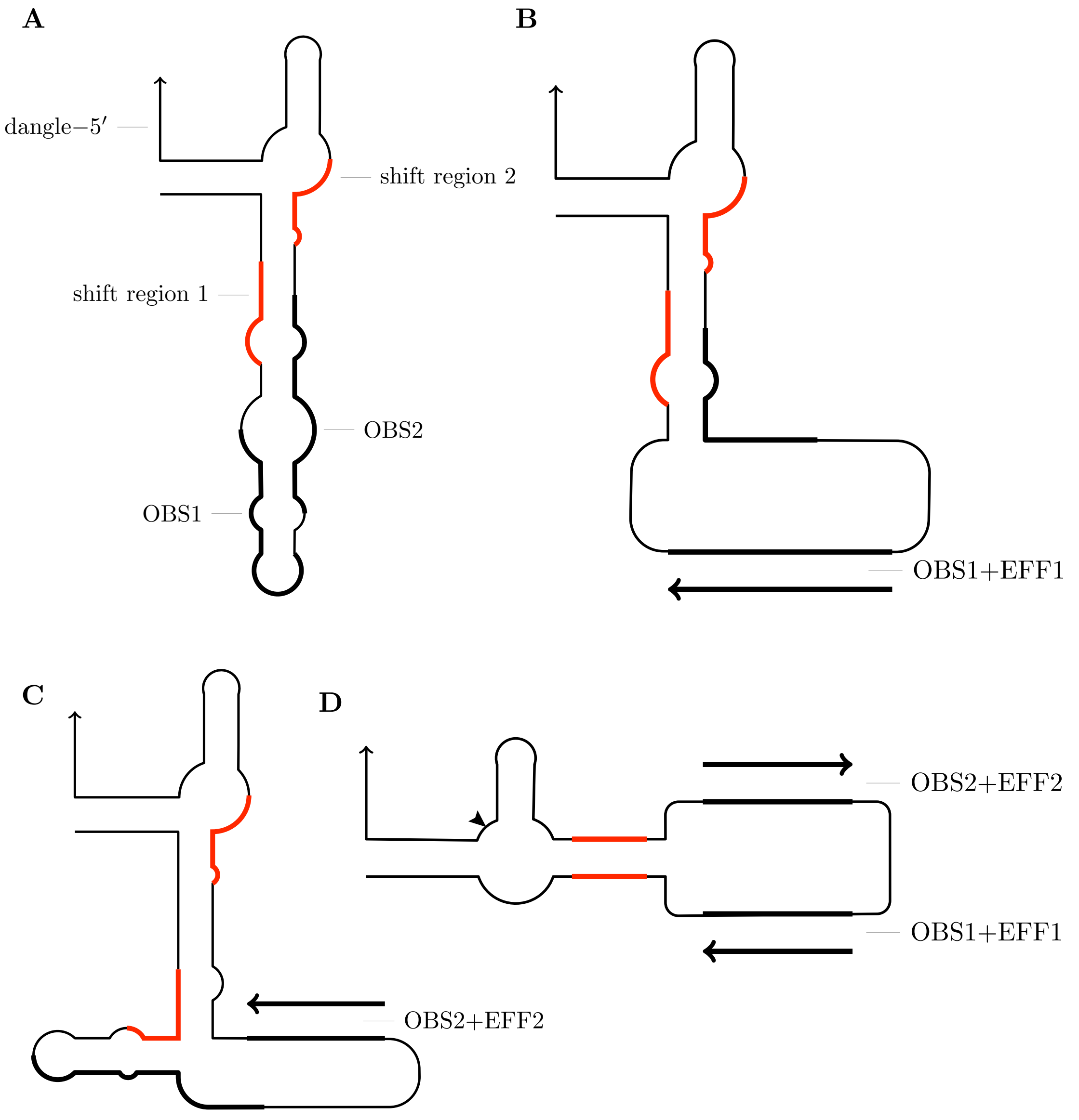
The benefit of the extended notation is most easily seen when it is rendered as two-dimensional structures. We present two different sets of renderings comprising of the two-dimensional structure of the four different states of ribonucleic acids AND gates (cf. Fig. 3.4) and a selection of arbitrary structures from several publications (Fig. 3.3). In Fig. 3.3, renderings of four different states of the ribonucleic AND gate designed by Penchovsky and Breaker (2005) are shown. The interplay of multiple molecules and multiple conformational states is crucial to the computing schemes based on functional nucleic acids. They are also a challenge to represent in a convenient notation. Panels

A shows the secondary structures of the AND gate without effector molecules, panels B and C show the structures the AND gate assumes if only one of the effector molecules is present. If both oligonucleotide binding sites (OBS1, OBS2) are occupied by effector molecules the ribozyme changes into the catalytically active conformation shown in panel D. The extended dot-bracket-notations corresponding to the four states of the gate are shown in panel E. In this example, the presentation aspects (e.g., size, colouring and labelling) of the rendering would be equivalent to the type of figures common in publications. The rendering quality however, in term of the outline of the structures is much smoother and refined compared to the conventional secondary structure representation from RNA drawing tools such as *RNAdraw* (Matzura and Wennborg, 1996) and *RNAmovies* (Giegerich and Evers, 1999). Thus far, the placement of labels defaults to either sides of the structure based on the direction of the rendering.

The positioning of the complementary pair for the external pairing motif defaults to the outer region of the other matching pair. At the moment our rendering tool is unable to handle customised positioning of the external base pairing motif, however one could manually change the position by modifying the Tikz code. Compared to the monotonous representation of Fig. 3.3, Fig. 3.4 shows the rendering of several sample structures from the literature that utilise the colour operator. Panel (A) shows a ribonucleic acid OR gate (Penchovsky and Breaker, 2005) in its active state (i.e., where the binding of any one of two effectors to their respective binding sites steer a conformational change that activated the ribozyme). Panel (B) also depicts the active state of the TRAP strategy for the design of an allosterically controlled hammerhead ribozyme, i.e., where in its inactive state the extended arm of helix I block the conserved bases of the ribozyme (Burke et al., 2002). Panel (C) and (D) illustrate two representation in colour of the ribonucleic acid AND gates from of Penchovsky and Breaker (2005) (cf. Fig. 3.3A and B). The corresponding extended-notation for the four structures is shown in Fig. 3.4E.

In figure 3.4A and D, we show the ability of the renderer to directly plot the bases representing a particular region in the molecule. The bases of the effector molecule and its respective binding site is shown. The ability of specifying a custom decorative symbol is shown in Fig. 3.4B using ”`x`” to indicate the binding region. Complex rendering, however, will make the notation less readable (Fig. 3.4E).

An apparent disadvantage of the proposed annotation is the increased complexity. For human readers this means the addition of more symbols to the syntax, as well as operator scoping of elements that need detailed attention to ensure the correct interpretation of the syntax. On the hand, for machines, establishing the equivalence secondary structures representation denoted by the strings is no longer as simple as extracting the correct pairing of braces. However, we would like to stress that the extended dot-bracket notation inherits the existing framework of its conventional notation. In its most fundamental form, the extended dot-bracket notation is completely similar to the standard notation. The inclusion of additional symbols or even the run length encoding are dependent on



E

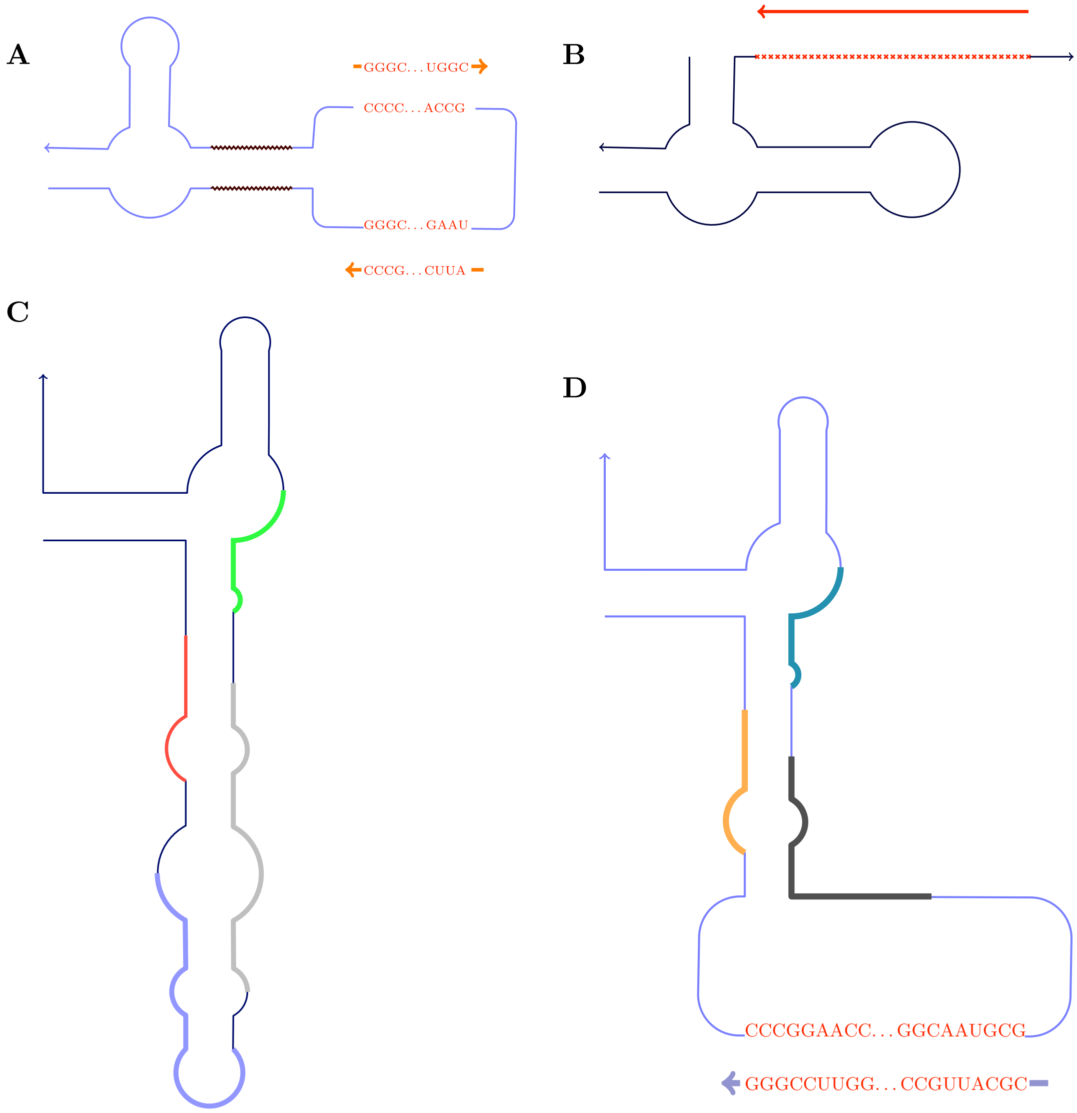
A	`({15}[(4]$1@{shift region 1}[(.4]$1(3.3[.2(3.]_1@{OBS1}[(3.7]_1`
	`)3[.)3.3]_1@{OBS2}[.3)3.2)2]_1)6[.)5]$1[.3]$1@{shift region 2}.3`
	`(5.4)5.)8.9`

B	`({15}[(5.3]$1(3.3[({16}]+1_1.3[.9)3.2)2]_1)5[.)5.2]$1.3(5.4)5.)8.9`
	`& [){16}]+1_1@{OBS1+EFF1}`

C	`({15}[(3.)3.]$1.2(2.3[.2)2.)3.8]_1.3[({16}]+2_1)5[.)5.2]$1.3(5.4)5.)8.9`
	`& [){16}]+2_1@{OBS2+EFF2}`

D	`(8.7[(8]$1.6[({16}]+1_1.3[({16}]+2_1.5[)8]$1.3(5.4)5[.]%)8.9`
	`& [){16}]+1_1@{OBS1+EFF1} & [){16}]+2_1@{OBS2+EFF2}`

FIGURE 3.3: RNA molecular AND gate after Penchovsky and Breaker (2005) in different states. Rendered from the extended dot-bracket notation depicted in (E).



E

A	`[(6.7)]${#87CEFA}(8~1(3.3${#87CEFA}[(16]+1:{GGGC...GAAU}$1.6[(16]+2:{CCCC...ACCG}`
	`$1.3)3)8~1.4(5.4)5.)6${#87CEFA} & [)16]:{CCCG...CUUA}$1_2 & [)16]:{GGGC...UGGC}_2`

B	`[(6]+1.7.(8.5)8.2]${#191970}[(5]+2${#191970}[(16]+3$1~{x}.3${#191970} &`
	`[)16]$1+1_2 & [)5]+2${#191970}.[)7]+1${#191970}`

C	`[(6(4]${#191970}[(4.4]$1_1[(2.3]${#191970}[.3(3.2(2.4]${#87CEFB}_2[)2.1]${#191970}`
	`[.1)3.6)2.2)2]${#778899}_2)4${#191970}[.)2.3]${#ADFF2F}_2[.2(5.4)5.2)6.5]${#191970}`

D	`[(6(4]${#87CEFA}[(4.4]_2${#FFA07A}[(2.5]${#87CEFA}[(16]:{CCCGGAACC...GGCAAUGCG}+1`
	`[.10]${#87CEFA}[.6)2.2)2]${#2F4F4F}_2[)4]${#87CEFA}[.)2.5]_2${#20B200}`
	`[.2(5.4)5.2)6.5]${#87CEFA} & [)16]+1:{GGGCCUUGG...CCGUUACGC}_2$1`

Figure 3.4: Rendering of four sample nucleic acids molecules, and the extended dot-bracket notation. RNA OR logic gate from (Penchovsky and Breaker, 2005) (A), Allosterically control nucleic acids with TRAP strategy (B) (Burke et al., 2002), RNA AND logic gate from (Penchovsky and Breaker, 2005) (C and D) and the corresponding extended dot-bracket-notation (E).

the type of application. For instance, the use of iterator operators make it easier for the users to compare the length of hybridised regions within a molecule, but the relative lengths of oligonucleotides is more visible, if denoted without the iterator operators.

The notation is flexible enough to be translated from standard notation to the extended notation and vice-versa. The translation from standard notation to the extended notation only shortens the dot-bracket following the run-length encoding. The inclusion of interaction and rendering information must be made explicitly into the extended notation after conversion. Translation from extended notation to standard notation will loose the rendering information. However, the integrity of the molecular conformation remains intact. We developed an *RNAparser* tool to aid in the translation process between the standard to the extended notation (and vice versa) together with a structure renderer that translates these extended notations into conventional secondary structure diagrams.

Chapter 4

Inverse Prediction of Nucleic Acids

4.1 Introduction of RNA Sequence Designer

The mapping of RNA structures to their sequences involves the process of designing RNA sequences that fold into a predefined conformation. RNA sequence design works is the reverse operation to secondary structure prediction. The secondary structure is known and the task at hand, is to generate possible sequences that fold to form this structure. Both secondary structure folding and sequence design can be depicted as an optimisation problem. In the simplest form, for folding one has to minimise the free energy value belonging to different loop motifs to arrive to the most likely structure. Instead of minimising the free energy, the sequence design problem minimises the distance value $\mathcal{D}(S_1, S_2)$ of two RNA secondary structures, S_1 and S_2, where one represents the conformation for a sequence candidate, while the other represents the reference structure.

Popular distance measures are the “base pair distance” ($\mathcal{D}_{bp}$) (Wuchty et al., 1999) and the “hamming distance” ($\mathcal{D}_h$) (Hofacker, 1994). The “base pair distance” counts the the number of base positions in S_1 that is not paired to the same position as in structure S_2 and vice versa. For example, given two structures (in dot-bracket notation) $S_1 =$ `(((...)))` and $S_2 =$ `.(((..)))`, the base pair distance $\mathcal{D}_{bp}$ is 6 nt. If $S_3 =$ `.((...)).` then $\mathcal{D}_{bp}(S_1, S_3) = 1$ nt. The “hamming distance” counts the number of base positions in which the two sequences differ, i.e., the minimum number of base position mutations needed to convert S_1 into S_2. Using the three structures S_1, S_2 and S_3 given earlier, the hamming distance between S_1 and S_2 and the hamming distance between S_1 and S_3 is 2 nt ($\mathcal{D}_h(S_1, S_2) = \mathcal{D}_h(S_1, S_3) = 2$ nt). Unlike the base pair distance, the hamming distance counts the position by position difference between the two structures.

Using a distance measure (i.e., base pair or hamming distance) as the objective function, a sequence $\mathcal{X}$ folds into the given structure S, if and only if, the distance $\mathcal{D}$ between the target structure S and the folded candidate sequence $\mathcal{X}$ vanishes, e.g., $\mathcal{D}_{bp}(S, \mathcal{F}(\mathcal{X})) = 0$ or $\mathcal{D}_h(S, \mathcal{F}(\mathcal{X})) = 0$, where $\mathcal{F}(\mathcal{X})$ denote the secondary structure folding of $\mathcal{X}$. The optimisation of distance $\mathcal{D}$ is used in three RNA sequence design algorithms, *RNAinverse* from the *Vienna RNA package* (Hofacker et al., 1994), *RNAdesigner* from the *RNAsoft* suite (Andronescu et al., 2004), and *INFO-RNA* (Busch and Backofen, 2006). There is a common theme across the three RNA sequence design algorithms of *RNAinverse*, *RNAdesigner* and *INFO-RNA*. Each conducts a recursive search by initialising one probable candidate that will later undergo an iterative refinement process that involves bases mutation within the initialised sequence. Different heuristic local search strategies and initialisation procedures are implemented in each algorithm.

RNAinverse uses an unbiased uniform-base assignment in initialising the the start sequence, coupled with an adaptive walk strategy for refinement. During refinement, base mutations can apply to any position in the sequence, but only base mutations that lower the distance are kept. *RNAdesigner* and *INFO-RNA* employs a more selective initialisation strategy. Both tools agreed that a good start candidate is significant for solving the inverse prediction problem. In *RNAdesigner*, a probabilistically-biased initialisation method is implemented to ensure that appropriate base assignments are made in the unpaired and paired regions within the strand. *INFO-RNA*, in contrast, focuses on initialising a sequence with the lowest free energy. This is accomplished by iteratively selecting base pairs (to fill the paired regions), and bases (for the unpaired regions) that lower the free energy of the folded sequence. The calculation of free energy for the partial sequences at each iteration during the initialisation procedure is similar to the one discussed in Section. 2.4.1.2 with the target structure as reference. In refinement, *RNAdesigner* employs a stochastic local search strategy (SLS). Meanwhile, a derivative of SLS is later implemented in *INFO-RNA*. Detailed description of these RNA sequence designers is discussed next.

RNAinverse

RNAinverse is an algorithm available as part of the Vienna RNA Package (Hofacker et al., 1994), one of the most popular and widely used packages for both RNA folding and sequence design. The algorithm starts with an unbiased initial sequence, and uses a simple adaptive walk heuristic to refine the sequence so that the distance value is minimised. The initialisation procedure was designed unbiased to generate sequences that can be arbitrary, intended for the statistical study of the distribution of sequences with a common secondary structure across the sequence space (Hofacker, 1994). The adaptive walk randomly produces a single mutation that is accepted if and only if, the distance value $\mathcal{D}$ between the candidate structure and the target structure improves. The mutation can either be a change of a single base in an unpaired position, or a single base pair replacing the previous pairs. Iteratively, the adaptive walk strategy is applied to the candidate sequence until the termination condition is met. This can either be, when a solution is found, or until a certain number of steps have been completed. The candidate sequence resets after a fixed number of trials. An outline of the algorithm follows,

Algorithm 1 *RNAinverse* as proposed in Hofacker et al. (1994)

```
initialise start sequence S ← uniform distribution
Structural decomposition of S
count distance $\mathcal{D}$
while $\mathcal{D}$ not = 0 or step not = MAX do
    mutate S = S$_1$ ← random change of base or base pair
    count distance $\mathcal{D}_1$ for S$_1$
    if $\mathcal{D}_1 \leq \mathcal{D}$ then
        replace S = S$_1$
    end if
end while
```

In Algorithm 1, the dissimilarity value of the two structures is counted based on the "hamming distance" ($\mathcal{D}_h$), explained earlier in this section. "MAX" denotes the maximum number of iteration for termination of the algorithm, if $\mathcal{D} \neq 0$. At each iteration, the *RNAfold* program is invoke to generate the secondary structure conformation of the candidate sequence before the distance of the two structures can be calculated. An alternative objective function is suggested in (Hofacker, 1994),

$$E(\mathcal{X}, S) - E(\mathcal{X}) \geq 0 \tag{4.1}$$

where, $E(\mathcal{X})$ represents the minimum free energy of candidate sequence $\mathcal{X}$ and $E(\mathcal{X}, S)$ denotes the minimum free energy for candidate sequence $\mathcal{X}$, given that it folds to the target structure S. The latter can be evaluated using the program *RNAeval* from the *Vienna RNA package* (Hofacker et al., 1994). Compared to the distance measurement, this energy dependent objective function can be lowered incrementally at each iteration. Thus increasing the chance of arriving to the target structure with the expense of increasing the CPU time, because many steps are required to reach the optimal solution.

This algorithm assumes that it is likely, (although not guaranteed) that the optimal solution can be found by solving smaller optimisation problems involving each substructure of the molecule. The target structure is broken down into smaller fragments, and the algorithm then solves the problem for these fragments individually. The subsequences generated from these smaller problems, are concatenated together to form the full candidate sequence. For long structures, this approach not only reduces the likelihood of getting stuck in a local minimum, but also reduces the number of calls needed for the folding programs *RNAfold* to fold the full length sequences (Hofacker, 1994). However, Andronescu et al. (2004) conclude that the random adaptive walk heuristic supported by Hofacker et al. (1994), increases the likelihood of the candidate sequence to be trapped in a local minimum.

RNAdesigner (RNA-SSD)

RNA Secondary Structure Designer or *RNAdesigner* (Andronescu et al., 2004) is distributed as part of the *RNAsoft package* (Andronescu et al., 2003). In contrast to the *RNAinverse* approach, the *RNAdesigner* is a biased approach that favours the formation of C-G and G-C base pairs and the selection of base "A" for the unpaired regions. However, the probability biases assigned for each unpaired base and base pair combinations are well-balanced to avoid the generation of homogeneous sequences comprised of only three bases (A,C and G). This probability bias influences the initialisation routine, as well as the mutation operator during the heuristic procedure. The algorithm is outlined as follows:

Algorithm 2 *RNAdesigner* as proposed in Andronescu et al. (2004)

```
SS ← hierarchical decomposition of target structure
for each $ss_i$ in SS do
    initialise start sequence S ← probability biases with tabu mechanism
    count distance $\mathcal{D}$
    while $\mathcal{D} \neq 0$ or step $\neq$ MAX do
        if step == $\mathrm{MAX}_{INIT}$ then
            initialise start sequence S ← probability biases with tabu mechanism
        else
            mutate $S = S_1$ ← randomised first-improvement strategy
            count distance $\mathcal{D}_1$ for $S_1$
        end if
        if $\mathcal{D}_1 \leq \mathcal{D}$ or KEEP then
            replace $S = S_1$
        end if
    end while
    return $S_{all}$ ← S for $ss_i$
end for
```

Similar to *RNAinverse*, *RNAdesigner* uses structural decomposition routine divide the structure into smaller substructures, again to improve accuracy (in terms of difference between the target structure and structure generated from sequence candidates) and processing time. Unlike the structural decomposition routine of *RNAinverse*, in *RNAdesigner* a structure is recursively split into two substructures in each decomposition step, resulting in a binary decomposition tree. At the root of the decomposition tree is the target structure, and at each level two substructures as branches. Each of these subsequences will undergo individual optimisation using its substructure as target.

The hamming distance $\mathcal{D}_h$ measure is chosen as the objective function. An additional MAX$_{INIT}$ variable represents the maximum number of iterations before re-initialisation of the candidate sequence takes place. The KEEP operator is assigned a probability value and when this value is met, allow the algorithm to select a mutated sequence regardless of the distance value counted for its folded structure and the target structure.

A comparison study by Andronescu et al. (2004), reports a significantly better results for *RNAdesigner* when compared to *RNAinverse*. The unbiased uniform-base initialisation procedure of *RNAinverse* produces a more random distributed sequence, and as speculated by Andronescu et al. (2004), this randomness is one of the factors that deter the sequence from its optimal path, since sequences that are too far away from the target conformation can be trapped in local minima. Andronescu et al. (2004) also speculated that the random adaptive walk strategy implemented by *RNAinverse* for its refinement process is insufficient in tackling the problem of local minima. To resolve this issue, the *RNAdesigner* algorithm applies a probabilistic biased initialisation approach (based on well understood principles of RNA folding) and a stochastic local search (SLS) heuristic for its refinement. The following rules are applied in the initialisation procedure:

- Base pairs in the target structure are assigned complementary bases, ensuring that the sequence would at least have the right canonical pairing (cf. Section. 2.4.1.1) distribution;
- Unpaired sequence positions are assigned non-complementary bases, to prevent unwanted helix extensions;
- Because C-G and G-C base pairings are energetically more favourable than A-U, U-A, G-U and U-G pairings, applying more C-G and G-C pairs in the helix regions, it is more likely that the helix regions can be preserved in the MFE structure
- Fixed base combinations for loop motifs are assigned to contiguous segments of the target structure, to minimise potential for undesired interactions between subsequences

The stochastic local search (SLS) heuristic applies single base mutation on the candidate sequences, especially for conflicted base (i.e., base that are paired in its MFE structure, but not in the target structure) chosen randomly in the candidate sequence. However base mutations outside of the conflicted regions is also permitted. As depicted in Algorithm 2, the SLS heuristic also allows for the worse candidate to be kept subject to the KEEP operator. The results reported by Andronescu et al. (2004) indicates the advantage of the probability biased initialisation as the initialisation allows the algorithm to start with a candidate sequence that is close the target structure. This is achieved by applying known principles of RNA hybridisation as discussed above. The heuristic of SLS on the other hand, guides the refinement procedure towards the target structure and aids in escaping from local minima.

INFO-RNA

INFO-RNA, short for *INverse FOlding-RNA* by Busch and Backofen (2006) is the latest offering for the inverse prediction problem. The algorithm comprises of a MFE based initialisation procedure and a derivative of the stochastic local search (SLS) heuristic of (Andronescu et al., 2004). Compared to the two algorithms discussed earlier, the initialisation procedure of *INFO-RNA* is a bit more complicated. Given the base pairing requirement of the target structure, an assignment of a base pair to a set of base positions is made only if the base pair that will be added lowers the free energy of the target structure. This is done by estimating the free energy of the structure generated from the candidate sequence (with the inclusion of the suggested base pair) using *RNAEval* from the *Vienna RNA package*. The assignment of bases for the unpaired base positions is made in a similar manner. In INFO-RNA, a dynamic programming algorithm is implemented to generate the free energy estimates for every possible base pair (for paired positions) and base (for unpaired positions) combinations according to the target structure. The base pair and base combinations with the lowest free energy are then selected as the start sequence. The outlined of the algorithm is as follows:

Algorithm 3 *INFO-RNA* as proposed in Busch and Backofen (2006)

```
initialise start sequence S ← lowest MFE assignment
decompose S into sub-structures
count distance $\mathcal{D}$
while d not = 0 or step not = MAX do
    find and preorder neighbour set ← one-step look ahead
    mutate S = S$_1$ ← from the set of neighbour
    count distance $\mathcal{D}_1$ for S$_1$
    if $\mathcal{D}_1 \leq \mathcal{D}$ or KEEP then
        replace S = S$_1$
    end if
end while
```

Unlike *RNAinverse* and *RNAdesigner*, the "base pair distance" ($\mathcal{D}_{bp}$) value is used as the objective function. The "KEEP" operator in the algorithm denotes the same function as in the *RNAdesigner* algorithm (cf. Algorithm 2). Although the refinement heuristics of the two algorithms is relatively similar, *INFO-RNA* employs a neighbour selection method as its mutation operator in which the neighbours of the current sequence are tested to see if they offer any improvement with regard to the objective function. Any sequence that differs either by a single unpaired base or a single base pair is classified as an immediate neighbour of the candidate sequence. Given the value of the "base pair distance", a set of sequences with base pair and base mutations that is assigned to the positions where mismatches occurred, is identified. This set of sequences is called a neighbour set and it is sorted according to the free energy estimates generated for each neighbour sequence. Recursively, the distance of each neighbour sequence is measured and an immediate neighbour sequence that lowers the distance value is later selected for the next iteration. Because the set of neighbour sequences is sorted based on their free energy estimate, the selection of an immediate neighbour sequence should pick up a sequence that not only lowers the distance value but possesses a lower free energy estimate among other sequences with lower distance values in the set.

4.2 The Development of RNA Sequence Designer

In the context of designing nucleic acid units for information processing tasks, the generation of sequence candidates that predictably conform to a target structure is important. A type of computing unit that is being considered in this research is illustrated in Fig. 2.8. The conformation a molecule assumes is affected by its physio-chemical environment including cosolutes such as ions. The binding with another molecule has a large effect on a molecule's environment and consequently binding events can cause a change in conformational state. As illustrated in Fig. 2.8, a functional molecule with a binding site which, if occupied by an effector molecule, will facilitate a conformational change that in turn gives rise to a change in function, is said to be allosterically controlled.

As discussed earlier in Chapter 2, ribozymes can act as catalysts. The catalytic activity may be enabled or suppressed upon binding of a nucleotide strand (cf. Fig. 2.9). As illustrated by the RNA AND logic gate presented in Fig. 2.8, it is possible to design and fabricate ribozymes endowed with multiple interacting effector biding sites. On one hand, the combinatorial variety of nucleic acid strands allows for the numerous different effector molecules and accordingly facilitates the independent parallel operation of several allosterically controlled ribozymes. On the other hand, the fact that such ribozymes can have the same type of molecules, i.e., RNA oligonucleotides, as effectors and as products of the reactions they catalyse, opens up a path to cascading several processing stages for molecular signals.

As part of our effort to develop a computer-aided design procedure to construct nucleic acid units for information processing, the evaluation of various sequence design algorithms against a restricted design space is required. The design space is derived from the structural property of small catalytic RNAs, artificially engineered DNA enzymes and nucleic acid logic gates that have been constructed in the laboratory. In the design space, the length of the molecules are limited to 200 nt and the structural motifs are specifically defined in term of their length and the number of occurrences. Compared to the naturally occurring RNA molecules, the conformations of the type of molecules that belong to the design space are less complex. For instance, the conformation of a 16S rRNA (small subunit ribosomal RNA) which is 491 nt in length comprises a number of nested multibranch loops and nested internal loops (Wuyts et al., 2002). A study to compare the accuracy of *RNAinverse* and *RNAdesigner* has been conducted by Andronescu et al. (2004), this study however, focuses on structures ranging from 165 to 850 nt to cater for the design of naturally occurring RNA molecules. In order to find the sequence design algorithm that is best suited to our restricted design space, an investigation is conducted to compare the accuracy of *RNAinverse*, *RNAdesigner* and *INFO-RNA*.

In our trial runs, *RNAdesigner* is the most accurate when compared with *RNAinverse* and *INFO-RNA*. During the adaption of *RNAdesigner*, we found that we can simplify the algorithm, and thus reduce the CPU time. We also tuned the probability biases of assigning base pairs and unpaired bases specifically to suite our restricted design space. Details of the simplified version of *RNAdesigner*, named *StochSrch*, are discussed in the next section. In contrast to the optimisation scheme implemented by *RNAinverse*, *RNAdesigner* and *INFO-RNA*, we suggest an alternative approach named *RepInit* that focuses only on the initialisations, without any refinement procedure (with the application of rules and probability biased to ensure that the base pairing and unpaired base requirements of the target structure are met) to eliminate the problem of local minima. Details of the implementation of *RepInit* are discussed after *StochSrch*. Following *RNAdesigner*, the hamming distance measure ($\mathcal{D}_h$) is used in both *StochSrch* and *RepInit*. A comparison of *RNAinverse*, *RNAdesigner*, *INFO-RNA*, *StochSrch* and *RepInit* concludes the chapter.

Stochastic Search (*StochSrch*)

As an initiative to reduce the CPU time of the *RNAdesigner* algorithm in our restricted design space, we derived a simplified version of the *RNAdesigner* program that we named *StochSrch*. Given the length restriction of the sequence space of interest the hierarchical decomposition procedure which split RNA structure into smaller substructures in *RNAdesigner* (cf. Algorithm 2) is omitted in *StochSrch*. The algorithm originally divides RNA structures into sub-structures and recursively generates sequences for these sub-structures, building into larger sub-structures and subsequently, forms the sequence of the complete structure. When we run *RNAdesigner* without the hierarchical decomposition procedure in our trial runs, i.e., by increasing the sequence length that triggers the decomposition to 200 nt, the accuracy is unchanged compared to running the algorithm with the decomposition procedure turned on.

This procedure is intended to reduce the complexity of iteratively folding a long RNA sequence. Each call to the folding algorithm has a time complexity of $O(n^3)$, where n denotes the length of the structure. The default value to trigger the hierarchical decomposition procedure is 70 nt, with the minimum length of substructures being 30 nt. Andronescu et al. (2004) report that the CPU time of *RNAdesigner* increases when a substructure is hard to solve, i.e., trapped in local minima. Because our design space is limited to 200 nt and the majority of the nucleic acid computing units that have been developed range between 80 to 150 nt (Stojanovic and Stefanovic, 2003b; Penchovsky and

Breaker, 2005), we decided to exclude the hierarchical decomposition in our development of *StochSrch* to reduce possibility of the substructure sequence designs to be trapped in local minima. The variant algorithm is outlined as follows:

Algorithm 4 Minimisation of *RNAdesigner* that yields *StochSrch*

```
~~SS ← hierarchical decomposition of target structure~~
initialise start sequence S ← probability biases ~~with tabu mechanism~~
count distance $\mathcal{D}$
while $\mathcal{D} \neq 0$ or step $\neq$ MAX do
    if step == $\text{MAX}_{INIT}$ then
        initialise start sequence S ← probability biases ~~with tabu mechanism~~
    else
        mutate $S = S_1$ ← randomised first-improvement strategy
        count distance $\mathcal{D}_1$ for $S_1$
    end if
    if $\mathcal{D}_1 \leq \mathcal{D}$ or KEEP then
        replace $S = S_1$
    end if
end while
```

StochSrch uses the basic initialisation procedure of *RNAdesigner* (cf. SeqInit–a type initialisation in (Andronescu et al., 2004)) instead of the default procedure that requires an implementation of a tabu mechanism for assigning unpaired motifs. The tabu mechanism is intended to minimise the potential for undesired but energetically favourable interactions to occur between subsequences of a target structure (Andronescu et al., 2004). The mechanism assigns short base combinations to the unpaired region of the sequence. Given a list of possible base combinations for the unpaired region, only a base combination that does not forms base pairs with the previous base combination assignment for another unpaired region is selected. This mechanism was first introduced by Heitsch et al. (2003).

As the length of the molecule becomes shorter, the stretches of unpaired regions for the molecule also decreases. This fact, together with the exclusion of the decomposition procedure are the two main reasons that support the cancellation of the tabu mechanism. We also anticipated that undesired folding can be handled by increasing the probability values of both C-G and G-C base pair and, the base "A" for unpaired assignment. These changes are aimed at reducing the CPU time of the algorithm and we expect that these changes should not affect the accuracy of the algorithm in our restricted design space. The recursive stochastic local search heuristic of *RNAdesigner* for the refinement stage was left untouched.

Repeated initialisation (*RepInit*)

In general, there is a large number of arbitrary RNA sequences that can fold to a given secondary structure (Schuster et al., 1994), and accordingly there is an enormous possibility for a local heuristic search to become stuck in local minima. A comparison study by Andronescu et al. (2004) concludes that both the initialisation procedure and the mutational operators can easily disarray a candidate sequence from arriving at its optimal state. In another variant of *RNAdesigner*, which we called "Repeated initialisation" (*RepInit*), we focus on the initialisation procedure, instead of the heuristic for optimising a candidate sequence. In order to fulfil the base pairing and unpaired base requirements of the target structure, a set of rules based on known principles of nucleic acids hybridisation is applied.

The canonical base pairing rule (i.e., the formation of base pair for C-G, G-C, A-U, U-A, G-U and U-G) and the base pairing precedence (i.e., C-G and G-C are more favoured than A-U and U-A, which in turn is more favoured than G-U and U-G) becomes the primary rule of the initialisation procedure. In order to maintain a level of randomness of the generated sequence, we assign biases to each base pair and each unpaired base according to the favoured base pairing rules mentioned earlier. The initialisation procedure starts by assigning base pairs to each of the paired base positions according to the probability biases implemented in *RNAdesigner*. Once the paired base positions are filled, a set of rules is applies to the candidate sequence. If any unassigned base positions remain after the rules have been applied, an assignment of bases according to the probability biases is conducted. The rules are applied for the second time once the remaining positions are filled. There is no heuristic search to optimise the candidate sequence. If mismatches occur between the structure folded from the candidate sequence and the target structure ($\mathcal{D} \neq 0$), the algorithm simply discards the sequence, and initialises a different sequence. As in *StochSrch* algorithm, we have excluded the hierarchical decomposition procedure. The algorithm is as follows:

Algorithm 5 Repeated initialisation (*RepInit*) algorithm

```
initialise start sequence S ← probability biases and set of rules
count distance $\mathcal{D}$
while $\mathcal{D}$ not = 0 or step not = MAX do
    re-initialise start sequence S ← probability biases and set of rules
    count distance $\mathcal{D}$
end while
```

From our observation of RNA sequences that were generated during the trial runs for *RNAdesigner*, we derived a set of rules to govern the assignment of base pairs and unpaired bases for the target structure. So far, only five rules have been defined. Because the set of rules only applies when certain conditions are met, then as a start point, we implement the probability biased approach to assign base pair to the paired base position. The set of rules is listed as follows:

1. If there exists a region of four or more consecutive base pair positions, then half of the base pairs in that region must be assigned either C-G or G-C base pairs.
2. For stretches of unpaired regions, we disallow any occurrences of consecutive bases of C,G or U (e.g., `CC`...`C`).
3. For each unpaired base position, it is mandatory that no complementary base pairing can be formed with the base located 3 position from it, given current base position i, then if $h = i - 3$, then $\alpha_i \alpha_j \notin \mathcal{B}$ (cf. Sec. 2.4.1.1).
4. For any base positions that come immediately before or after a base pair (in dot-bracket, `X(....)X`, where `X` represent before and after position), an identical base is assigned to both positions.
5. For any two unpaired base positions, if one has been assigned G, C or U, then the other position is assigned base A.

Base positions that do not trigger any of the above rules are assigned bases according to the probability biases suggested by the "SeqInit" procedure of (Andronescu et al., 2004). Unlike the assignment of base pairs where one can easily choose from any base pair combination in $\mathcal{B}$ (cf. Sec. 2.4.1.1), special attention is required for the base assignment of the unpaired positions as seen in the rules above.

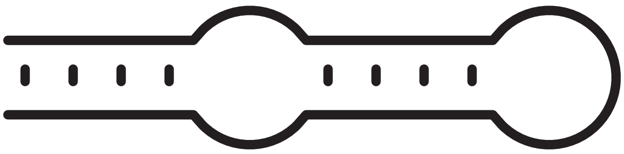
For the paired base positions, the unbiased approach of *RNAinverse* allows any base pair combinations to be assigned, while *RNAdesigner* favours an assignment of C-G/G-C base pairs compared to A-U/U-A and G-U/U-G pairs. *INFO-RNA* also prefers a selections that biased to the base pair that lowers the free energy of the structure. Among the three, the probability biased method of *RNAdesigner* is the most attractive as it enforces the base pairing rules that favours the more stable C-G/G-C pairings over A-U/U-A and the wobble pair of G-U/U-G (cf. pg. 12) in a subtle manner thus allowing sequence diversity. In addition to the probability bias, rule 1 ensures that half of the base pairs in four or more consecutive paired bases are assigned C-G/G-C base pairs. Motivated by the fact that the presence of C-G/G-C base pairs (that binds with three hydrogen bonds) will stabilise helices (Nowakowski and Tinoco, 1997; Tinoco and Bustamante, 1999; Moore, 1999).

The main purpose of the rules for the unpaired region is to prevent the formation of unwanted base pairs that may be energetically more favourable to form compared to the intended pairs of the target structure. For Rule 2, we rejects the formation of consecutive C's or G's or U's in the unpaired regions to reduce the likelihood of unwanted folding. In the nearest neighbour models, the stability of a given base pair depends on the identity of the adjacent base pair (Xia et al., 1998). If given a sequence where stretches of C's or G's are present in the unpaired positions, following the nearest neighbour model, if there exists another short stretch (more than 2 nt) of consecutive C's or G's then the formation of base pairs which are thermodynamically more stable are likely to occur. Resulting into a mismatch pairing in the secondary structure. Some of the thermodynamically stable base pairs are:

$$\begin{array}{cccc} 5' - \mathrm{GC} - 3' & 5' - \mathrm{CG} - 3' & 5' - \mathrm{GG} - 3' & 5' - \mathrm{CC} - 3' \\ 3' - \mathrm{CG} - 5' & 3' - \mathrm{GC} - 5' & 3' - \mathrm{CC} - 5' & 3' - \mathrm{GG} - 5' \end{array}$$

For stretches of U's in the unpaired positions, there is a high likelihood of unintended formation of base pairs in the unpaired positions because the algorithm assigns a higher probability bias for base A to be allocated in the unpaired positions.

In rule 3, we directly check that no immediate base pairs within a range of 3 nucleotides can form. This test is made regardless of the region where the previous 3 and next 3 base position reside. This is to prevent a formation of base pair with base position in either



5'-((((....((((....))))....))))-3'

Rule 1	GG((....(CC(....)GG)....))CC
	⋮
Rule 2	GGGA....ACCG.C$\overset{\boxed{A}}{\emptyset}$.CGGU....UCCC
Rule 3	GGGA..$\overset{\boxed{A}}{\emptyset}$.ACCG.CA.CGGU....UCCC
Rule 4	GGGA..A.ACCG.CA$\boxed{A}$CGGU$\boxed{A}$...UCCC
Rule 5	GGGA..A.ACCG.CAACGGUA.U$\boxed{A}$UCCC
	⋮
End	GGGA*agaa*ACCG*gcaa*CGUU*acua*UCCC

FIGURE 4.1: An example of a rule-based initialisation in *RepInit*. The target structure comprises of two helices, a hairpin (at the far right) and a symmetrical internal loop (in the middle). In the figure, we depict the five sample conditions that will trigger the set of rules listed on page 67. This is a sample scenario that invoked the five rules and other base assignments are made using the probabilistic biased approached. A cross character with an attached box denotes the substitution of a base. For the final sequence, capital denotes base pairing, and lower case characters in italics stand for unpaired bases alphabets in italics represent unpaired bases.

directions. In rule 4, we assign the base positions that come immediately before and after a paired region with identical bases to prevent base pair formation. For instance given the following RNA sequence, . . . `XCCCCX`. . . `XGGGGX`. . . , where `C` and `G` forms a base pair, then the base position of `X` is assigned an identical base. This should reduce the chance of undesired folding because each position of `X`, now acts as a terminal base where no more adjacent base pair can form. Finally, rule 5 fills stretches of unpaired regions, effectively increasing the probability of assigning base A for unpaired positions.

Sample run of the initialisation procedure is illustrated in Fig. 4.1. It shows the evolution of the sequence as the five rules listed above are applied. The target structure is chosen such that all five rules (cf. pg. 67) are invoked. Remaining base positions that do not trigger any of the rules will be assigned with fixed probabilistic bias. If the target structure is not acquired ($\mathcal{D} = 0$), then *RepInit* is executed until a maximum number of trial (MAX) is reached. The program immediately terminates once the target structure is acquired. The algorithm ignores the optimisation schemes implemented by the different approaches presented earlier. As a consequence it cannot become trapped in local minima. The algorithm relies solely on the initialisation procedure to generate sequences that should conform to the target structure. In our implementation, the rules are not fixed. Rules can be added, substituted or removed to suite a particular structure domain or to improve accuracy. In addition, one can also tune the base composition (i.e., adjusting the probability rate for bases and/or base pair) in order to increase or decrease the structural stability of the target structure.

4.3 Evaluating the Performance of Sequence Designers

To access the usefulness of the two modified sequence design algorithms, we evaluate the performance of *RepInit* and *StochSrch* against *RNAinverse*, *RNAdesigner*, and *INFO-RNA* using sets of molecular computing units from the literature and sets of artificially generated structures from the structural space defined in Table 4.1. The structure

TABLE 4.1: List of parameters for the generation of artificial RNA molecules derived from the structural space compiled in Tab. 4.2. The size of the generated RNA structures varies between 50 and 150 nt. The occurrence of a particular structural motifs in the structure is given as the number of elements. For instance, as depicted in the table, sample structures from the table can have up to 5 bulges, with the minimum size of 4 nt and maximum size of 25 nt.

	No. of Elements	Length(nt)
Helix	–	4–25
Hairpin Loop	–	4–25
Internal Loop	0–3	4–25
Bulge	0–5	4–25
Junction	0–3	4–5

described in the table comprises sets of biomolecular logic gates and natural occurring Ribozymes with structural characteristics that are suitable for the construction of computational nucleic acids. A summary of the properties of these structures is depicted in Tab. 4.2. Furthermore, from this analysis, we compiled a simpler table of parameterisation (cf. Tab. 4.1) that should be sufficient for the structural space intended for the construction of computational nucleic acids. This parameterisation enables the generation of artificial structures with characteristics similar to the conformations that could either resembles the structural motifs of RNA or DNA enzymes or completely distorted into some random structures that can be classified as a meta-stable conformations of molecules with RNA and DNA enzyme motifs embedded.

Table 4.2: Summary of RNA structural characteristics. A survey 13 biomolecular gates and the natural occurring hammerhead ribozyme and hairpin ribozyme is provided. The 13 biomolecular gates comprise both DNA and RNA molecules complied from (Stojanovic et al., 2002; Stojanovic and Stefanovic, 2003b; Penchovsky and Breaker, 2005). The notation $\langle x_i, x_{ii}, \ldots, x_n \rangle$ represents the length (nt) of each elements (cf. Fig. 4.2).

Type	Junct.	Hairpin	Helix	Bulge	Internal	OBS
PASS_1	2	$2\langle 3, 15\rangle$	$2\langle 3, 7\rangle$	-	-	$1\langle 15\rangle$
PASS_2	2	$2\langle 4, 15\rangle$	$2\langle 5, 6\rangle$	-	-	$1\langle 15\rangle$
PASS_3	3	$2\langle 4, 7\rangle$	$3\langle 5, 8, 16\rangle$	$2\langle 1, 1\rangle$	$1\langle 3\rangle$	$1\langle 22\rangle$
PASS_4	3	$2\langle 4, 7\rangle$	$3\langle 5, 8, 16\rangle$	$2\langle 1, 1\rangle$	$1\langle 4\rangle$	$1\langle 22\rangle$
NOT_1	1	$1\langle 15\rangle$	$1\langle 5\rangle$	-	-	$1\langle 15\rangle$
NOT_2	3	$2\langle 6, 6\rangle$	$3\langle 4, 10, 13\rangle$	-	$3\langle 2, 4, 8\rangle$	$1\langle 22\rangle$
AND_1	3	$3\langle 3, 15, 15\rangle$	$3\langle 3, 8, 9\rangle$	$1\langle 15\rangle$	-	$2\langle 15, 15\rangle$
AND_2	3	$3\langle 4, 15, 15\rangle$	$3\langle 5, 6, 6\rangle$	-	-	$2\langle 15, 15\rangle$
AND_3	2	$2\langle 15, 15\rangle$	$2\langle 8, 9\rangle$	-	-	$2\langle 15, 15\rangle$
AND_4	3	$2\langle 4, 7\rangle$	$3\langle 5, 8, 21\rangle$	$1\langle 1\rangle$	$3\langle 2, 5, 10\rangle$	$2\langle 16, 16\rangle$
OR	3	$2\langle 4, 7\rangle$	$3\langle 5, 8, 23\rangle$	$2\langle 1, 1\rangle$	$3\langle 2, 4, 6\rangle$	$2\langle 20, 20\rangle$
$a \wedge \neg b$	2	$2\langle 15, 15\rangle$	$2\langle 5, 7\rangle$	-	-	$2\langle 15, 15\rangle$
$a \wedge b \wedge \neg c$	3	$3\langle 15, 15, 15\rangle$	$3\langle 5, 6, 6\rangle$	-	-	$3\langle 15, 15, 15\rangle$
Hammerhead[a]	3	$2\langle 4, 4\rangle$[b]	$3\langle 5^{c}4^{d}, 7^{e}\rangle$	-	-	-
Hairpin	2	$1\langle 4\rangle$	$4\langle 10, 4, 5, 3\rangle$[f]	-	$2\langle 8, 6\rangle$	

[a] *In-cis* has an extra hairpin loop in either helices I or III while *in-trans* would consists of one hairpin at helix II.

[b] Tetra-loop hairpin at the end of stem (Tanner, 1999; Hertel et al., 1994, 1996; Usman et al., 1996).

[c] Minimum number of stem loops is 1 and the optimum length varies between 5 and 8, after (Tuschl et al., 1995; Tanner, 1999; Hendry et al., 2005; Birikh et al., 1997; Usman et al., 1996; Amarzguioui and Prydz, 1998; Hertel et al., 1996, 1994). A longer helical arms of 30 nt for both helices I and III was also reported (Lieber and Strauss, 1995).

[d] The optimum length of stem II as reported in (Tanner, 1999; Tuschl et al., 1995; Usman et al., 1996; Amarzguioui and Prydz, 1998; Birikh et al., 1997) with minimum of 2 nt to ensure cleavage reactions.

[e] Minimal length of stem III for optimum efficiency (Hertel et al., 1996, 1994; Tuschl et al., 1995; Birikh et al., 1997; Hendry and McCall, 1996; Tanner, 1999; Hendry et al., 2005) with a maximum of 8 nt. Amarzguioui and Prydz (1998) as well as Tanner (1999) point out the importance of stem III for specificity and that normally stem III is longer then stem I.

[f] Length of helices according to (Tanner, 1999; Puerta-Fernández et al., 2003; Fedor, 2000; Porschke et al., 1999). The size of stem I and IV can be extended up to 27 and 25 nucleotides respectively (Porschke et al., 1999).

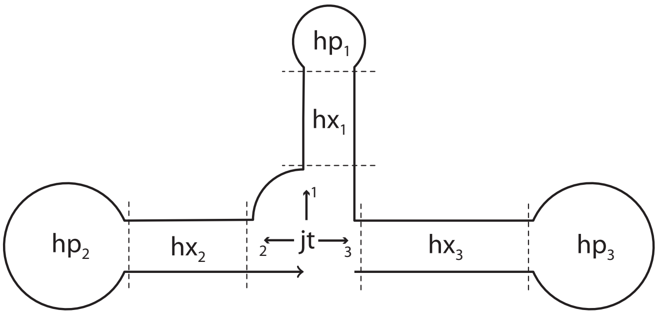


FIGURE 4.2: Decomposition into structural elements. Molecular AND gate (Stojanovic et al., 2002) composed of Hairpin (hp) = $\langle hp_1, hp_2, hp_3 \rangle$, Helix (hx) = $\langle hx_1, hx_2, hx_3 \rangle$ and Junction (jt) = 3 elements.

In this study, we used two test sets to evaluate the performance of 5 RNA sequence design algorithms, i.e., *RNAinverse*, *RNAdesigner*, *INFO-RNA*, *StochSrch*, and *RepInit*. First, we generate arbitrary structures (denoted as "artificial") using the parameterisation of Tab. 4.1, and secondly we select from the literature 31 nucleic acid computing units that have been verified in the laboratory (denoted as "engineered") as our benchmark tests. These two cases incorporate structures with and without sequence constraint. For the unconstrained setting, only the conformation of the molecule is supplied to the algorithm. Any combination of bases are allowed in the sequence. For the constraint setting, in addition to the conformation, we assign certain base positions in the sequence to have fix bases to represent the conserved regions of the molecule, as illustrated by the hammerhead ribozyme (cf. Fig. 2.4) and the hairpin ribozyme (cf. Fig. 2.5). This can be further extended to include conserved bases in substrate strands and at binding site regions of a computational unit, which is essential in constructing networks of computational nucleic acids.

The following tools were selected for the experiment; *RNAinverse* distributed in the *Vienna RNA package* (Hofacker, 2007), the latest binaries and C codes of *RNAdesigner* obtained kindly from Andronescu et al. (2004) and the latest binary for *INFO-RNA* obtained kindly from Busch and Backofen (2006). Default parameter settings were kept for each tools in the experiment with the exception of *StochSrch*. In the process of developing *RepInit*, we managed to construct a set of parameters setting that is compatible with the structure space defined here. These parameters were originally adapted from *RNAdesigner*, but underwent a number of preliminary tuning routines to ensure that with proper parameter biases, the initialisation procedure can solely generate good sequence candidates, without the need of the optimisation heuristic. The same set of parameters was later adapted for *StochSrch*. The parameter setting for both *StochSrch* and *RepInit* is given in Tab. 4.3.

Each algorithm processes five data sets of "artificial" structures. Each data set consists of 500 structures without sequence constraints (D1-U, D2-U, D3-U, D4-U, and D5-U) and another 500 structures with the same conformations as their unconstrained sets but

Table 4.3: Parameters setting for *StochSrch* and *RepInit* algorithms. As *StochSrch* and *RepInit* are derived from *RNAdesigner*, we adopt the parameter names of the original implementation. Parameters not listed remain unchanged from (Andronescu et al., 2004).

Parameter Name	Parameter Description	Parameter Value
MAX	No. of iterations for *RepInit*	1000
pb_paired	Probability of paired bases	$P_G = 0.55$, $P_C = 0.30$, $P_A = 0.10$, $P_U = 0.05$
pb_unpaired	Probability of bases being unpaired	$P_A = 0.80$, $P_U = 0.10$, $P_G = 0.06$, $P_C = 0.04$
nl	Repetitions for refinement in *StochSrch*	10000
reset	Iterations before re-initialisation in *StochSrch*	1000
pb_prand	Probability of selecting non-conflicting bases in the refinement process	0.2
pb_acc	Probability of selecting a poor distance value in refinement process	0.001[a]

[a] The selection of sequence with poor distance value is 1 in every 1000 iterations. We assign a small pb_acc value because the probability biases of base pair and unpaired base have been tuned to suit the design space prior to the comparison study

with added sequence constraints (D1-C, D2-C, D3-C, D4-C, and D5-C). The generation of these “artificial” structures follows a computational procedure developed from the algorithm described in (Andronescu et al., 2004). For the constrained sets, we fixed short stretches of base positions (≈10% of the structure length) in the molecule with arbitrary bases. The number of stretches with fixed bases in a structure is randomised between 1 and 5 depending on the structure length. For instance, given a molecule with the size of 150 nt, we could have two (of length 10 nt and 5 nt) conserved regions with arbitrary assigned bases. In addition to the data sets of “artificial” structures, we include a data set of “engineered” structures which comprises 31 unconstrained structures (DE-U) and 31 structures with fixed bases in the conserved regions (DE-C). Because all of the sequence designer are randomised algorithms, 20 runs were performed for both the unconstrained and constrained data sets.

As discussed at the beginning of this chapter, the accuracy of the sequence design algorithm is measured by the distance ($\mathcal{D}$) between the target structure and the folded structure of the candidate sequence . A candidate sequence will fold to the target structure, if $\mathcal{D} = 0$. Small distance values indicate a good sequence design. Sequences with small distance values are normally within a few kinetic moves of the target structure and can be perceived as stuck in a local optimisation minimum. Additional steps of base mutations on these sequences would likely result in the target structure, indicating premature convergence of the algorithm.

Table 4.4 summarises the performance results for the five algorithms discussed so far. Performance is measured as the mean number of structures across all 20 runs, where the generated sequences fold (according to *RNAfold*) exactly to the target structures ($\mathcal{D} = 0$). *RepInit*, *StochSrch* and *RNAdesigner* performed relatively well with a success rate of 75%, if perfect accuracy ($\mathcal{D} = 0$) is demanded. *RNAdesigner* performs worse than *RepInit* and *StochSrch*. Despite the omission of the usual gradual refinement (heuristic optimisation) from *RepInit*, the algorithm performs surprisingly well on the unconstrained data set. A possible explanation for this observation can be found in the fact that for any given structure, typically a large number of sequences can be found that will fold into this structure. The set of sequences supporting a given structure grows exponentially with sequence length because base pairs can be substituted with pairing bases and unpaired bases can be substituted with other non-binding bases (Hofacker, 1994; Higgs, 1993).

Table 4.4: Performance of RNA sequence design algorithms against the six unconstrained datasets. Mean number of structures with $\mathcal{D} = 0$ obtained across all 20 runs. D1-U to D5-U consists of 500 structures and for DE-U 31 structures.

Dataset	RepInit	StochSrch	RNAdesigner	RNAinverse	INFO-RNA
D1-U	393.0±0.00	393.0±0.00	356.8±0.61	135.3±2.10	0.0±0.00
D2-U	387.2±0.12	389.0±0.00	355.4±0.58	129.6±2.00	0.0±0.00
D3-U	392.3±0.13	393.0±0.00	378.6±0.55	144.3±1.68	0.0±0.00
D4-U	394.9±0.16	396.0±0.00	373.4±0.62	136.5±1.71	0.0±0.00
D5-U	382.4±0.18	384.0±0.00	353.2±0.64	134.0±1.60	0.0±0.00
DE-U	28.6±0.11	29.1±0.08	26.7±0.11	15.4±0.65	0.0±0.00

Table 4.5: Performance of RNA sequence design algorithms against the six constraint datasets. Mean number of structures with $\mathcal{D} = 0$ obtained across all 20 runs. D1-C to D5-C consists of 500 structures and for DE-C 31 structures.

Dataset	RepInit	StochSrch	RNAdesigner	RNAinverse	INFO-RNA
D1-C	300.4±0.52	393.0±0.00	291.0±0.39	128.8±1.68	0.0±0.00
D2-C	283.3±0.38	389.0±0.00	283.2±0.34	125.6±1.14	0.0±0.00
D3-C	301.5±0.43	393.0±0.00	304.8±0.50	140.4±2.08	0.0±0.00
D4-C	279.8±0.40	396.0±0.00	280.2±0.30	127.9±1.53	0.0±0.00
D5-C	281.8±0.34	384.0±0.00	275.1±0.31	127.9±1.56	0.0±0.00
DE-C	22.8±0.09	29.1±0.08	20.0±0.13	15.4±0.71	0.0±0.00

However, for datasets with sequence constraints, only *StochSrch* showed similar performance as the unconstrained data set (cf. Tab 4.5). Upon closer inspection, we found only two structures which previously in the unconstrained setting had sequences folding exactly into the target structure ($\mathcal{D} = 0$) that did not fold accurately with constraints ($\mathcal{D} = 4$ for both structures). For the remaining structures (i.e., with sequences that fold exactly into the target structure ($\mathcal{D} = 0$) in the unconstrained setting), sequences that correctly fold to the target structure are generated. This shows the ability of the algorithm to work around the ≈10% conserved bases constrained assign to the data sets. There is a drop in success rate for both *RepInit* and *RNAdesigner* for the constraint datasets. For constrained sequences, a significant different (of ≈100 structures) between *StochSrch* and both *RepInit* and *RNAdesigner* is observed. We anticipated the drop in performance for *RepInit* since in its current form, it lacks the rules that can handle fixed base assignments. However, it still performs comparably well to *RNAdesigner*, despite having no optimisation heuristic. The difference between *StochSrch* and *RNAdesigner*, even though the former is a simplified version of the latter, could be due to the parameter settings employed in *StochSrch*. Initially we thought that the parameter setting in *RNAdesigner* was to generic and intended to handle a broader structure space. To investigate this hypothesis we swapped the parameters between *StochSrch* and *RNAdesigner*, we are able to retained the accuracy of *StochSrch*, while there is a slight drop in performance recorded for *RNAdesigner*. This ruled out that the performance difference is caused by parameter settings.

For the unconstrained as well as the constraint data sets, approximately 30% of the structures folded from sequences generated by *RNAinverse* match the target structures. Compared to the design space investigated by Andronescu et al. (2004), the drop of performance for *RNAinverse* in our restricted design space is significant. The results produced by *INFO-RNA* fairs even worse. From the six unconstrained datasets and six constraint datasets, *INFO-RNA* did not generate even a single sequence that fold correctly to the target structure in all 20 runs. The poor performance of *INFO-RNA* might be caused by the initialisation of a candidate sequence that is predicted to fold into a structure with low free energy given the base pairing and unpaired bases requirements of the target structure. It is possible that the initialisation sequence is already trapped in a global minimum, and the number of mutations required to change the combination of bases exceed the iteration limit, thus preventing the algorithm from arriving at the solution. In term of accuracy (i.e., sequences that fold exactly to the target structure), it seems that the probabilistic biased algorithms (*StochSrch*, *RepInit* and *RNAdesigner*) perform significantly better than *RNAinverse* and *INFO-RNA*, with *StochSrch* having an advantage over the other *RepInit* and *StochSrch* specifically for constrained sequences.

Figure 4.3 shows the performance (without discarding sequences with mismatches, $\mathcal{D} \geq 0$) of *RNAdesigner* (A), *StochSrch* (B), *RepInit* (C), *RNAinverse* (D), and *INFO-RNA* (E) for the unconstrained data set across 20 runs. *StochSrch*, *RepInit* and *RNAdesigner* managed to produced sequences with smaller distance values $\mathcal{D}$, compared to *RNAinverse* and *INFO-RNA*. A more detailed analysis revealed that, the distance value for any mismatch sequence designed is not more than 4 nt ($\mathcal{D} \leq 4$) for *StochSrch*, *RepInit*, and *RNAdesigner*. As seen in Fig. 4.3A, B and C, the accuracy of the sequences produced by *StochSrch*, *RepInit* and *RNAdesigner* remained constant across the structure length (i.e., length of a sequence representing a structure) range. In contrast to *INFO-RNA*,

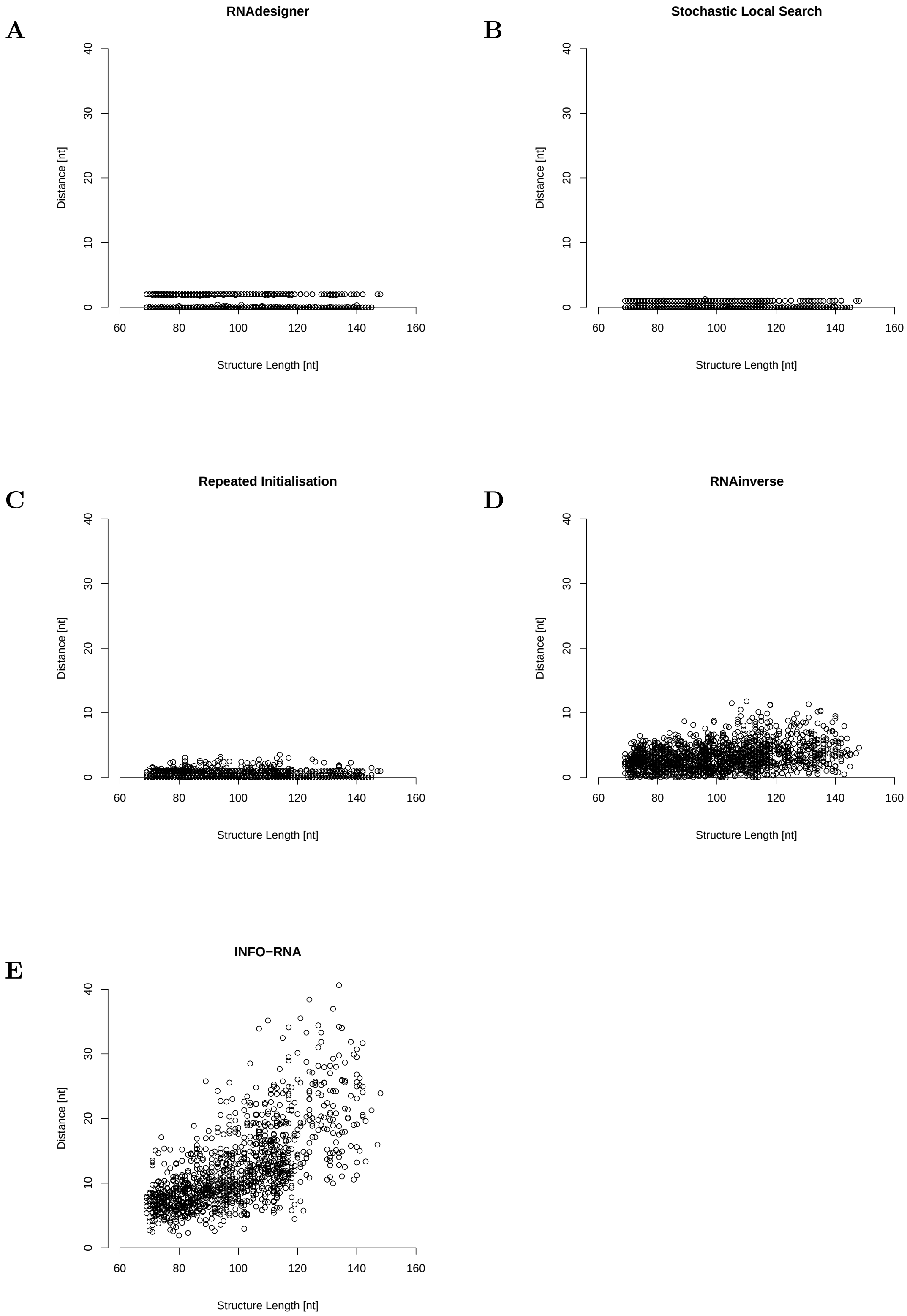


Figure 4.3: The performance of *RNAdesigner* (A), *StochSrch* (B), *RepInit* (C), *RNAinverse* (D), and *INFO-RNA* (E) against the unconstrained datasets. *RNAdesigner*, *StochSrch*, and *RepInit* show a low distance ($\mathcal{D} \leq 4$ nt) that is consistent for all structures regardless of length. The distance for *RNAinverse* is generally higher but largely below 10 nt. However, *INFO-RNA* lacks accuracy in its overall performance and it is directly influenced the by structure length.

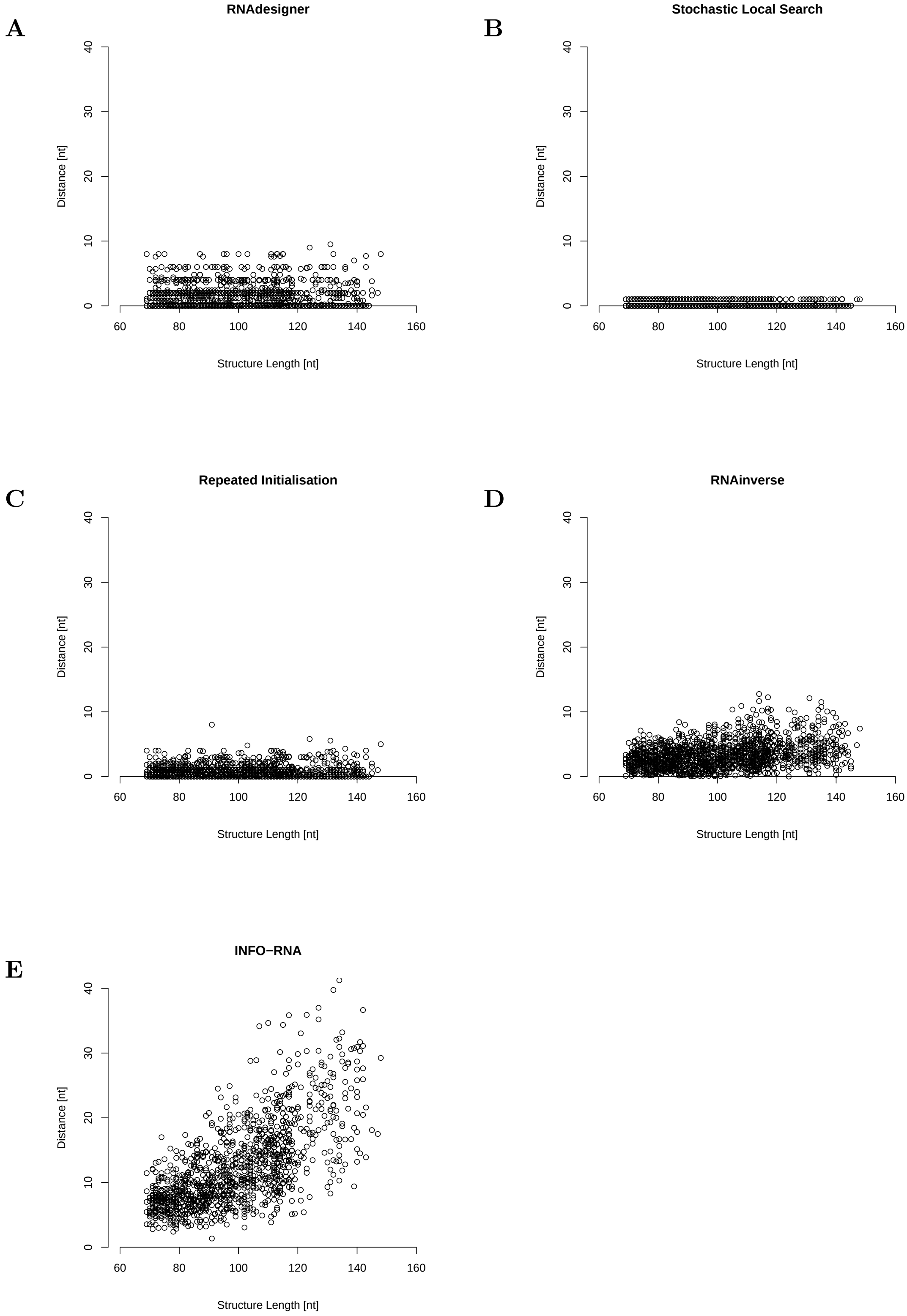


FIGURE 4.4: The performance of *RNAdesigner* (A), *StochSrch* (B), *RepInit* (C), *RNAinverse* (D) and *INFO-RNA* (E) against the constraint datasets. The three algorithms with probability biased initialisation (*RNAdesigner*, *StochSrch* and *RepInit*) outperformed *RNAinverse* and *INFO-RNA*.

RNAinverse managed to keep the distance value lower than 15 nt. The distance increases slowly with the increase of structure length. The results for *INFO-RNA* show that the accuracy of the algorithm, drops quickly with increasing structure length (Fig. 4.3E).

For constraint sequences, the overall behaviour of the algorithm shows a similar picture. Somewhat surprisingly, for *StochSrch* the addition of constraints did not reduce the accuracy of the generated sequences (Fig. 4.4B). Where mismatches occur, the distance remain smaller than 4 nt ($\mathcal{D} < 4$). However, with growing structure length there is a steady decline in accuracy observed for *RepInit* and *RNAdesigner*. The results of the latter show a larger variance that those of the former, even though *RepInit* is a simplified version of *RNAdesigner*. Although there is a significant decrease in accuracy (cf. Tab. 4.5), the overall performance of *RepInit* is still comparable to *StochSrch*, with the majority of sequences across the length range having a small or zero distance to the target. But compared to the unconstrained data set, *RepInit*, unlike *StochSrch*, shows a slight decrease in accuracy. The performance of *RNAinverse* and INFO-RNA are similar for constraint and unconstrained sequences. The mean number of sequences where the folded structure match the target structure ($\mathcal{D} = 0$) for *RNAinverse* in the constraint dataset resembles the number generated in the unconstrained dataset.

At this point, it is safe to say that both *RNAinverse* and *INFO-RNA* failed to produce result comparable to the trio of *StochSrch*, *RepInit* and *RNAdesigner*. Therefore, the two algorithms are excluded from the minimum free energy analysis. Minimum free energy (MFE) is an important factor in accessing the stability of a molecule. As described in Section 2.3, the framework of these computational units rely on possessing multiple stable conformations. If for instance, a molecule is designed to change conformation from one state to another state, then one must construct the molecule, to have two stable states with an energy barrier sufficient to separate the two states and at the same time susceptible to kinetic transformation for conformation switching. The MFEs of the generated sequences is therefore as another criterion in evaluating the performance of the sequence design algorithms.

For the detailed analysis, only sequences that fold correctly to the target structure ($\mathcal{D} = 0$) were selected. Mismatch structures ($\mathcal{D} > 0$) are assumed to be trap in local minimum and they are discarded from the analysis. Except for *RepInit*, both *StochSrch* and *RNAdesigner* use optimisation heuristic in generating their sequences. A candidate sequence is iteratively mutated to minimise its distance to the target structure. However, once a candidate sequence is at a local minimum, it is possible that the number of iterations will eventually reached the maximum limit set for the algorithm. Table 4.6 comprises only structures of the unconstrained sequence set for which *RepInit*, *StochSrch* and *RNAdesigner* generate an accurate ($\mathcal{D} = 0$) sequence. For each algorithm the number of structures with the lowest free energy sequences among the sequence generated by the three algorithms is listed.

Table 4.6: The number of structure with the lowest free energy among sequences with $\mathcal{D} = 0$ generated by *RepInit*, *StochSrch*, and *RNAdesigner* on the unconstrained datasets.

Dataset	RepInit		StochSrch		RNAdesigner	
	Seq. Count	%	Seq. Count	%	Seq. Count	%
D1-U	354±0.14	99.4	2±0.66	0.6	0±0.00	0.0
D2-U	351±0.13	98.8	4±0.75	1.2	0±0.00	0.0
D3-U	374±0.15	99.2	3±1.21	0.8	0±0.00	0.0
D4-U	371±0.17	99.4	2±0.35	0.6	0±0.00	0.0
D5-U	352±0.19	99.7	0±0.00	0.0	1±0.05	0.3
DE-U	26±0.12	100.0	0±0.00	0.0	0±0.00	0.0

In the unconstrained setting, *RepInit* outperformed both *StochSrch* and *RNAdesigner*, delivering for ≈99% of the structures, the sequence that fold exactly to target structure with the lowest free energy among the three algorithms. There are two observations that can be made from this finding. Firstly, the probability biases and the rules (cf. pg. 5) of *RepInit* contributed significantly to the outcome. To test whether the advantage of *RepInit* can be attributed to the invocation of rules or the probability biases, we rerun the *RepInit* algorithm using the default probability biases of *RNAdesigner*. Only a slight reduction of performance was observed, indicating that the rules alone are sufficient to generate sequences that fold into secondary structures with lower free energies then the algorithm using optimisation methods. Secondly, if one compares only *StochSrch* and *RNAdesigner*, ≈97% of the sequences that fold correctly to the target structures with the lowest free energy are generated by *StochSrch*. When we substituted the probability biases of *StochSrch* with the default value in *RNAdesigner* (denoted as *StochSrch**, the overall performance dropped to ≈90%, which indicates the simplified variants of the algorithm, *StochSrch*, is sufficient to cater for the restricted structure space, and the additional procedures present in *RNAdesigner*, i.e., tabu search mechanism and hierarchical decomposition optimisation are a mere distraction in our application scenario.

We plot the difference between the lowest free energy of the folded sequences generated by *StochSrch* and the lowest free energy of the folded sequences generated by *RepInit* for the 50 randomly selected samples in Fig. 4.5. The two algorithms that together generate the sequences that fold with the lowest free energy for almost every structures in the datasets are *StochSrch* and *RepInit*. For the overwhelming majority of the structures, *RepInit* yields the sequence that folds with the lowest free energy. While *RepInit* is considerably better, this leads to the question of how much of a quantitative advantage *RepInit* provides. It turns out, as seen in Fig. 4.5 that the energy difference between sequences generated by *RepInit* and *StochSrch* is typically small. We then compare the difference between sequences generated by *RepInit* and *RNAdesigner*. A more significant gap can be depicted in Fig 4.5. The sampling of structures across the six datasets show a consistent gap distance exists between *RNAdesigner* and *RepInit*.

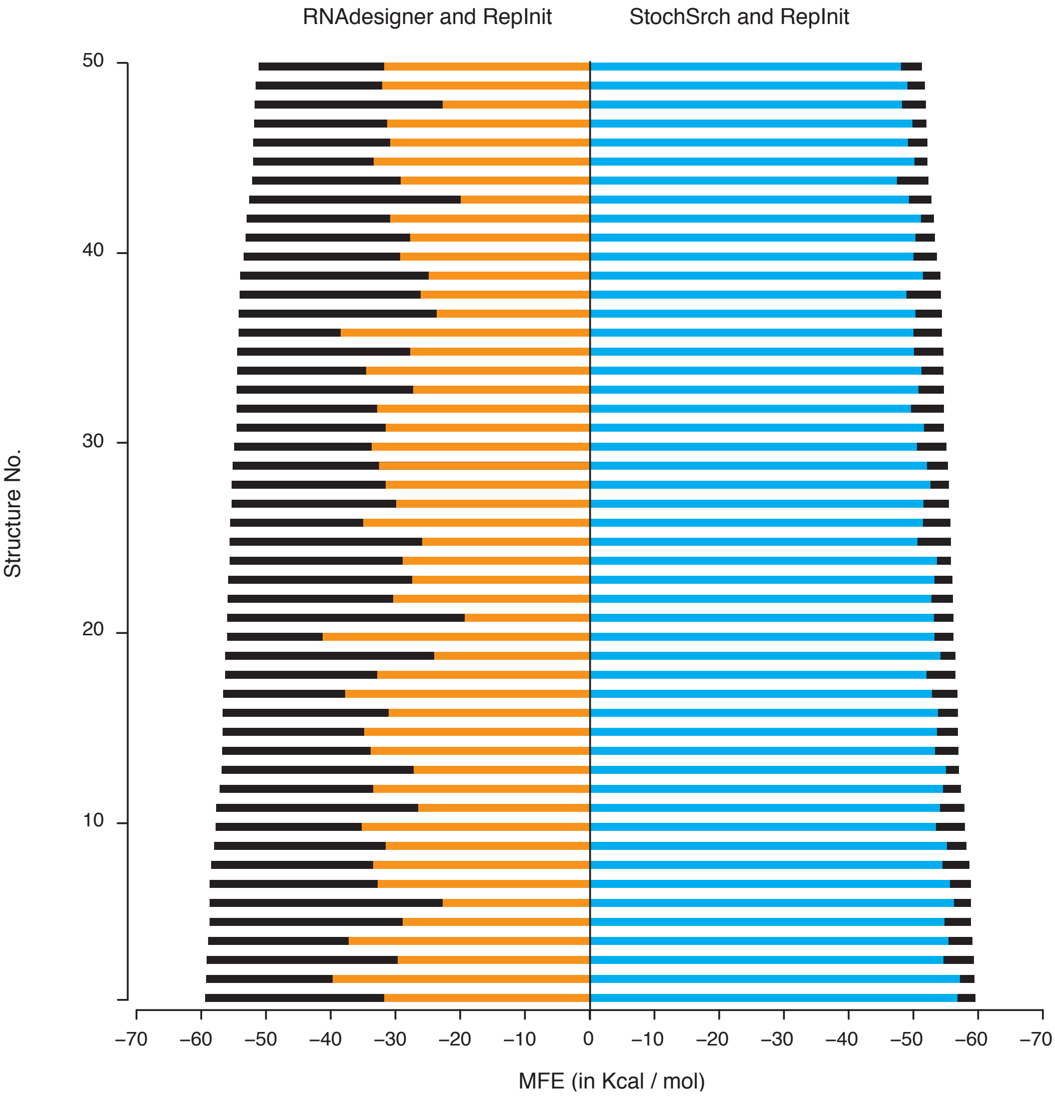


FIGURE 4.5: Minimum free energy for sequences generated by *RepInit* (in black), *RNAdesigner* (in orange), and *StochSrch* (in blue). Note the difference of free energy between *RepInit* and *StochSrch*, as well as *RepInit* and *RNAdesigner*. The graph shows data for 50 structures randomly selected from the 354 structures in the unconstraint dataset D1-U for which all algorithms delivered accurate sequences ($\mathcal{D} = 0$).

Next, we inspect the performance of the three sequence design algorithms for constrained sequences, shown in Tab. 4.7. Only sequences that fold exactly to the target structures are considered and among these sequences generated by *RepInit*, *StochSrch*, and *RNAdesigner*, for each structure, sequence that fold to the lowest free energy structure is selected. *RepInit* again dominates, generating sequences that fold exactly to target structures with lowest free energy. For *StochSrch* the free energy of the folded sequence are comparable to the lowest free energy of the folded sequence generated by *RepInit*, similar to the one depicted earlier in Fig. 4.5. Despite the inability of *RepInit* to cope with the sequence constraints (indicated by the reduced mean accuracy), it still produced ≈98% of the lowest free energy structure within the pool of the accurately designed se-

Table 4.7: The number of structure with the lowest free energy among from *RepInit*, *StochSrch*, and *RNAdesigner* where sequences with $\mathcal{D} = 0$ were obtained by all three algorithm on the constraint datasets.

Dataset	RepInit		StochSrch		RNAdesigner	
	Seq. Count	%	Seq. Count	%	Seq. Count	%
D1-C	287±0.45	98.6	4±0.21	1.4	0±0.00	0.0
D2-C	273±0.41	96.4	10±1.20	3.6	0±0.00	0.0
D3-C	295±0.37	97.3	8±0.32	2.7	0±0.00	0.0
D4-C	293±0.21	97.6	7±0.11	2.4	0±0.00	0.0
D5-C	269±0.10	97.8	6±0.29	2.2	0±0.00	0.0
DE-C	19±0.01	95.0	1±0.05	5.0	0±0.00	0.0

quences ($\mathcal{D} = 0$). The combination of both rules and probability biased proved to be sufficient in designing a highly stable RNA molecule. *RepInit* maintained its ability to generate sequences with the lowest free energy, while the drop in accuracy affected its overall performance in the sequence constraint setting.

In contrast to *RepInit*, *StochSrch* not just maintained its accuracy, but at the same time generates sequences with free energy comparable to *RepInit*, which indicates that *StochSrch* is the algorithm most suited to the design of RNA molecules with fixed bases. In order to understand the contribution of the probability biases for base pairs and unpaired bases, we again use the default probability bias of *RNAdesigner* in *StochSrch* (*StochSrch**). If we compared only *StochSrch* and *RNAdesigner*, the number of sequences that folds exactly to the target structure with the lowest free energy is ≈97%. Using *StochSrch** on the constraint sequences, we observed a slight drop of performance (≈93%) compared to the ≈97% produced using the probability biases of *StochSrch*. When we plot the free energy difference between *StochSrch* and *StochSrch** from the 50 randomly selected structures, the two values remain relatively close together (Fig. 4.6). This indicates that probability bias alone is insufficient in generating sequences that fold to low free energy structure. Similar observations hold for the constraint datasets.

The findings that we gathered in the unconstrained and constraint datasets demonstrate the ability of the *StochSrch* and *RepInit* with regards to the restricted structure space, compared to its ancestor, *RNAdesigner*. We anticipate that if the number of fixed bases (the constraints) increases (i.e., currently only ≈10% of the sequence is assigned with conserved bases), the accuracy of *RepInit* to decline further, if it stays in the current form. However, we expect, as discussed earlier, additional rules to cater specifically for constraint sequences can resolve this issue. *StochSrch* is a viable alternative which can tackle both unconstrained and constrained structures rather well and yields sequences that fold with free energy comparable to *RepInit*. Before, we conclude our comparison we study the speed of each algorithm.

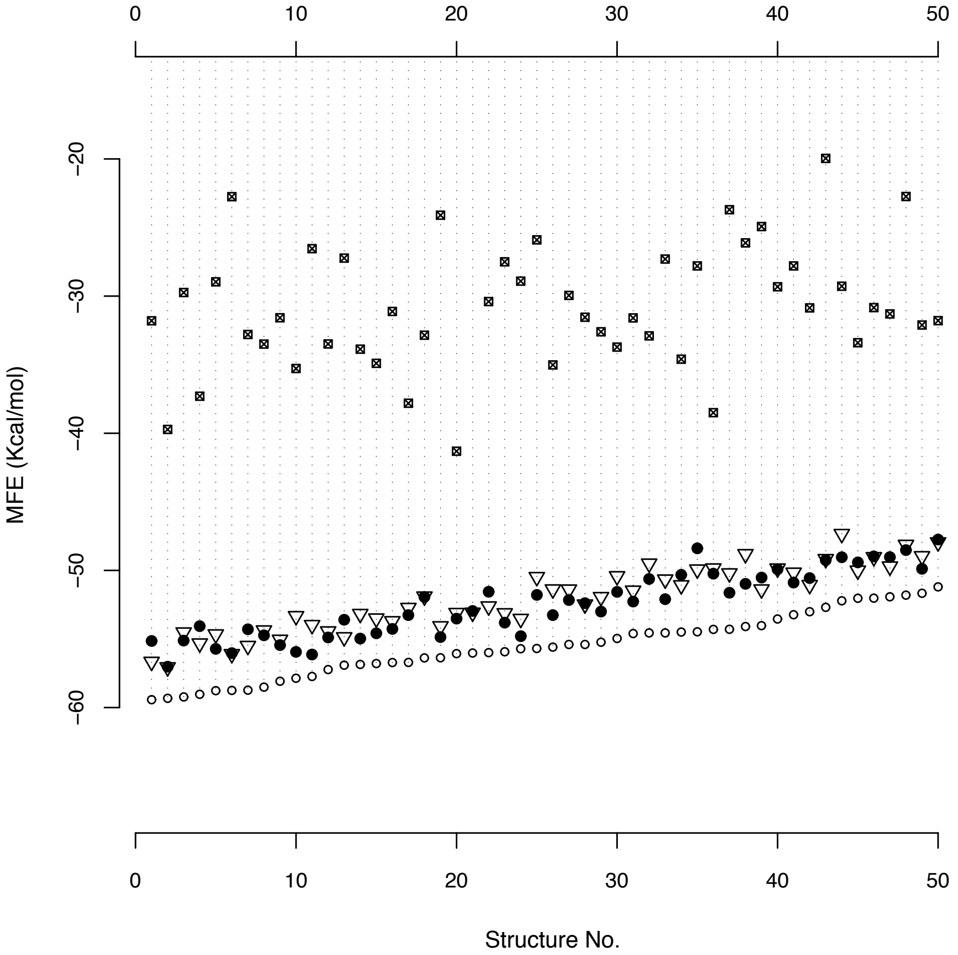


FIGURE 4.6: Number of times the lowest free energy sequence was obtained. A comparison of *RepInit*, *StochSrch*, *StochSrch**, and *RNAdesigner* applied to 50 structures randomly sampled from the subset of dataset 1 (D1-C) for which all sequence design algorithms achieved $\mathcal{D} = 0$. The *StochSrch** algorithm is similar to the *StochSrch* algorithm with the default probability biases of *RNAdesigner*. Each line represents a structure, and the different points represent the free energy of a sequence generated by ○ - *RepInit*, ● - *StochSrch*, ▽ - *StochSrch**, and ☒ - RNAdesigner.

Thus far, the quality of sequences generated by the design algorithms have been evaluated in term of their accuracy (distance between the structure folded from the generated sequences and the target structures) and their free energy. For the datasets (unconstrained and constraint) used in the study, *RepInit* and *StochSrch* have performed better than the other sequence design algorithms. Both algorithms (*RepInit* and *StochSrch* are the simplified versions of *RNAdesigner*. Similar to the other sequence design algorithms, numerous calls to the folding algorithm are required to evaluate the cost function ($\mathcal{D}$) of the candidate sequence against the target structure at each iteration. The time complexity of the folding algorithm is O(n^3), with sequence length n. Therefore the majority of the algorithms (*RNAinverse* and *RNAdesigner*) applied a structural decomposition procedure in order to reduce the processing time. However, because our design space only considers structures that are less than 200 nt, we omitted the decomposition and in the case of *RepInit* introduced a set of rules that refines the candidate sequence during the initialisation stage.

Following Andronescu et al. (2004), we conducted an empirical study of the computational time for *RepInit*, *StochSrch*, and *RNAdesigner* by measuring the mean CPU times from multiple runs on all structures where sequences that fold exactly to the target structures were obtained by all three design algorithms. Andronescu et al. (2004) assumes that the CPU time of *RNAdesigner* increases as the number of branches for the decomposition tree increases. Repetitive calls for the folding algorithm are still made to evaluate the cost function, but the processing time is reduced since each folding is required only for shorter sequences (substructures of the molecule). The CPU time for designing natural occurring RNAs of ≈1000 nt is well within the capacity of the current computers (Andronescu et al., 2004; Tulpan et al., 2005). The time complexity study of the the new algorithms (*RepInit* and *StochSrch*) is excluded because the length of structure in the design space of interest is limited to 200 nt. Therefore, we focus on the empirical study of the processing time, in order to get a direct comparison with the performance reported by Andronescu et al. (2004).

Table 4.8: Number of sequences for which the lowest processing time was achieved among *RepInit*, *StochSrch* and *RNAdesigner*. The sequences are those for which all the algorithms achieved $\mathcal{D} = 0$ in both unconstrained and constraint settings.

Dataset	RepInit		StochSrch		RNAdesigner	
	Seq. Count	%	Seq. Count	%	Seq. Count	%
D1-U	319±0.32	89.6	21±1.20	5.8	16±1.07	4.6
D2-U	313±0.27	88.1	15±0.78	4.2	27±2.23	7.7
D3-U	344±0.45	91.2	7±0.05	1.8	26±0.56	7.0
D4-U	338±0.40	90.1	8±0.31	2.1	29±0.42	7.8
D5-U	309±0.19	87.5	10±0.11	2.8	34±0.62	9.7
DE-U	13±0.02	50.0	3±0.03	11.5	10±0.31	38.5
D1-C	245±0.55	84.1	11±0.77	3.7	35±1.32	12.2
D2-C	225±0.37	79.5	17±0.56	6.0	41±1.46	14.5
D3-C	247±1.21	82.0	15±1.10	4.9	39±1.30	13.1
D4-C	229±0.17	82.0	16±0.21	5.7	34±0.75	12.3
D5-C	229±0.20	83.2	16±0.36	5.8	30±1.28	11.0
DE-C	16±0.36	80.0	1±0.01	5.0	3±0.12	15.0

Table 4.8 shows the number of structures that are generated with the lowest mean CPU time for sequences that fold exactly to the target structures were obtained by *RepInit*, *StochSrch* and *RNAdesigner*. The experiments were conducted on machines with dual 3.8GHz Intel Pentium 4 processors, 1.21GB cache and 3.21GB RAM, running SUSE Linux version 10.1 with 750GB hard drive. Each algorithm is terminated after a fixed number of 5000 trials, and in order to ensure that the results are obtained

fairly, we exclude the generation of inaccurate sequences ($\mathcal{D} > 0$). Inaccurate sequences normally reached the maximum number of iterations during their refinement. Because the maximum number of iterations differs for each designer, if we include these inaccurate sequences then the processing speed of each designer will be dependent on the maximum number of iterations, which can be irrelevant for the comparison study.

We expect that *RepInit*, which is the least complex of the three algorithms would take the least time for most of the sequences, and, as seen from the table, this is indeed the case for ≈89% of the "artificial" unconstrained datasets and for ≈50% of the "engineered" unconstrained datasets. If we evaluate only *StochSrch* and *RNAdesigner*, ≈65% of the sequences were obtained the fastest using *RNAdesigner*. In our implementation of *StochSrch*, we require external folding program (*RNAfold*) calls to evaluate our cost function, in contrast to *RNAdesigner* that uses an internal folding algorithm. In later version of *StochSrch*, an internal call to our folding program has been implemented. Although *RepInit* uses the same external calls, the lack of refinement optimisation heuristic help in reducing the processing time.

As indicated in Tab. 4.8, for the unconstrained dataset, ≈80% of the sequences for which all algorithms achieved $\mathcal{D} = 0$ were obtained by *RepInit* with the least CPU time. This finding confirms the effect of removing the refinement optimisation heuristic in reducing the processing time. Although, the introduction of conserved bases did not affect the overall number of successful sequences which is obtained by *RepInit*, we have observed an increase in the processing time. Without specific rules to handle the pre-assigned bases, an increased number of iterations is required in order to obtain the correct sequences. Despite the increase in the number of iterations, *RepInit* is still capable of arriving at the sequence that folds to the target structure quicker than *StochSrch* and *RNAdesigner*. The effect of the external calls to the folding program is also evident in the constrained dataset. When we compare only *RNAdesigner* and *StochSrch*, ≈60% of the generated sequence were obtained by *RNAdesigner* quicker than *StochSrch*. As we have explained earlier, this issue has been resolved in later version of *StochSrch*, which also uses an internal folding algorithm. By comparing the CPU time taken by *RepInit*, *StochSrch* and *RNAdesigner*, we find that the improved performance in term of accuracy and quality shown in our analysis earlier, are not a result of increased CPU usage. *RepInit* in particular, took the least amount of time in generating the majority of the correct sequences, while the performance of *StochSrch* with the external folding program is still comparable to *RNAdesigner*.

To conclude, we found that the rule-based algorithm (*RepInit*) is sufficient to generate sequences that fold exactly to the target structures and have the lowest free energy among all RNA sequence design algorithms. The performance in terms of processing time (shown above) also indicates its efficiency as compared to optimisation based algorithms of *RNAdesigner* and *StochSrch*. However, we realised a need of more rules to handle the sequence constrained setting, in which stretches of the sequence are assigned fixed bases.

StochSrch performs the best in terms of accuracy in both unconstrained and constraint datasets. In terms of quality, the sequences generated by *StochSrch* are comparable with those generated by *RepInit*. The finding shows that the removal of the hierarchy decomposition procedure and the tabu mechanism did not affect the performance of *StochSrch* in our design space. We investigated the effect of removing the two procedures from *RNAdesigner* by substituting the default probability biases in *StochSrch* with the probability biases in *RNAdesigner*, and although the result indicates that there is a slight drop in performance, the overall performance in terms of accuracy and quality of the generated sequences is still comparable with the sequences generated by *StochSrch* with its default bias setting. Our findings imply that for *RepInit*, the exclusion of the optimisation heuristics improved both the quality of solutions and the processing time. Meanwhile, the removal of the two procedures from *RNAdesigner*, allow us to derived a more efficient sequence design algorithm (*StochSrch*) for our design space.

Chapter 5

Inverse Prediction of Interacting Multi-stable Nucleic Acids

5.1 Design of Multi-stable States RNA molecules

RNA molecules can possess multiple meta-stable conformations. An example from nature are riboswitches that directly bind to a specific metabolite, and harness the conformational changes in RNA to control gene expression (Mandal and Breaker, 2004; Nudler and Mironov, 2004; Nudler, 2006). At the current state of technology, RNA molecules up to about 60 nt long can be synthesised in vitro. From the perspective of designing computational nucleic acids, the design of sequence for RNA molecules with multi-stable conformations is important (cf. Sec. 2.3). The development of a molecular switch that is triggered to alter its function in response to the binding of a small molecule (Soukup and Breaker, 1999) demonstrates the possible application of this mechanism in computational nucleic acids.

The single-state sequence designers presented in chapter 4 generate RNA sequences based on a target structure. The occurrence of alternative conformations is explicitly avoided during the design process. Our objective is to find RNA sequences that possess two or more meta-stable conformations with low free energies separated by a certain energy barrier that facilitates or hinders switching among them. The conformational switching, in our approach is triggered by a short oligonucleotide. In the laboratory, the conformational switching can be achieved through the process of melting and rapid quenching of the molecule, as described for the multi-stable SV11 molecule (Biebricher and Luce, 1992; Zamora et al., 1995) (cf. Fig. 5.1). The structure (Fig. 5.1A) is converted to the structure (Fig. 5.1B) through the process of heating and rapid cooling. In order to convert back to the meta-stable structure a process of melting and slow cooling is conducted. Several others naturally occurring RNA switches (*Tetrahymena* group I intron (Pan and Woodson, 1998; Wu and Tinoco, 1998; Russell and Herschlag,

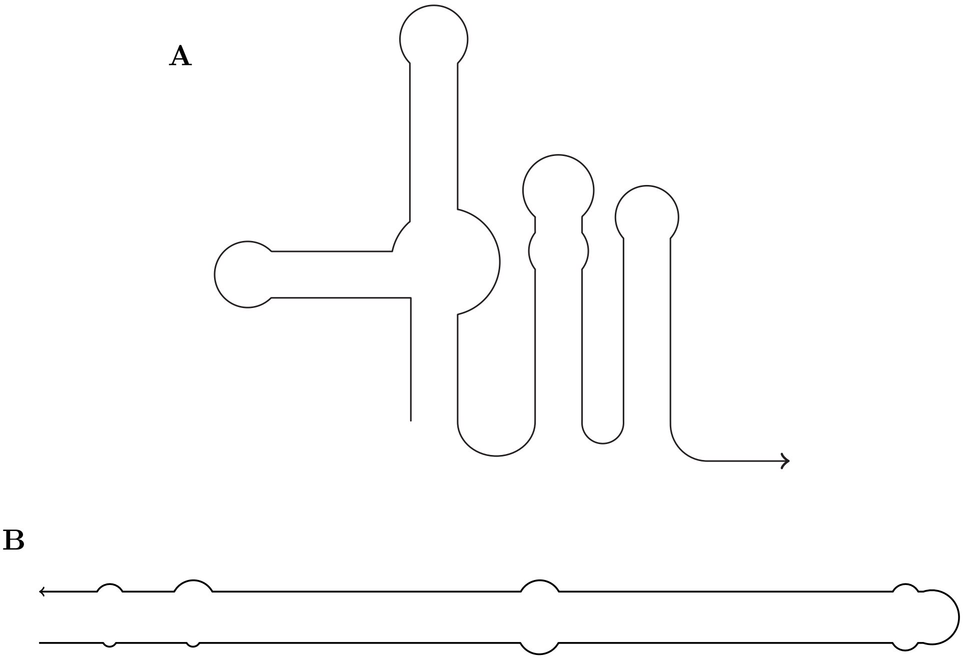


Figure 5.1: The secondary structure representation of SV11 molecule as depicted in (Biebricher and Luce, 1992). The molecule has predominantly two confirmations. An active metastable RNA (A), an in vitro selected template for Qβ replicase, and a rod-like stable conformation (B) that no longer functions as the substrate in the replication process.

2001), HDV ribozyme (Ferré-D'Amaré et al., 1998; Chadalavada et al., 2000) and the *hok/sok* system of plasmid R1 from *E.coli* (Gultyaev et al., 1997; Franch et al., 1997; Møller-Jensen et al., 2001)) follow the self-induced switches (Nagel and Pleij, 2002) that does not require trans-acting factors (e.g., introduction of oligonucleotide) to initiate the conformational change. During transcription, the self-induced RNA is kinetically trapped into its meta-stable conformation. Stretches of RNA (trigger region) that become unpaired after synthesis is completed, will then initiate refolding of the meta-stable conformation into the stable conformation (Nagel and Pleij, 2002).

At present, there is no designer for multi-stable nucleotide sequences that has been developed. There is, however, a software tool called *paRNAss* that has been developed to investigate the possibility of structural switching in RNA molecule (Geigerich et al., 1999). The program *paRNAss* uses clustering of suboptimal structures (based on structural similarity and energy barrier) to isolate the different conformational states that might be occurring in an RNA molecule. The clusters of suboptimal structures are differentiated by significant energy barriers. The *paRNAss* program provides a method of evaluating the accuracy of multi-stable generated sequences. One can implement a designer for multi-stable sequences by extending the design algorithms for single fold RNA described in chapter 4 and make use of *paRNAss* to evaluate the accuracy of the candidate sequence. However, the probability biased assignment of bases in the multi-stable case would not work, because the dependency relationship of each base position is unknown. For instance, given an RNA molecule with two meta-stable confirmations (state A and B), we must consider the base pairing requirement for both state A and state B. This is trivial for base pairs that are present in both states, but rather complicated for base pairs that are present in only one of the state, or when the base position pairs with two different base positions in each state.

A computational method to design sequences for RNA molecules with meta-stable conformations has been suggested by Flamm et al. (2001). In their approach, the optimisation procedure is localised to the "intersection region" of the meta-stable states. According to the intersection theorem by Reidys et al. (1997),

Intersection Theorem 5.1.1. If the nucleic acid alphabets admits at least one type of complementary base pairs, then, for any two secondary structures S_1 and S_2 there exists at least one sequence that is compatible with both structures, in symbols, $\mathcal{C}[S_1] \cap \mathcal{C}[S_2] \neq \varnothing$, where $\mathcal{C}[\Omega]$ denotes the set of all sequences that are compatible with structure Ω.

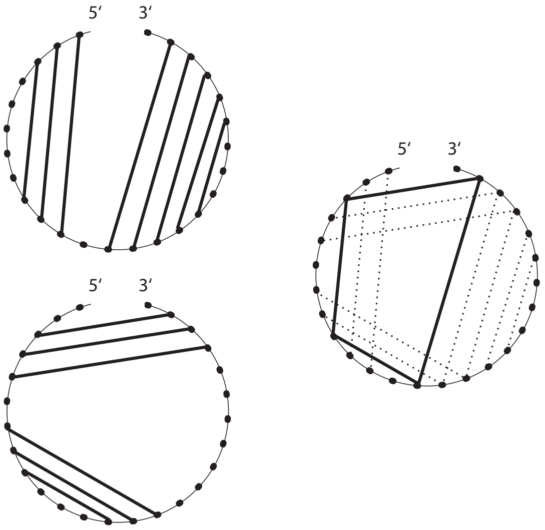


FIGURE 5.2: The dependency graph (Ψ) of two unique secondary structure conformations. Depicted on the left hand side, the two conformations of RNA secondary structure, in dot-bracket notation from 5'–3' as: `(((....))).(((((...))))).` and `.(((.(((...))).......))).` A bold line connecting two vertices form an edge that represents base pairing for the two positions. The union of the two conformations for the dependency graph is represented on the right hand side. Bold lines connecting four edges in the figure indicate the formation of a cycle, while the dotted lines represent paths (for vertex with more than one edge), single base pair (vertex with one edge–base pairs that can occur in only one conformation or present in both for the same vertices). The circular line connecting the set of vertices represents covalent bonds of the bases in a strand and does not contribute to the edges. Dependency graph was originally introduced originally by Flamm et al. (2001).

Consider an RNA molecule with two meta-stable conformations S_1 and S_2, if we construct a dependency graph (Ψ) (cf. Fig. 5.2), where the vertex set comprises sequence positions $\{1,\ldots,n\}$ and edges represent each base pair present in both S_1 and S_2, we can observe that each vertex can only be part of at most two edges. Any vertex with no edge represents a base position that remains unpaired in both conformations. Vertices with a single edge represent base positions that can either be a base pair in one of the conformation or form base pairs with the same base positions in both conformations. Vertices with two edges represent base positions that form base pairs in both conformations with two different base positions.

The combination of edges leads to the formation of paths (as indicated by the dotted lines in the dependency graph Fig. 5.2) or cycles (solid lines). A cycle is a path that starts and ends at the same base position. For instance, given a set of base pairs $B_1 = \{(a,b),(g,h),(k,q),(m,o)\}$ and $B_2 = \{(a,q),(b,k),(g,j)\}$, edges that contribute to the formation of a cycle are $(a,b),(b,k),(k,q)$ and back to (a,q), while the formation of path is given by (g,h) and (g,j). Paths also include the occurrences of "singles" or path with the length one, where only a single base pair exists (e.g., (m,o)). Cycles therefore must have an even number of edges and vertices. Given, B_1 and B_2 earlier, if a cycle $(a,b),(b,k),(k,q),(a,q)$ is present, then nodes a,b,k,q are likely to be a palindrome such as $a = \texttt{C}, b = \texttt{G}, k = \texttt{C}, q = \texttt{G}$, which forms a CGCG cycle. Paths can be assigned in a similar manner with the exception that the start and end base positions are not required to form a base pair (e.g.,CGUA).

For molecules with more than two meta-stable conformations, the following **Generalised Intersection Theorem** (Reidys et al., 1997; Schuster, 2006) applies,

Generalised Intersection Theorem 5.1.2. Given $\mathcal{B} = \{AU, UA, CG, GC, GU, UG\}$, $\mathcal{A} = \{A, G, C, U\}$, and S_x represents the secondary structure of x, let us assume that $\mathcal{B} \subseteq \mathcal{A} \times \mathcal{A}$ contains at least one symmetric pair, that is, $XY \in \mathcal{B}$ and $YX \in \mathcal{B}$, then

1. $\mathcal{C}[S_1] \cap \mathcal{C}[S_2] \cap \cdots \cap \mathcal{C}[S_n] \neq \varnothing$, if the dependency graph Ψ is bipartite

2. The number of sequences that are compatible with all structures can be written in the form
$$|\,\mathcal{C}[S_1, S_2, \ldots, S_n]\,| \quad = \prod_{\text{component } \psi \text{ of } \Psi} F(\psi) \tag{5.1}$$
where $F(\psi)$ is the number of sequences that are compatible with the connected component ψ

3. For the nucleotide bases $\bigcap_j \mathcal{C}[S_j] \neq \varnothing$ holds if and only if Ψ is a bipartite graph. In particular, for the case of bistable sequences, $n = 2$, the size of the intersection can be expressed explicitly in terms of Fibonacci numbers as follows,
$$\begin{aligned} F(P_n) &= 2\Big(\mathrm{Fib}(n) + \mathrm{Fib}(n+1)\Big), \\ F(C_n) &= 2\Big(\mathrm{Fib}(n-1) + \mathrm{Fib}(n+1)\Big), \end{aligned} \tag{5.2}$$
where P_n and C_k are the path and cycle components of Ψ with n vertices.

The easiest way to evaluate the bipartite property of a dependency graph Ψ is by calculating the number of edges belonging to the cycles. The formation of an odd cycle is not allowed because of the complementary base pair combination. As we have shown earlier, the first base position of a cycle must be a complementary pair of the last base position. For instance for a cycle of size 4, we can assign the following base configurations `CGCG`, with the following base pair formation, `C-G` for position 1 and 2, `G-C` for position 2 and 3, `C-G` for for position 3 and 4 and `G-C` for position 4 and 1. However, for odd cycles, the first and last base positions would not form base pair. If we assign bases `CGC` for a cycle containing three edges, the base pair `C-C` for position 3 and 1 would violates the complementary pairing rule.

Rather than minimising the distance value between the candidate sequences and the target structures, multi-stable sequence designers focus on minimising the energy difference between the meta-stable conformations. Flamm et al. (2001) suggest three different objective functions for the optimisation procedure to find an RNA sequence for an RNA molecule with multi-stable conformations The suggested objective functions $\Xi(x)$ are given as follows,

$$\Xi_A(x) = \Big(\sum_{i=1}^{n} E(x,S_i)\Big) - nG(x) \; + \; \xi \sum_{i=1,j=1}^{n} \Big(E(x,S_i) - E(x,S_j)\Big)^2 \tag{5.3}$$

$$\Xi_B(x) = \Big(\sum_{i=1}^{n} ET_i(x,S_i)\Big) - nG(x) + \xi \sum_{i=1,j=1}^{n} \Big(ET_i(x,S_i) - ET_j(x,S_j)\Big)^2 \tag{5.4}$$

$$\begin{aligned}\Xi_C(x) = \Big(\sum_{i=1}^{n} E(x,S_i)\Big) - nG(x) \; &+ \; \xi \sum_{i=1,j=1}^{n} \Big(E(x,S_i) - E(x,S_j)\Big)^2 \\ &+ \; \zeta \sum_{i=1,j=1}^{n} \Big(B(x,S_i,S_j) - \Delta E^2\Big)^2 \end{aligned} \tag{5.5}$$

where, S_i represents folded states of an RNA molecule, $G(x)$ be the ensemble free energy of sequence x, $E(x,S_i)$ denotes the free energy of conformation S_i, $ET_i(x,S_i)$ denotes the free energy at temperature T_i, while $B(x,S_i,S_j)$ denotes the height of energy barrier between two conformations S_i and S_j. The constant $\xi > 0$ is a constant value that weights the relative importance of thermodynamic stability and having a Boltzmann ensemble consisting of exclusively S_i and S_j with roughly equal frequencies, while $\zeta > 0$ is the weighting factor that influences the importance of the height of energy barrier against thermodynamic stability and equal frequency of each conformation in the complete ensemble.

The objective function specified in Eq. 5.3 generates an RNA sequence (x) by optimising the frequency of occurrences for all meta-stable conformations to be equal based on the Boltzmann ensemble. In Eq. 5.4, an RNA sequence is generated for an RNA molecule that will structurally switch in response to a change in temperature. Equation 5.5 optimises the free energy of the stable states to be relatively close and in addition it optimises the energy barrier between each conformation with a target of ΔE. The calculation of the energy barrier between each conformation ($B(x, S_i, S_j)$) however, requires a complete kinetic folding simulation (Flamm et al., 2000) which is time consuming, even for short RNA molecules.

As suggested by Flamm et al. (2001), the sequence optimisation of design algorithms for multi-stable RNA molecules applies only to the intersection region. Each base position in the sequence (as illustrated in the dependency graph Ψ in Fig. 5.2) can be classified as:

1. Remaining unpaired in all conformations
2. Forming a base pair with the same complementary position in all conformations
3. Forming base pairs with different base positions in each conformation

A dedicated base assignment procedure is applied for each of these cases. Flamm et al. (2001) suggest that for case 1 and 2, a base assignment similar to the method implemented by *RNAinverse*, while for case 3, a fill algorithm (cf. Algorithm. 6, p. 91) that assigns bases for paths and cycles using uniform distribution based on the size of the intersection (cf. item 3 in the generalised intersection theorem, Sec. 5.1.2 on p. 88) is used. Flamm et al. (2001) propose two mutation operators during the optimisation of the sequence. These mutation operators, only apply to base pairs in the intersection region. The first operator called “local mutation” conserves the purine/pyrimidine pattern when mutating a base position (G ↔ A | C ↔ U). The second operator called “non-local mutation” assign to any selected cycle or path a completely new base configuration. For the implementation, Flamm et al. (2001) suggest an extension of the *RNAinverse* algorithm by replacing the objective function with any of the functions $\Xi(x)$ presented earlier (Eqs. 5.3, 5.4 and 5.5) and the inclusion of the two new mutation operators in the adaptive random walk. The mutation applies to any base positions belonging to a path or cycle. Although the application is not publicly available, Flamm et al. (2001) describe a Perl script that managed to design RNA sequences for the multi-stable SV11 molecule by using Eq. 5.5 as objective function.

Based on our findings discussed in chapter 4, we choose to extend the *StochSrch* algorithm with the multi-stable conformation sequence design approached suggested by Flamm et al. (2001) discussed earlier. *StochSrch* is extended with the objective function in Eq. 5.4 and the two new mutation operators in the SLS heuristics as shown in Algorithm 7. As suggested by Flamm et al. (2001), *StocSrchMulti* also restricts the optimisation only to base positions that belong to the intersection region. A simple test of designing an RNA sequence for the SV11 molecule produced a similar result in terms

Algorithm 6 Initialisation of intersection in (Flamm et al., 2001)

```
procedure FILLCYCLE($n$)
    if n = 0 then
        return $\emptyset$
    end if
    $\xi \leftarrow$ uniformly distributed number[0,1]
    if $\xi <$ Fib$(n-1)/2\cdot$(Lucas$(n)$) then
        return 'AU' $\Leftarrow$ fillUPath$(n-2)$
    else
        if $\xi <$ Fib$(n-1)/$Lucas$(n)$ then
            return 'CG' $\Leftarrow$ fillGPath$(n-2)$
        else
            $\xi \leftarrow \xi$ - 2(Fib$(n-1)/(2\cdot$Lucas$(n))$
            if $\xi <$ Fib$(n+1)/(2\cdot$Lucas$(n))$ then
                return 'U' $\Leftarrow$ fillUpath$(n-1)$
            else
                return 'G' $\Leftarrow$ fillGpath$(n-1)$
            end if
        end if
    end if
end procedure

procedure FILLGPATH($n$)
    if n = 0 then
        return $\emptyset$
    end if
    $\xi \leftarrow$ uniformly distributed number[0,1]
    if $\xi <$ Fib$(n-1)/$Fib$(n+1))$ then
        return 'CG' $\Leftarrow$ fillGPath$(n-2)$
    else
        return 'U' $\Leftarrow$ fillUPath$(n-1)$
    end if
end procedure

procedure FILLUPATH($n$)
    if n = 0 then
        return $\emptyset$
    end if
    $\xi \leftarrow$ uniformly distributed number[0,1]
    if $\xi <$ Fib$(n-1)/$Fib$(n+1))$ then
        return 'U' $\Leftarrow$ fillUPath$(n-2)$
    else
        return 'CG' $\Leftarrow$ fillGPath$(n-1)$
    end if
end procedure
```

Algorithm 7 Multi-stable states model of *StochSrch* = *StochSrchMulti*

```
generate dependency graph
categorised each base pair
initialise start sequence (S - intersection) $\leftarrow$ probability biases
initialise intersection $\leftarrow$ Alg. 6
calculate $\Xi(x)$ (Eq. 5.4)
while $not = \Xi(x)$ or step $not =$ MAX do
    $\xi \leftarrow$ uniformly distributed number[0,1]
    if $\xi < prob_{local}$ then
        mutate S = $S_1 \leftarrow$ local mutation
    else if $\xi <$prob$_{non-local}$ then
        mutate S = $S_1 \leftarrow$ non-local mutation
    end if
    calculate $\Xi(x)$ for $S_1$
    if min($\Xi(x)$) or $pb_{acc}$ then
        replace S = $S_1$
    end if
end while
```

of accuracy and a slightly better result in terms of minimum free energy of the generated sequences when compared to the results of the Perl script described by (Flamm et al., 2001).

5.2 A Deterministic Approach to Designing Interacting Multi-stable RNA molecules

The search space landscape for the mapping of RNA sequences to their structures is rugged (Schuster et al., 1994; Schultes et al., 1998; Reidys et al., 1997). Probability biased assignment and stochastic heuristic have been implemented in a number of the sequence designer algorithms to explore this rugged landscape (cf. Chapter. 4). As the length of the molecule increases, the task of exploring the search space becomes exponentially harder (Schuster, 2006). In Chapter. 4, we have demonstrated the ability of a set of rules to solve the inverse prediction problem for RNA without the need for optimisation heuristics. The result in Chapter. 4 for *RepInit* shows that because of the complementary base pairing rules, the assignment of bases can be determined from certain characteristics of each base position. A similar approach has been proposed by Mir (1996) using the three bases assignment (A's, C's and T's) technique in the design of DNA words (short strands of DNA) (Braich et al., 2000, 2002). Although this technique reduces the sequence variability, it improves the stability of the secondary structure by eliminating C-G base pair and accuracy by allowing only the A-T base pair to form. Mir (1996) introduced the technique to improve the predictability of self-assembly among multiple DNA words. Limiting the number of possible base combinations in our case would, however, restrict the ability to exploits the conformational flexibility of nucleic acid molecules for the design of our computational units.

As an alternative, a deterministic approach is proposed where the assignment of every possible base combination is evaluated. In the worst case, such an approach would not scale well in term of time complexity, as the algorithm depends not only on the structure's length, but also the number of meta-stable conformations of the molecule. There are two arguments that motivate the implementation of this deterministic approach: the design space that is restricted to a parameter envelope that includes all known computational nucleic acid units and, the multi-objective nature of the problem. The length of the molecules is limited to only 200 nt, but equally important is the length restriction that is being apply to each of the secondary structure motif as listed in Tab. 4.1. Because individual secondary structure elements are limited to the size of 25 nt, we can pre-compute the possible tree of base combinations to speed up the processing time. The objective functions in Eqs. 5.3, 5.4 and 5.5 are directed not only towards generating sequences for RNA molecules with multi-stable conformations according to their frequency of occurrence (Eq. 5.3), temperature dependence (Eq. 5.4) and the energy barrier height (Eq. 5.5), but at the same time minimise their free energies. The meta-stable conformations are known to exist within a certain energy gap from the minimum free energy of the molecule (Schultes and Bartel, 2000). By evaluating the free energy contribution of each possible base combination during base assignment, we can eliminate earlier a number of base combinations, and this makes the approach tractable for the design space of interest.

As proposed by Flamm et al. (2001), our initial task in developing our deterministic approach is to derive a dependency graph for identifying the "intersection" region. From the dependency graph, we extract a pathway list that consists of the paths and cycles that are present in the multi-stable conformations of the molecule. The sequence optimisation routine is then applies to only the base positions belonging to the "intersection" region. In the design of nucleic acid molecules for information processing, we focus on the trans-acting switches. For example, a short nucleic acids strand is introduced to bind with a receptor site of the allosterically controlled ribozyme and triggers a conformational switch of the molecule. To enable such designs, we need to extend the designer to include the possibility of multi-folding formation between two or more molecules.

As discussed in Sec. 2.3, the design of a functional RNA unit with a single input receptor acting as regulator, requires two meta-stable conformations. These are the "inactive" conformation where a particular region of the molecule that is responsible for catalytic reaction (ribozyme core) is distorted and the "active" conformation which resembles the conformation of a functional RNA unit. With the addition of multiple molecules in our designer, we also allow each molecule to have its own meta-stable conformations. This enables our sequence design algorithm to generate sequences for the design of a network of nucleic acid units where each unit has its own meta-stable conformations that affect

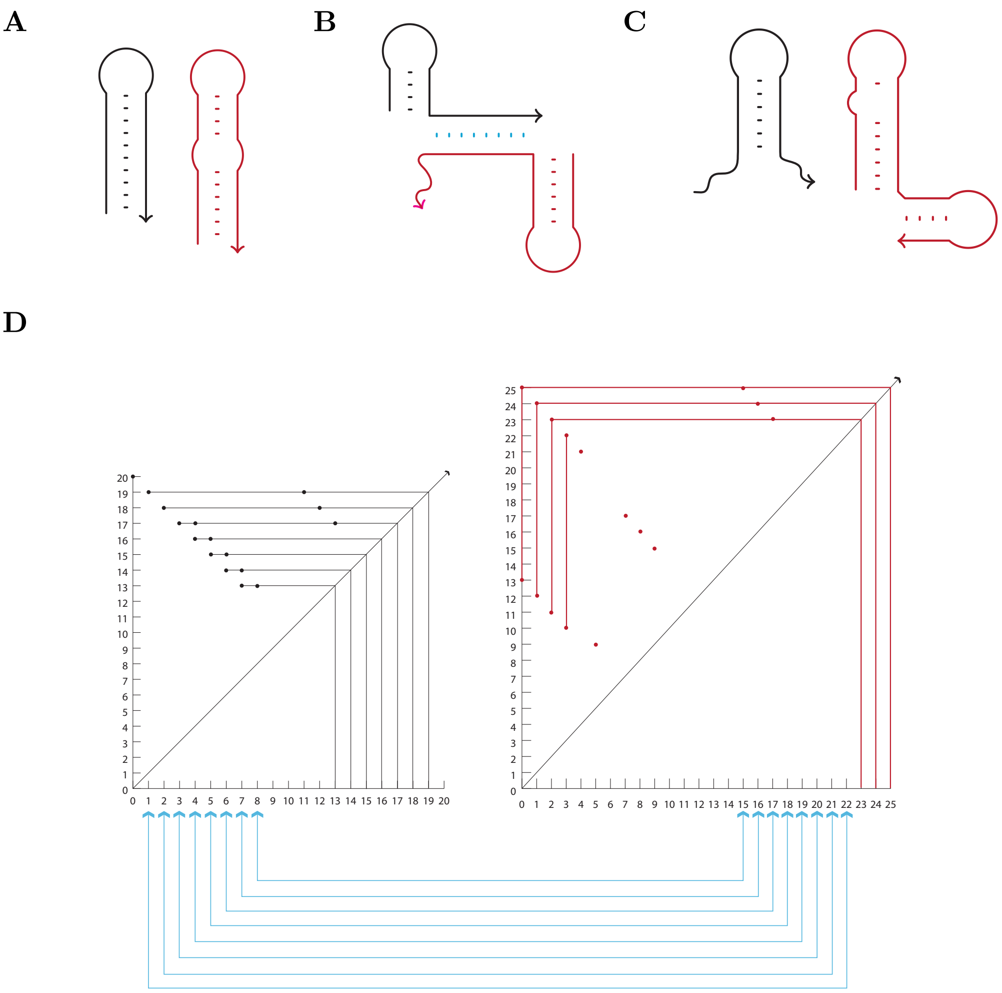


Figure 5.3: The extended dependency graph Ψ^* depicting the base pairing interactions between two arbitrary multi-stable molecules. Both molecules `8(....8)` (black) and `5(..3(....)3..5)` (red) have three meta-stable conformations (A, B and C), for A and C, these molecules fold internally to themselves and in B, the two molecules externally bind together. In each state, the conformations are unique as visibly presented in the figure. The extended dependency graph (D) plots the base pairing relationship for both molecules separately into two dot-plot-like graphs. The base positions are represented in the x and y-axes, with dots denoting base pairs between node (x,y), plotted in the direction of 5′ to 3′, without repeating any previous base pairs. Paths are denoted by edges that connect these base pair points, while any edges that form a triangle path represent cycles (shown in Fig. 5.4B). The cross-linkage between the dot-plots (in blue) represents the inter-molecular binding between the two molecules. Refer to Figs. 5.4 and 5.5 for explanation of panel D.

other units (i.e., form external base pairs with other units) within the network. We propose a new type of dependency graph Ψ^* illustrated in Fig. 5.3. For clarity, the details of Fig. 5.3D are explained in Figs. 5.4 and 5.5.

In this dependency graph, the base position dependencies for both internal (meta-stable conformations for each molecule) and external (the inter-molecular pairing in each conformation for each molecule) base pairs are shown. In Fig. 5.3, we illustrate two random molecules, molecule I = `8(....8)` and molecule II = `5(..3(....)3..5)`. Both molecule I and II have three meta-stable conformations as illustrated in Fig. 5.3A, B and C. In state B, the two molecules form external base pairs which is indicated by the blue edges that connect two sets of vertices in the dependency graph. Instead of using the circular planar graph representation proposed by Flamm et al. (2001), we use the dot-plot representation (cf. Fig. 2.12D on p. 31) to make paths and cycles more closely identifiable. This is particularly useful where the length of the molecules or the number of molecules and the number of meta-stable conformations overwhelm the circular planar graph.

Similar to the dot-plot representation (Hofacker, 1994), x-axis and y-axis represent the position in the sequence, in the usual 5′ to the 3′ order. We then plot all the occurring base pairs (in all states) in the structures only once, moving in the direction of the 5′ to the 3′ end. If two or more dots with the same values of x or y are present, a vertical line (for the y-axis) and a horizontal line (for the x-axis) are drawn to represent a path or a potential of a cycle formation (bold lines in Fig. 5.5A and B). These lines are then connected with the diagonal line. Any vertical line that is not connected to a horizontal line (in the upper half of the dot-plot, which is separated by the diagonal line) represents single (marked as "single" in Fig. 5.5A) which is a position that forms base pairs with

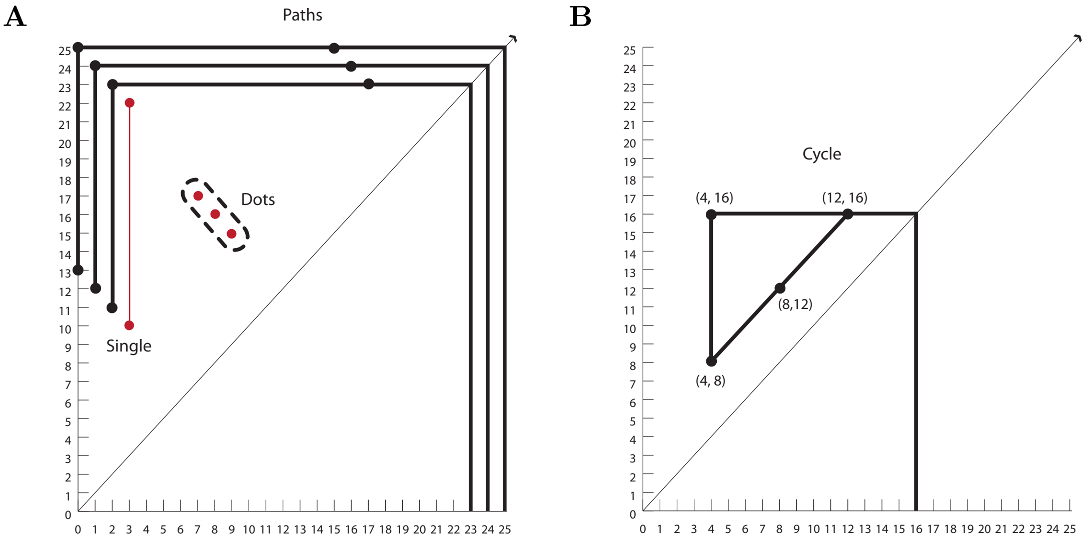


FIGURE 5.4: Elements of an extended dependency graph (Ψ^*). Path, singleton, and dot elements are presented in (A), while cycle element is shown in (B). A Path is presented as black bullets connected by a vertical line and a horizontal line that connects to the diagonal line. A singleton is represented as two red bullets that are connected with a vertical line. Dots are red bullets and are not aligned with any other bullets (either vertically or horizontally). The coordinates in (B) are equivalent to the base pairs that form the cyclic dependency. In two conformations (A and B), position 4 can paired with position 8 and position 12 can paired with position 16 in conformation A, while position 4 and position 16, as well as position 8 and position 12 can form base pairs in conformation B.

the two different positions in each conformation. In order to find the start position of a path or cycle, we draw a vertical line from the intersection point with the diagonal line to the x-axis.

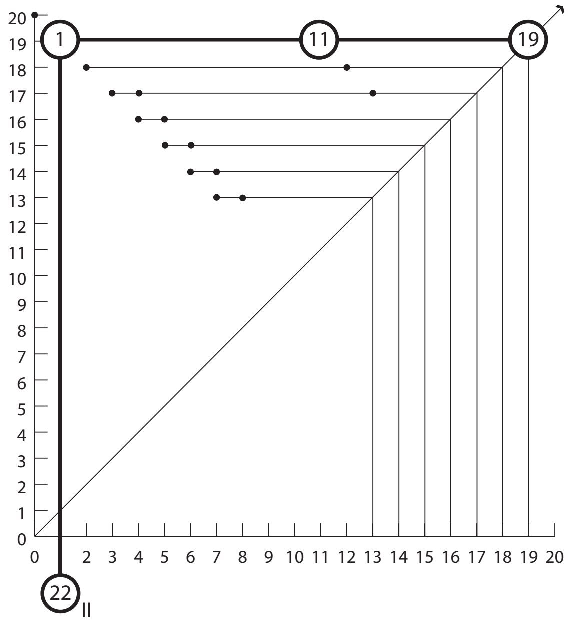


Figure 5.5: Detail of the dependency graph shown in Fig. 5.3D. Path shown in bold, starts with position 19 which can pair with position 11 in one conformation, position 11 paired with position 1 in another conformation, and position 1 have the possibility of forming an external base pair with position 22 in molecule II, in another conformation.

For instance, in Fig. 5.5, we can trace a path that start from position 19 in molecule I and position 22 in molecule II as follows: 19 - 11 - 1 - $22_{(II)}$. Cycles are identified, if in the upper half of the dot-plot, there exist a diagonal line that connects the start point of the vertical lines and the end point of the horizontal line forming a triangle (Fig. 5.5B). If there is an odd number of dots present in the triangular path, then one can conclude that the Ψ^* is not bipartite. From the graph, we can also pick up dots (marked as "Dots" in Fig. 5.5A) which is a base position that forms a base pair with the same base position in all conformations. In our implementation, a procedure that resembles the recognition of patterns in the dot-plot-like dependency graph (presented here) has been developed to produce a list of dependency pathways consisting of paths, cycles, singles, dots and unpaired bases.

With the list of dependency pathways established, we can construct the trees of all possible base pairing combinations based on the length of the paths and cycles in the dependency pathways. Let us consider a simple case where all of the base positions will be unpaired, then the combination of all bases would be equivalent to a set of bases $\mathcal{A}$ = {A,U,C,G}. Accordingly, for dots (a base pair occurring in all conformations), the possible base combinations are equivalent to the set of complementary base pairings $\mathcal{B} = \{\mathrm{CG,GC,AU,UA,GU,UG}\}$. To assign the complete base combination for the path, we can combine any base pairs from the set of $\mathcal{B}$, but for a cycle, we take the formation of base pair between the start and end positions into consideration.

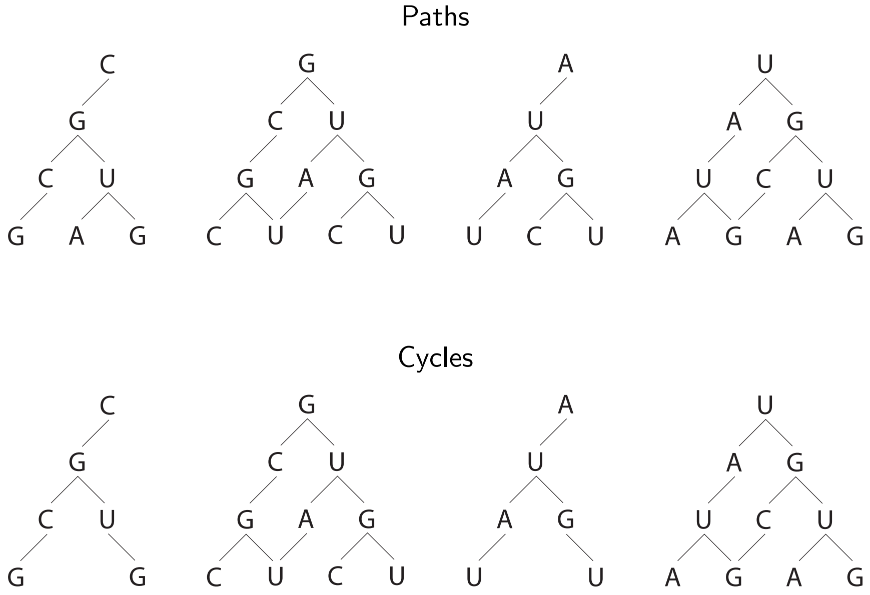


FIGURE 5.6: The combinatorial base configuration tree for a path and a cycle 4 nt long. For a base configuration starting with the bases G or U, the total number of configurations for both path and cycle are identical. However, for bases C or A, the end position can only be occupied by a complementary base to ensure that a cycle is formed.

In Fig. 5.6, the process of deriving the complete base pair configuration of a cycle and a path 4 nt long is illustrated as the recursive process of adding bases from the complementary set $\mathcal{B}$ to the rooted tree. If a tree starts with cytosine (C), then the next branch that follows would comprise guanine (G), followed by the CG base pair from the complementary set $\mathcal{B}$. This G-branch can forms base pairs with both cytosine (C) and uracil (U), as both GC and GU are also present in $\mathcal{B}$. In the figure, we observe the differences between the trees of "Paths" and "Cycles" that begin with cytosine (C) and adenine (A). The two trees for "Paths" that start with base C and base A have one extra branch when compared with the same two trees for "Cycles". The possible base combinations for cycles are always less than the possible base combinations for paths because cycles, require pairing base between the start and end positions. Base pair AC is not complementary for the tree of cycles that starts with cytosine (C), and the base pair CA is not complementary for the tree of cycles that starts with adenine (A). In our implementation, we pre-generate all possible base combinations up to the depth of 25 edges to reduce computation time.

Given the list of dependency pathways, we consecutively select each element from the list and iteratively calculate the objective function ($\Xi(x)$) for each of the possible base assignments (x_i) to the element using a dynamic programming algorithm. For each element, we sort the possible base combination according to $\Xi(x_i)$ and retain the best candidates. The selection of element from the list of dependency pathways is made either randomly or in a descending order, i.e., long cycles are selected first, followed by long paths, singles, dots and unpaired positions. After all elements in the dependency pathway list have been evaluated, a "traceback" routine (Sect. 2.4.1.3 p. 34) then retrieves the optimal solution. In addition, we also retrieve suboptimal solutions where the value of the objective function is within a defined range from the minimal value.

During our trial runs, we found out that possible base combinations were discarded prematurely. Sequences that were later found to be close to the best solution were eliminated early. To resolve this issue, we decided to keep a number of suboptimal solutions during each iteration. The size of the set of suboptimal solution is denoted as (Λ). Given μ as the maximum number of possible base combinations for each element in the dependency pathways, if $\Lambda = \mu$, then all possible base combinations are kept and explored. However, setting the $\Lambda = \mu$ is computationally expensive. A large value of Λ would be desirable to improve the solution candidates, but significantly increases computation time. In our algorithm, we also include a tournament ranking selection (Bäck et al., 2000) to sort the suboptimal solutions during each iteration. This method decreases the bias of selecting only the best suboptimal solutions after each iteration. The algorithm, called *multiSrch*, is presented in detailed in Algorithm. 8. The KEEP variable denotes the error difference for the cost function between a partial candidate sequences against a minimum solution during each iteration, used to measure the selection of possible suboptimal candidate into a list (L_s). The Λ variable denotes the number of suboptimal candidates that will be carried over to the next iteration.

The *multiSrch* has more in common with the dynamic programming approach for predicting RNA secondary structure, then a conventional optimisation approach as used in *StochSrchMulti*. The *multiSrch* algorithm calculates the objective function value for every possible base combination belonging to a given pathway element. At each stage, base combinations that contributed to the minimal objective function value are kept, together with base combinations with objective function values that are within the KEEP threshold. The process is repeated until all elements in the dependency pathways list have been selected. The algorithm prunes any possible base combinations that are not part of the suboptimal list. If $\Lambda = \mu$, then every possible base assignment is kept, which can grow exponentially and increases the time complexity of the algorithm. Using the values of the objective function calculated during the "fill" operation (Sect. 2.4.1.3 p. 34), we then apply a "traceback" routine to generate the set of sequences.

Two different objective functions are used by the algorithm, one for "self-induced switches" and the other for "trans-acting switches". The thermodynamic characteristics of the two

Algorithm 8 The algorithm for interacting multi-stable states molecules sequence design = *multiSrch*

```
generate set of pathways $\mathcal{D}(\Psi^*)$
initialise dummy sequence $(x)$ with conserved bases and X's
add $x$ into list of candidates $L_c$
for each pathway $(p)$ in $\mathcal{D}(\Psi^*)$ do
    for each candidate sequence $x \in L_c$ do
        if $p$ is cycle then
            Calculate $\Xi(x)$ for all cycle combinations where size $= l(p)$
            if $\Xi(x) < \Xi(X)$ then
                replace $X = x$
            else if $\Xi(x) <$ KEEP then
                add $x$ to suboptimal list $L_s$
            end if
        else if $p$ is path then
            Calculate $\Xi(x)$ for all cycle combinations where size $= l(p)$
            if $\Xi(x) < \Xi(X)$ then
                replace $X = x$
            else if $\Xi(x) <$ KEEP then
                add $x$ to suboptimal list $L_s$
            end if
        else if $p$ is unpaired then
            Calculate $\Xi(x)$ for assignment of $\forall\, p \in \{\texttt{A,U,C,G}\}$
            if $\Xi(x) < \Xi(X)$ then
                replace $X = x$
            else if $\Xi(x) <$ KEEP then
                add $x$ to suboptimal list $L_s$
            end if
        end if
    end for
    re-sort $L_s$ according to $\Xi(x)$
    for each $x \in L_s$ do
        randomly select opponents
        compare $x$ with random opponents
        re-sort $L_s$ based on tournament score
    end for
    for each $x \in L_s$ until $\Lambda$ do
        add $x$ into $L_c$
    end for
end for
```

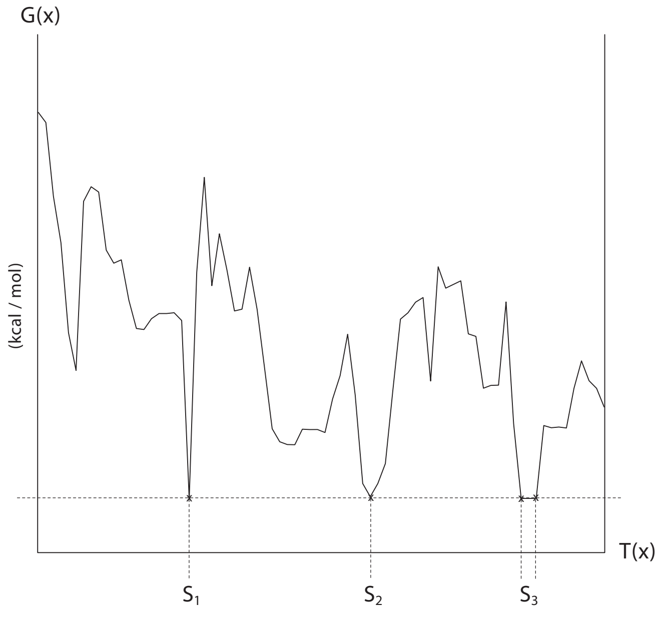


FIGURE 5.7: A simplified representation of the free energy profile of a multi-stable RNA molecule with self-induced switching capability. Conceptually, for a molecule x, there is a time series of T(x) required for the molecule to kinetically fold into its native state. The G(x) axis represent the Gibbs free energy. The different peaks in the free energy axis indicate the formation and displacement of base pairs (structure rearrangement) in order to arrive to its native conformation. The three lowest folds of S_1, S_2 and S_3 which are positioned almost at the same energy level represent the condition in which the three unique meta-stable states in a self-induced structural switching unit are normally present. The plots are simulated using *Kinfold* (Flamm et al., 2000) for a random sequence with predicted folding equivalent to both conformations of the SV11 RNA species described in (Biebricher and Luce, 1992).

types of switches can be seen as sampling of the calculated free energy shown for the self-induced switch in Fig. 5.7 and for the trans-acting switch in Fig. 5.8. In figure 5.7, we observe that the free energy ΔG of the meta-stable conformations (S_1, S_2, and S_3) is relatively similar. This is consistent with the experimental finding of Schultes and Bartel (2000). As suggested by Flamm et al. (2001), the objective function (Ξ Eq. 5.3) is sufficient in evaluating the performance of sequences for self-induced structural switching molecules. Equation 5.3 optimises the candidate sequences to favour the two meta-stable conformations by minimising the free energy differences between these meta-stable states to be close to or exactly zero (i.e., each state has the same free energy)

From Fig. 5.8, we observe that the meta-stable states (S_1, S_2 and S_3) sit at a different energy level in contrast to the multi-stable state depicted in Fig. 5.7 above. The trans-acting switching molecule of an RNA PASS gate (described in Penchovsky and Breaker (2005)) simulated here requires the binding of an effector molecule to the receptor site to trigger a conformational shift. In a time series of T(x) folding, calculated by *Kinfold* (Flamm et al., 2000) the S_1 fold corresponds to the meta-stable state where the gate is inactive, fold S_2 denotes the meta-state where a short RNA effector molecule is intro-

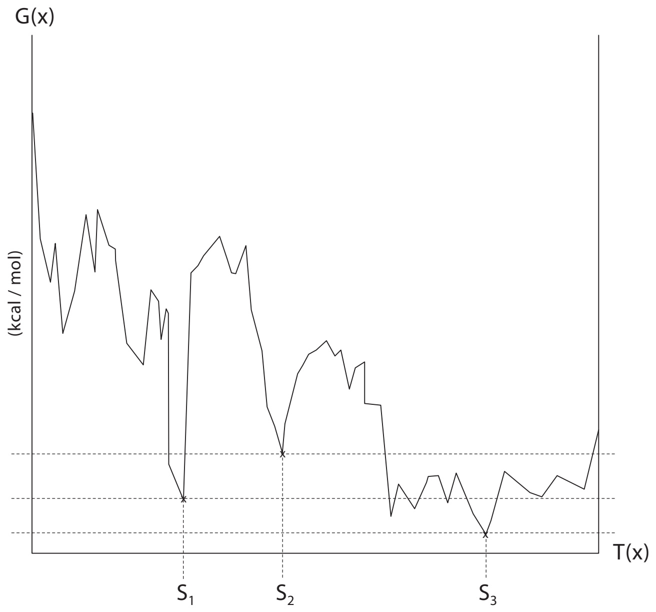


FIGURE 5.8: A simplified representation of the free energy profile of a molecular PASS gate (Penchovsky and Breaker, 2005) with trans-acting structural switching capability. The schematic is produced with free energy calculation, simulated from *Kinfold* with estimated inter-binding energies by *RNAup* (Mückstein et al., 2006), under the assumption that an RNA effector molecule is chosen to trigger the conformational change. Fold S_1 corresponds to an inactive gate, with fold S_2 representing the meta-stable state where refolding occur (as the effector molecule is introduced) and later form the active conformation in S_3. The three conformations differed in free energy.

duced, and fold S_3 corresponds to the active conformation of the PASS gate. *RNAup* predicts the binding energies between the RNA effector and the RNA PASS gate in Fig. 5.8. Multi-stable trans-acting molecules are more suited for the design of nucleic acid computers. Control molecules (e.g., effector as input) can be introduced to activate or deactivate the function of these computers. Equation 5.5 suggested by Flamm et al. (2001) is suitable for the trans-acting case, but computing the energy barriers between these meta-stable states to calculate the objective function is time consuming for the algorithm. A complete kinetic folding simulation is required for each possible base combination belonging to an element in the dependency pathway list. This significantly increases the CPU time of the algorithm.

To resolve this issue, we augment the objective function given in Eq. 5.3 with two additional terms. First is the energy gap value, (E_{gap}), that corresponds to the desired energy differences among meta-stable states, introduced by Penchovsky and Breaker (2005). The energy gap value is written as the difference of free energy between two meta-stable states given a sequence x, $E_{gap}(S_1, S_2) = |E(x, S_1) - E(x, S_2)|$. For more than two meta-stable states, the energy gap value is equivalent to:

$$E_{gap}(S_1, S_2, \ldots, S_m) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \mid E(x, S_i) - E(x, S_j) \mid \tag{5.6}$$

where, m is the number of meta-stable states. For instance, if $m = 4$, the energy gap value is given as,

$$\begin{aligned} E_{gap}(S_1, S_2, S_3, S_4) = \; & \mid E(x, S_1) - E(x, S_2) \mid + \mid E(x, S_1) - E(x, S_3) \mid \\ & + \mid E(x, S_1) - E(x, S_4) \mid + \mid E(x, S_2) - E(x, S_3) \mid \\ & + \mid E(x, S_2) - E(x, S_4) \mid + \mid E(x, S_3) - E(x, S_4) \mid \end{aligned}$$

The E_{gap} value can be tuned for specific applications. For the design of nucleic acid molecules for information processing, a value of E_{gap} = 6–10 kcal/mol is suggested in (Penchovsky and Breaker, 2005). The second term we introduce is the intermolecular base pairing efficiency, (ΔG_{int}) representing the disassociation of base pairs in the intramolecular binding region and formation of intermolecular base pairs calculated using the *RNAup* program (Mückstein et al., 2006). With these extension to Eq. 5.3, the objective function is now:

$$\begin{aligned} \Xi_D(x) \; = \; & \sum_{i=1}^{m} \Big[E(x, S_i) + \Delta G_{int}(x, S_i) \Big] - mG(x) \; + \\ & \xi \sum_{i=1}^{m} \sum_{j=i+1}^{m} \Big[E_{gap} - \big(E(x, S_i) - E(x, S_j) \big) \Big]^2 \end{aligned} \tag{5.7}$$

where $E_{gap} = E_{gap}(x_1, x_2, \ldots, x_m)$ and the $\Delta G_{int}(x, S_i)$ is the free energy of meta-stable conformation S_i, where $\Delta G_{int}(x, S_i) = 0$, if S_i is does not have any intermolecular binding. We introduce another term (ΔG_{opt}) to allow one to specify target minimum free energy to accompany the target conformation. The objective function with ΔG_{opt} becomes:

$$\begin{aligned} \Xi_E(x) \; = \; & \sum_{i=1}^{m} \Big[E(x, S_i) + \big(\Delta G_{opt} - \Delta G_{int}(x, S_i) \big) \Big] - mG(x) \; + \\ & \xi \sum_{i=1}^{m} \sum_{j=i+1}^{m} \Big[E_{gap} - \big(E(x, S_i) - E(x, S_j) \big) \Big]^2 \end{aligned} \tag{5.8}$$

Eq. 5.8 is the objective function we use for generating sequences where more than two molecular units are required.

The objective functions ($\Xi(x)$) used in the design of sequences for multi-stable conformation are single aggregate objective functions. For instance in Eq. 5.7, either the free energy or the energy gap is minimised depending on the weighting factor (ξ). From our trial runs, the implementation of weighting factor in the objective function failed to generate sequences that follow a specific criterion. In addition to the single aggregate objective function approach to handle the multi-objective optimisation required in the sequence design, a sorting-bins approach is derived for the algorithm. There are three different cost values that we need to consider. The first is the objective function itself (Eq. 5.7 or 5.8), secondly the intermolecular base pairing efficiency measured as ΔG_{int} and thirdly the minimum free energy value. Each bin contains all candidates, but each bin is sorted according to the different cost function in descending order. In the algorithm, only the top candidates from each bin are selected for the next iteration. For specific design goals it can be advantageous to work with only one of the cost function. For example to design an effector molecule, optimising ΔG_{int} will increase the probability of generating sequences where the effector and receptor binding site (OBS) are perfectly complementary.

Instead of just sorting the candidates in descending order, we also added a tournament-ranking selection (Bäck et al., 2000) to reduce the possibility of becoming stuck in local minima. This is necessary otherwise in the early stages of the execution, pre-mature pruning of base assignments would occurs. With only a fraction of the base positions already specified at this stage, the calculation of the objective function is not accurate. At every stage, in order to reduce the execution time of the algorithm, we need to prune any base combinations with poor objective function values. It is possible that these discarded base combinations could lead to the discovery of an optimal sequences. By starting with the longest path or cycle, we try to increase the number of bases available during the objective function calculation. The tournament-ranking selection adds to the effort of ensuring the list of suboptimal solution will be a sufficient representative of the complete base pair combinations in the early stages of the execution. In cases where different sequences for a given structure are required, the algorithm randomised the ordering of dependency pathway element. This can also be achieved by increasing the number of suboptimal candidate.

In this section we proposed a deterministic approach to design sequences for molecules with multi-stable conformations called *multiSrch*. The algorithm is developed specifically for trans-acting switches, which is the type of molecule investigate for constructing computational unit. The algorithm first creates a list of dependency pathways consisting of “intersection” elements. Using dynamic programming we calculate the objective function of assigning all possible base combinations for each element. During each iteration, a list of suboptimal candidates is kept. The “traceback” routine then retrieve the optimal sequence together with a list of suboptimal sequences. The performance of *multiSrch* algorithm is discussed in the next section.

5.3 Evaluating the performance of multi-stable sequence designers

There are two types of structural switching (self-induced and trans-acting) that we need to consider in evaluating the multi-stable sequence design algorithms. The thermodynamic characteristics representative of the two switches are illustrated in Fig. 5.7 for self-induced switches and Fig. 5.8 for the trans-acting switches. In order to evaluate the overall performance of the multi-stable sequence designers, we examine the thermodynamic characteristic of both cases. Firstly the performance when designing self-induced switching molecules (where the free energy levels of the state are equal, i.e., $E_{gap} = 0$) and secondly, the performance of designing trans-acting molecules where the meta-stable states have different free energy levels but the difference E_{gap} is a design target.

In nature, only a few RNA molecules with multi-stable conformations have been identified (cf. Sec. 5.1). Most natural switches are either too large (more than 300 nt in length) or to complex (contain pseudoknot motifs) to of interest for our purpose. Aside from the two SV11 molecules (Biebricher and Luce, 1992; Zamora et al., 1995) we have included a number of nucleic acid logic gates that have been engineered in the laboratory as part of the dataset for the comparison study. Although the selection of nucleic acid logic gates may seem inappropriate for self-induced structural switching because the logic gates follow the trans-acting switches, we use them because these molecules are known to have multi-stable conformations. Even the computational procedure that was used to design these nucleic acid logic gates initially ignored the influence of the effector molecules (Penchovsky and Breaker, 2005; Penchovsky and Ackermann, 2003). Only later in the design process, the E_{gap} parameter is factored in to simulate the trans-acting structural switching.

The structures included in the dataset of RNA molecules with meta-stable conformations are listed in Tab. 5.1. The dataset is referred as “Multi-stable” (DS-MS). In this section, we conduct two evaluation studies. First, *StochSrchMulti* is compared against *multiSrch* to generate sequences for self-induces switches. This comparison study is referred as Test-Self-Switch (TSS). Secondly, *multiSrch* is evaluated to generate sequences for the trans-acting switches. This study is referred as Test-Trans-Switch (TTS).

In this section we evaluate the performance of *multiSrch* against *StochSrchMulti* for the self-induced switches (TSS) using dataset DS-MS (Tab. 5.1). The dataset comprises 18 molecules that have two or four meta-stable conformations. For the TSS study, we assigned loop elements to the region in the meta-stable conformations where external base pairing occurs. This is necessary because the majority of the molecules are engineered to switch in the presence of another molecule (trans-acting switches). For *StochSrchMulti* (ref. Algorithm 7), the parameter settings are listed in Tab. 5.2. The probability for assigning bases (pb_paired and pb_unpaired) in *StochSrchMulti* is equivalent to the de-

Table 5.1: Multi-stable dataset (DS-MS) for the evaluation of designers for multi-stable molecules. Molecules numbered 1–11 have two meta-stable conformations, while the remaining molecules (12–18) have four meta-stable conformations. Only the two SV11 molecules are found in nature, while the remaining 16 have been constructed in the laboratory. In the table, St. denotes the number of states for each molecule and Eff. denotes the number of effector molecules that will bind to the molecules.

No.	Type	St.	Eff.	Ref.
1	SV11 plus	2	-	(Biebricher and Luce, 1992; Zamora et al., 1995)
2	SV11 minus	2	-	(Biebricher and Luce, 1992; Zamora et al., 1995)
3	RNA_PASS_1	2	1	Penchovsky and Breaker (2005)
4	RNA_PASS_2	2	1	Penchovsky and Breaker (2005)
5	RNA_NOT	2	1	Penchovsky and Breaker (2005)
6	RNA_OR	4	2	Penchovsky and Breaker (2005)
7	PORTA_PASS	2	1	Porta and Lizardi (1995)
8	BURKE_TRAP	2	1	Burke et al. (2002)
9	DNA_PASS_1	2	1	Stojanovic and Stefanovic (2003b)
10	DNA_PASS_2	2	1	Stojanovic et al. (2002)
11	DNA_NOT	2	1	Stojanovic et al. (2002)
12	RNA_AND	4	2	Penchovsky and Breaker (2005)
13	DNA_AND_1	4	2	Stojanovic and Stefanovic (2003b)
14	DNA_AND_2	4	2	Stojanovic et al. (2002)
15	DNA_AND_3	4	2	Kolpashchikov and Stojanovic (2005)
16	DNA_AND_4	4	2	Kolpashchikov and Stojanovic (2005)
17	DNA_MULTI_1	4	2	Stojanovic et al. (2002)
18	DNA_MULTI_2	4	2	Stojanovic and Stefanovic (2003b)

fault setting used in *StochSrch* (cf. Tab. 4.3). In order to set the MAX parameter (i.e., number of iteration) for *StochSrchMulti*, we first evaluated the CPU time required for *multiSrch* in our trial runs. Based on our observation, we then assigned a MAX value that is equivalent to five times the longest processing time required by *multiSrch* to design molecules in the multi-stable dataset (DS-MS). For each molecule in the datasets (DS-MS), a total of 100 runs are conducted for *StochSrchMulti*.

For the TSS comparison study, only the best 50 solutions are consider from these 100 runs. In contrast to *StochSrchMulti*, *multiSrch* is a deterministic algorithm and therefore one run is sufficient. The TSS comparison study is conducted in two phase. In the first phase, only molecules from dataset DS-MS with two meta-stable conformations are compared (structures number 1 to 11). In the second phase the remaining structures 12 to 18, which are molecules with four multi-stable conformation are compared. The parameter settings for the *multiSrch* algorithm (cf. Algorithm 8) are shown in Tab. 5.3. The value of '$\mathcal{D}(\Psi^*)$ Order' denotes the type of sorting applied to the dependency pathways. Here the dependency pathways are sorted in descending order with regards to the number of edges on the path. For paths with an equal number of edges, the path starting earlier on the sequence takes precedence.

Table 5.2: Default parameter setting for *StochSrchMulti*. The default parameters inherited from *StochSrch* (cf. Tab. 4.3) are left unchanged except for MAX and pb_{acc}. *StochSrchMulti* adds two additional parameters for the probability of the mutation operators. Equation 5.3 is used as the default objective function.

Parameter	Value
MAX	100000
pb_paired	$P_G = 0.55$, $P_C = 0.30$, $P_A = 0.10$, $P_U = 0.05$
pb_unpaired	$P_A = 0.80$, $P_U = 0.10$, $P_G = 0.06$, $P_C = 0.04$
reset	1000
pb_prand	0.2
pb_acc	0.5
$\Xi(x)$	Eq. 5.3
pb_local	0.65
pb_nonlocal	0.35

Table 5.3: Default parameter setting for *multiSrch*. The Λ denotes the size of list for the suboptimal candidates. The $min(\Xi(x))$ term represents the minimum value of the objective function over all base assignments to one element of the dependency pathways.

Parameter	Value
$\mathcal{D}(\Psi^*)$ Order	descending
$\Xi(x)$	Eq. 5.7 with $E_{gap} = 0$
Λ	50
KEEP	$\Xi(x) - min(\Xi(x)) = \pm 5.0$

For self-induced switches, we aim for an E_{gap} value of zero based on the free energy profile of self-induced switches illustrated in Fig. 5.7. The minimum free energy conformation is always a part of the meta-stable conformations. Therefore we can arrive at the sequence design of self-induced switches by minimising the free energy for any one of the meta-stable conformations and subsequently minimise the difference between the lowest free energy of a conformation against the other states.

Figure 5.9 shows the variance for both *StochSrchMulti* and *multiSrch* for the molecule types 1–11 of the DS-MS dataset (cf. Tab. 5.1). From the 100 runs conducted for *StochSrchMulti*, the best 50 solutions were selected and plotted in the graph. This is because, in *multiSrch*, only 50 sequences were generated and any unfairness in the comparison study should be in the favour of *StochSrchMulti*. In Figure. 5.9, *StochSrchMulti* and *multiSrch* found sequences with $E_{gap} = 0$, for most of the molecules. For the remaining molecules, sequences within $E_{gap} < 0.2$ were generated by both algorithms. This is indicated by the minimum data values (lower whiskers or bottom hinge of the boxplots). In terms of E_{gap}, the variance of *multiSrch* result (within 0.5 kcal/mol) is also smaller than that of *StochSrchMulti* (within 1.5 kcal/mol). This indicates ability

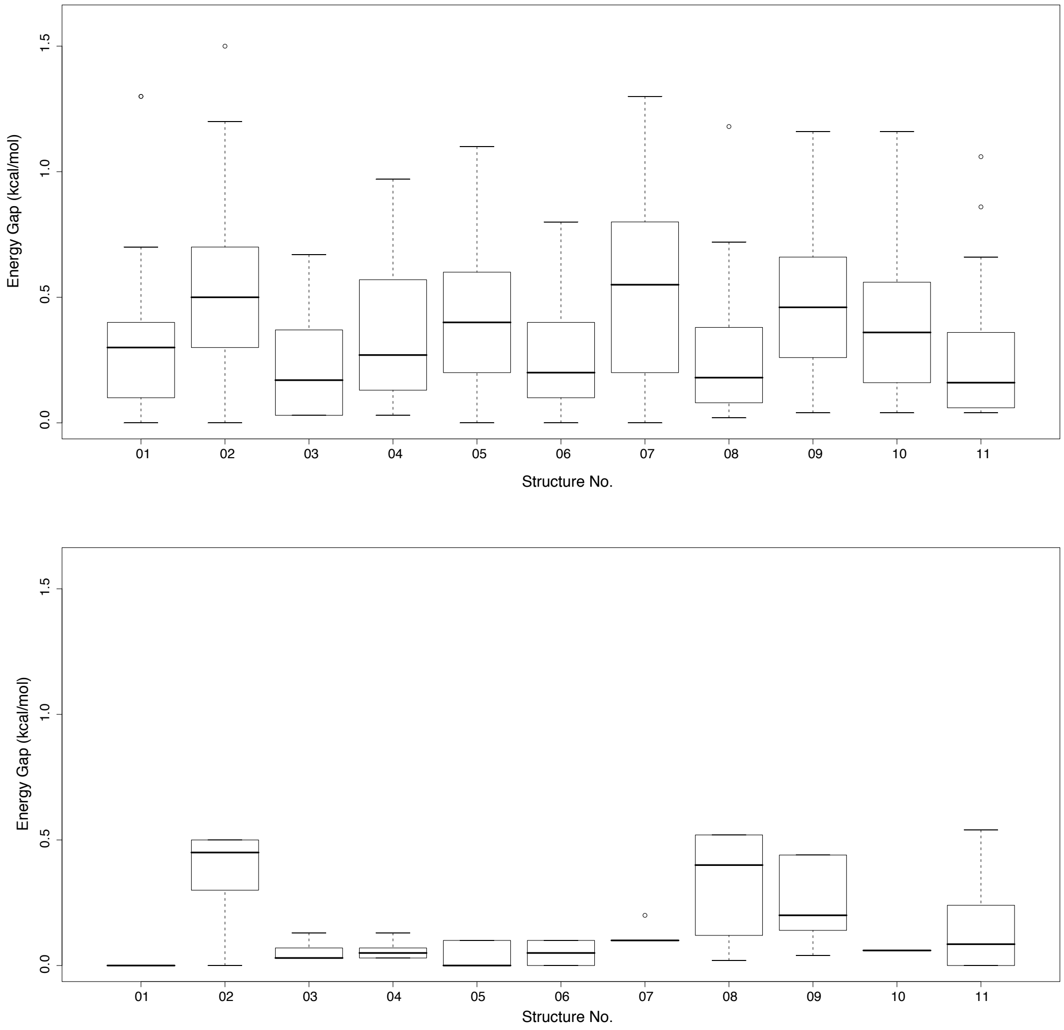


FIGURE 5.9: Variance of energy gap (E_{gap}) for the design of self-induced switching molecules with two states. Best 50 sequences from 100 runs of *StochSrchMulti* (top). The final 50 sequence candidates from one run of *multiSrch* (bottom). Both *StochSrchMulti* and *multiSrch* are able to generate sequences where $E_{gap} \leq 0.2$ for the molecules in the DS-MS dataset (cf. Tab. 5.1) as indicated by the minimum data points. The variance of the results from *multiSrch* is found to be smaller than the variance of those from *StochSrchMulti*. For the case of self-induced structural switching, meta-stable states of equal free energy ($E_{gap} = 0$) are desirable.

of the algorithm in generating a set of sequences (including suboptimal sequences) with comparable difference (E_{gap}) among these two meta-stable conformations.

To evaluate the quality of these sequences, we plotted the minimum free energy and the E_{gap} of each candidate into a bagplot graph (Rousseeuw et al., 1999), shown in Fig. 5.10. Each panel shows the two dimensional boxplot of both the minimum free energy of the secondary structures for the sequences and the energy gap between the two meta-stable conformations. The main components of a bagplot graph are a *bag* that comprises of 50% of the data points (darker blue region), a *fence* that separates inliers

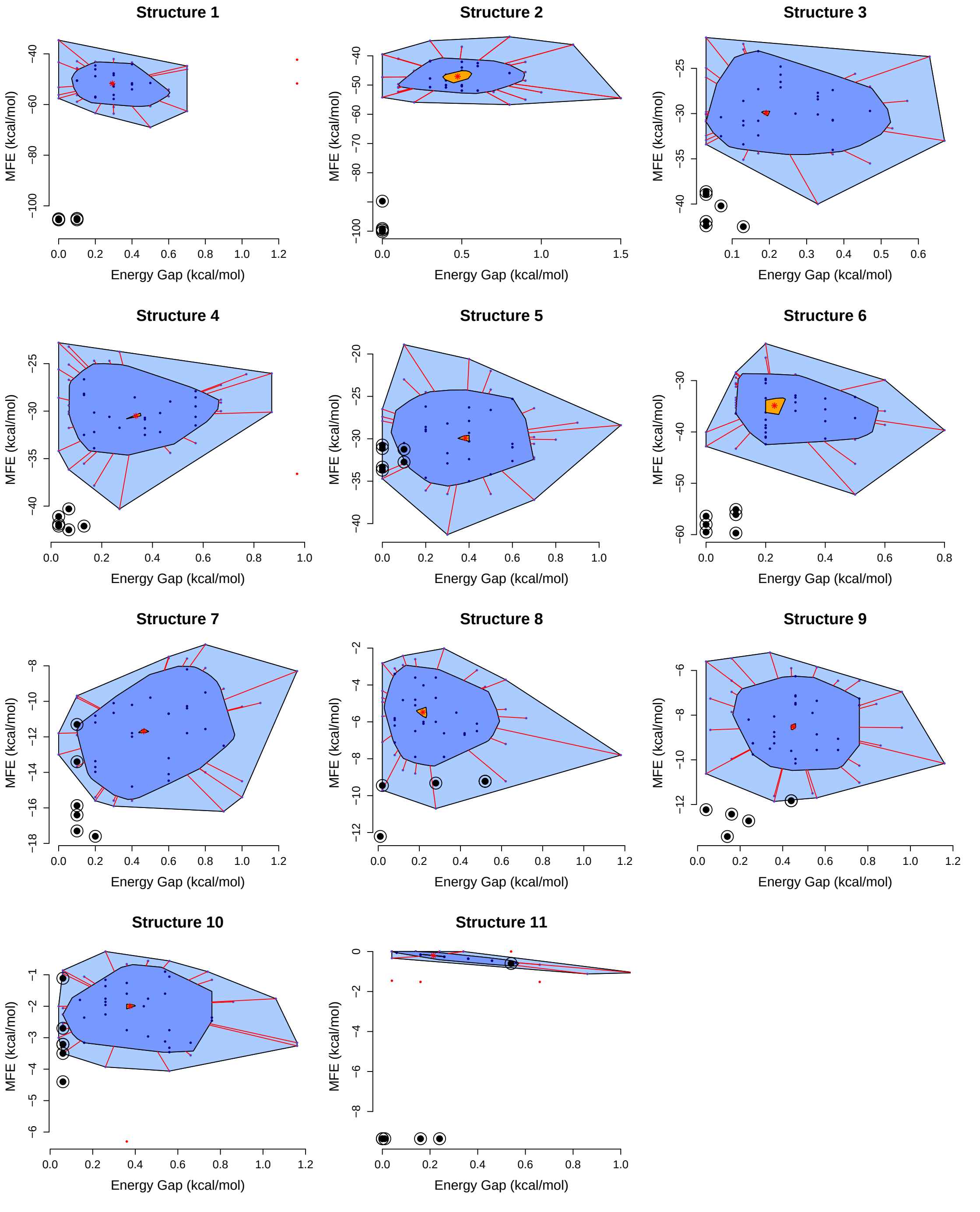


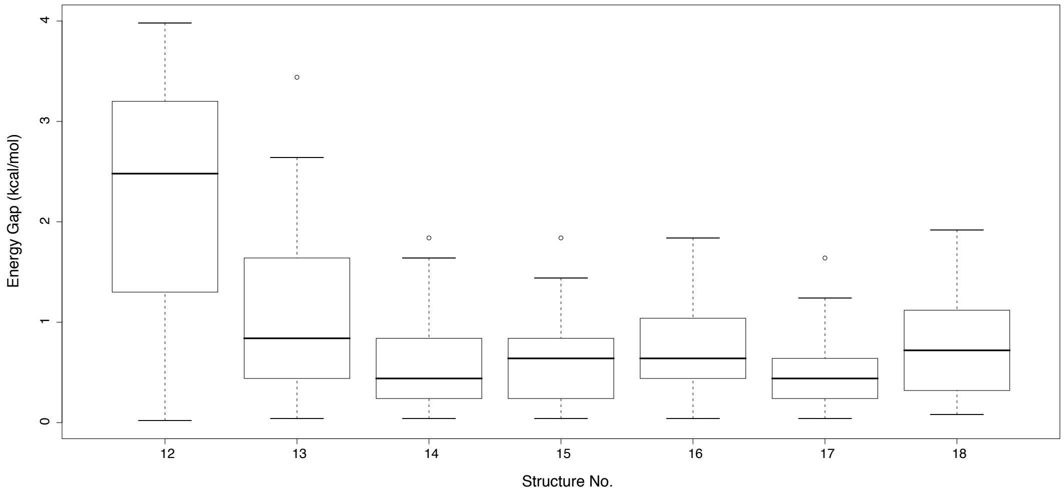
FIGURE 5.10: The MFE and energy gap for the two-states molecules of DS-MS dataset (cf. Tab. 5.1). The bivariate boxplots are represented as coloured region where the inner *bag* (dark blue) is comprised of 50% of the sequences from *StochSrchMulti*. The lighter blue region is a fence that separate inliers from outliers (equivalent to the whiskers of the univariate boxplot) representing the maximum and minimum data values from sequences generated by *StochSrchMulti*. Inliers and outliers are indicated by red lines. The depth median i.e., the point with highest halfspace depth is indicated by a ($*$) and yellow region. The ($\odot$) in the figure denotes sequences from *multiSrch*. In each panel, the lower left corner represents a sequence with low free energy and low E_{gap}. Most sequences from *multiSrch* fall within that area. See text for details.

from outliers and indicates the maximum and minimum data points equivalent to the whiskers in a univariate boxplot (light blue region). The depth median is represented by an asterisk ($*$) with or without a yellow area. The yellow area is a representation of the data points that are close to the depth median. The bagplot highlights the depth medium, the dispersion of data (size of *bag*), its correlation (indicated by the orientation of the *bag*, skewness (shape of the coloured regions) and its tails (points at the boundary of the *loops* and outliers). Inliers and outliers are indicated by the red lines. Inliers are position inside the fence, while outliers are position outside the fence.

The blue shaded regions of the graph are a representative of the sequences generated by *StochSrchMulti*. We then plot the data points of the sequences from the deterministic algorithm as ($\odot$). The lower left corner of each bagplots represents sequences with low free energy and low E_{gap}. Except for structures 5 and 7, sequences generated by *multiSrch* fall in the lower left corner. Generally the sequences of *multiSrch* (represented by $\odot$) are better than those generated by *StochSrchMulti*. An exception to this are the outliers in the plot of structure 10, where the outliers are of comparable in quality to the *multiSrch* sequences. This indicates that sequences from *multiSrch* more easily adhere to low energy gap. For structure 5, *StochSrchMulti* generates sequences with lower free energy from those generated by *multiSrch*, which have only a small E_{gap}. The opposite is observed in structure 7. Sequences from *multiSrch* have lower free energy but higher E_{gap} as compared to the sequences from *StochSrchMulti*. These two cases (structure 5 and 7) highlight the trade off between generating sequences with low E_{gap} and at the same time a low free energy, which are defined as two different terms in the objective function.

The E_{gap} of candidate sequences generated by *multiSrch* for molecules 12–18 in the dataset DS-MS (cf. Tab. 5.1) remain within 1.5 kcal/mol (Fig. 5.11), despite the increase in the number of meta-stable states from two to four. Both sequence design algorithms managed to generate sequences with an E_{gap} close to zero, as indicated by the minimum data points of *StochSrchMulti* and *multiSrch*. Compared to *StochSrchMulti*, the variance for sequences generated by *multiSrch* is lower. This is evident specifically for structures 13 to 16 where sets of sequences that differ in their base composition were generated with identical E_{gap} close to zero. Considering that the results of *StochSrchMulti* were generated from 50 of the best solutions and it was permitted to run five times longer to get to these solutions (i.e., larger numbers for refinement), the result of *multiSrch* in this case is considerably better.

In Fig. 5.12, we can find data points representing sequences with low free energy and low E_{gap} (lower left corner of the bagplot) only for structure 12. The remaining bagplots either have a low E_{gap} with a high free energy or low free energy but with a high E_{gap}. Sequences generated by *multiSrch* for structure 13, 14, 16 to 18 have a low E_{gap} and sequences generated by *StochSrchMulti* for structure 15 and 18 have a low E_{gap}. However the E_{gap} difference for sequences with low free energy in structure 13 to 18 is



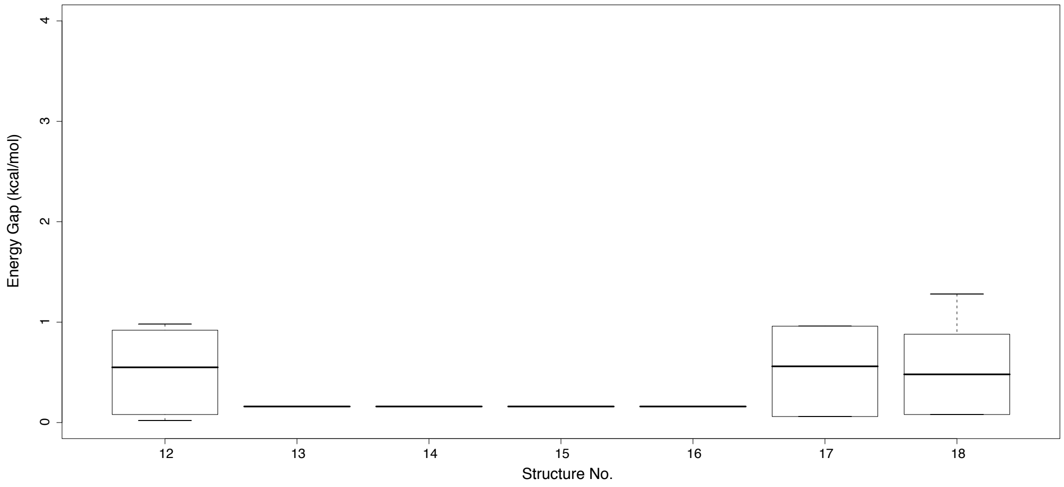


Figure 5.11: Variance of energy gap (E_{gap}) for the design of self-induced switching molecules with four states for the DS-MS dataset (cf. Tab. 5.1). The boxplot for *StochSrchMulti* is presented first (top), followed by the data for the sequences generated by *multiSrch* (bottom). Both design algorithms managed to generate sequences with low energy gap E_{gap}, as indicated by the minimum whiskers. For structures number 13 to 16, sequences with different base compositions generated by *multiSrch* have the same E_{gap}. The variance of the energy gap for sequences generated by *StochSrchMulti* is higher (E_{gap} < 4 kcal/mol) for each structure type compared to sequences generated by *multiSrch* (E_{gap} < 1.5 kcal/mol) Different from the E_{gap} calculation for the two states molecule, the E_{gap} value is equal to the summation of the complete subtraction combination between the four states (cf. Eq. 5.6).

small (E_{gap} < 1.0 kcal/mol). If we compare only the free energy, then the sequences generated by *StochSrchMulti* have the lowest free energy for structure 13 to 15, 17 and 18. This highlights the pre-mature pruning of base assignment in the *multiSrch* algorithm identified already in the test with two multi-stable states. As a consequence *multiSrch* has difficulty with balancing MFE and E_{gap} for DS-MS dataset. As discussed in Sec. 5.3 (p. 104), only a few natural occurring self-induced switching RNAs are known, and most

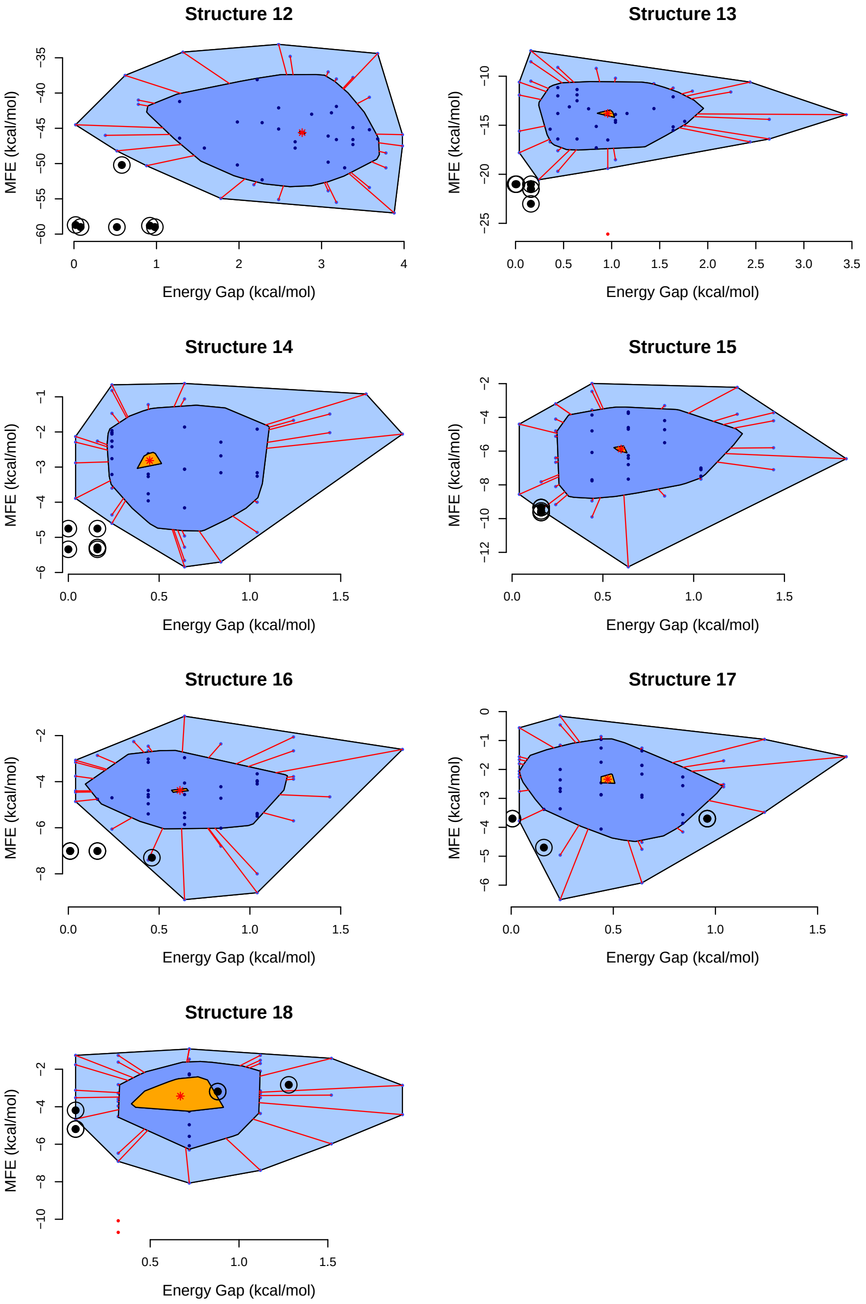


FIGURE 5.12: The MFE and energy gap for the four states molecules in the DS-MS dataset (cf. Tab. 5.1). For each panel, the lower left corner represents an area where a low free energy and a low E_{gap} sequence is obtained. Only for structure 12 there are data points in the lower left corner belonging to sequences generated by *multiSrch*. The remainder of the plots either have a low E_{gap} with higher free energy or a low free energy with a higher E_{gap}. Refer to Fig. 5.10 and text for further details.

are to long or to complex to be relevant for our design space. To evaluate the performance of *multiSrch* on a sufficiently large dataset DS-MS also includes molecular logic gates that have the advantage to have experimentally verified meta-stable conformations. However, these logic gates require the introduction of effector molecules in order to switch among their meta-stable conformations. To include these logic gate into the dataset for self-induced switches (TSS), we have substituted the external base pairing regions of the molecules with loops, thus leaving the majority of the meta-stable conformations to be loosely paired. For instance, structure number 12 to 16 that correspond to the nucleic acid AND gate would have a loosely paired meta-stable conformations because there exist two stretches of unpaired positions that initially (in trans-acting switches) should represent external base pairing formation with two effector molecules (cf. Fig. 2.8 for an example).

The lack of external base pairs has a direct affect for our deterministic algorithm. Premature pruning of the possible base combination can occur, and this lowers the ability of the algorithm to produce sequences with lower free energy. The majority of the unpaired base positions are assigned with bases `A` or `U`, in an attempt to balance the free energy of each conformation. If one increases the probability of finding candidates with lower free energy, then the E_{gap} value would increase because of the loose pairing that usually would be bound with an effector. These conformations with loosely pairing are unlikely to form because the formation individual base pairs in RNA folding happens rapidly (Tinoco and Bustamante, 1999). Because our calculation of free energy for secondary structure folding is based on the nearest-neighbour model (cf. Sec. 2.4.1), the free energy of base pair formation can be overwhelmed by the penalties the model applies to the unpaired bases. In our attempt of simulating these loosely paired conformation using *RNAsubopt* and *RNAeval*, we failed to find any significant meta-stable states that could represent the loosely-paired conformations.

The majority of these loosely paired conformations would yield significantly higher free energy (for instance, by definition, the free energy of an unpaired sequence is assumed to be 0.00 kcal/mol) and the E_{gap} between a meta-stable state where the majority of the base pairs are present against these loosely paired states can easily becomes large. Similar to *multiSrch*, the *StochSrchMulti* tends to generate sequences with a slightly higher free energy to compromise for the increase of E_{gap} value. The comparison study for the self-induced switching molecule is intended to test the generality of our deterministic algorithm (*multiSrch*). The results show that the *multiSrch* algorithm managed to perform much better than *StochSrchMulti*. The results generated by *multiSrch* are well within the depth median of the candidates generated by *StochSrchMulti* (cf. bagplot Fig. 5.10 and 5.12). Most of the time, *multiSrch* generates candidates with significantly smaller E_{gap} and lower free energy.

The trans-acting switch test case (TTS), are important as they closely resembles the type of designs intended for nucleic acid information processing. Each of the molecules has multi-stable conformations that switch when a trigger molecule is introduced. In order to evaluate the quality of sequences generated by *multiSrch* for trans-acting switches, we compiled a new dataset (DS-LG) comprised of the 14 nucleic acid logic gates from the previous DS-MS dataset. The complete meta-stable conformations (including external base pairing) are included in the DS-LG dataset instead of the loosely-paired conformation used in the DS-MS dataset. The randomised *StochSrchMulti* algorithm is limited to design only a single multi-stable state molecule. Therefore, in order to conduct the evaluation, the results generated by *multiSrch* are directly compared with the actual experimental sequences from publications listed in Tab. 5.1. As discussed in Section 5.2, the *multiSrch* algorithm has been developed to generate sequences not only for multiple multi-stable molecules, but also the external base pairing formation that might occur between these molecules. Ideally, one would be able to construct a network of information processing units, consisting of computational nucleic acid units by running *multiSrch*. To support this, *multiSrch* must be able to handle the design of a single computational unit with two or three effector molecules effectively.

Table 5.4: The thermodynamic characteristics of nucleic acids molecular gates, simulated by *RNAup* (Mückstein et al., 2006) and the sequences generated by *multiSrch* with default parameter setting. The MFE value is representative of the inactive (OFF) conformation of the gates, while the Energy gap value is equal to the summation of the complete subtraction combination between the four states (cf. Eq. 5.6). Mean of MFE and energy gap over 100 samples with errors for $\sigma = 1$. All the gates in the DS-LG dataset have been implemented, see Tab. 5.1 for references.

Type	Simulated		*multiSrch*	
	MFE	Energy Gap	MFE	Energy Gap
RNA_PASS_1	-35.80	8.60	-68.84±0.98	12.75±0.29
RNA_PASS_2	-26.97	8.94	-53.66±0.22	6.87±0.58
RNA_NOT	-37.77	6.74	-76.11±0.42	16.71±0.38
RNA_AND	-34.58	17.26	-66.19±0.18	16.32±0.70
RNA_OR	-53.56	17.45	-91.76±0.21	13.18±0.35
DNA_PASS_1	-12.82	0.63	-46.16±1.50	13.68±0.35
DNA_PASS_2	-12.80	0.84	-41.95±1.16	13.81±0.61
DNA_NOT	-16.33	2.41	-36.68±0.82	12.42±0.86
DNA_AND_1	-25.06	3.23	-91.96±0.22	17.22±0.96
DNA_AND_2	-19.01	0.54	-85.73±0.27	17.80±0.84
DNA_AND_3	-26.10	0.98	-100.40±0.29	19.89±0.66
DNA_AND_4	-28.07	4.90	-83.64±0.23	20.78±0.67
DNA_MULTI_1	-13.05	0.59	-80.41±0.34	25.85±0.69
DNA_MULTI_2	-42.36	3.28	-115.81±0.20	17.05±2.41

For trans-acting structural switching molecules, the E_{gap} value is optimised to be within a certain range, suited for the unfolding of existing base pairs and the formation of external base pair with the effector molecule(s). In order to test the two objective functions (Eqs. 5.7 and 5.8 p. 102,102), we conduct two separate runs. For the first run, we kept the same parameter setting of *multiSrch* as described in Tab. 5.3 except for the size of the list of suboptimal candidates (Λ) from 50 to 100 and the target value for E_{gap} in the objective function (cf. Eq. 5.7) from 0.00 kcal/mol to 6.00 kcal/mol. The latter is the minimum of the range suggested by Penchovsky and Breaker (2005). We examined both the sequence and conformation characteristics of the 14 nucleic acid logic gates and compiled their thermodynamic properties in Tab. 5.4.

From the table, note the significant decrease in free energy between the published sequences and the sequences generated by *multiSrch*. Because the generated sequences are gradually built, during the early phase base assignment, the efficiency for external base pairing (ΔG_{int}) is too small to affect the calculation of the objective function (Eq. 5.7). Therefore the algorithm tends to optimise the free energy term, and thus discards potential candidates with high free energies. Only after a number of iterations, the regions that formed external base pairs will start to affect the value of the objective function. It is likely that at this stage, the base combinations that had previously been selected have already contributed to a low free energy, resulting in the significant decrease in free energy observed in Tab. 5.4. Accordingly, if the free energy of a meta-stable conformation is already low, then because of the dependency of the "intersection" regions, the E_{gap} for the subsequent meta-stable conformation might increase. Although anything between 10-15 kcal/mol remains within range recommended by Penchovsky and Breaker (2005). The energy gap is used for the objective function instead of the energy barrier among the meta-stable states because the simulation of a complete kinetic folding (Pair Kinetics and Helix Kinetic programs using Monte-Carlo simulation as suggested by Higgs (1993); Morgan and Higgs (1998)) can increase the CPU time considerably.

Fig. 5.13 shows the free energy and energy gap of 100 candidate sequences generated by *multiSrch* for the nucleic acid logic gates specified in the DS-LG dataset. For most of the structures, *multiSrch* managed to generate sequences with E_{gap} in the range of 6–15 kcal/mol, which is within the range recommended by Penchovsky and Breaker (2005). However for RNA_NOT, DNA_AND_1, DNA_AND_2, DNA_AND_3, DNA_AND_4, DNA_MULTI_1, and DNA_MULTI_2 the value of E_{gap} exceeds 20 kcal/-mol. The free energy of published sequences simulated following (*RNAup*, Mückstein et al. (2006)) is significantly higher than then the free energy of the generated sequence, and this directly effected the E_{gap} value. Outliers in RNA_AND and RNA_OR resulted in the increase of E_{gap} to more than 20 kcal/mol. The quality of sequences for logic gates with a single effector molecule (RNA_PASS_1, RNA_PASS_2 and DNA_PASS_1, and DNA_PASS_2) is better that any logic gates that require two effectors. For logic gates with two effectors, the quality of the generated sequences varies. But for each panel, we observed that there are a number of inliers with low free energy and low E_{gap}. The bag for the logic gates with two effector molecules however, covers the full range of E_{gap}.

For the second run, we substituted the objective function of *multiSrch* with Eq. 5.8 (p. 102) and changed the size of the pool of suboptimal candidates (Λ) from 100 to 50. In addition to the E_{gap}, it is necessary to supply the target value of ΔG_{opt}. In order to assign the suitable target values, starting from the rounded simulated values of the published sequence (cf. Tab. 5.4) for free energy and E_{gap}, we conduct trial runs with the size of suboptimal candidates (Λ) set to 5 and Eq. 5.8 as objective function, we gradually increase by 1 kcal/mol the value of ΔG_{opt} and E_{gap} to 5 kcal/mol, and also

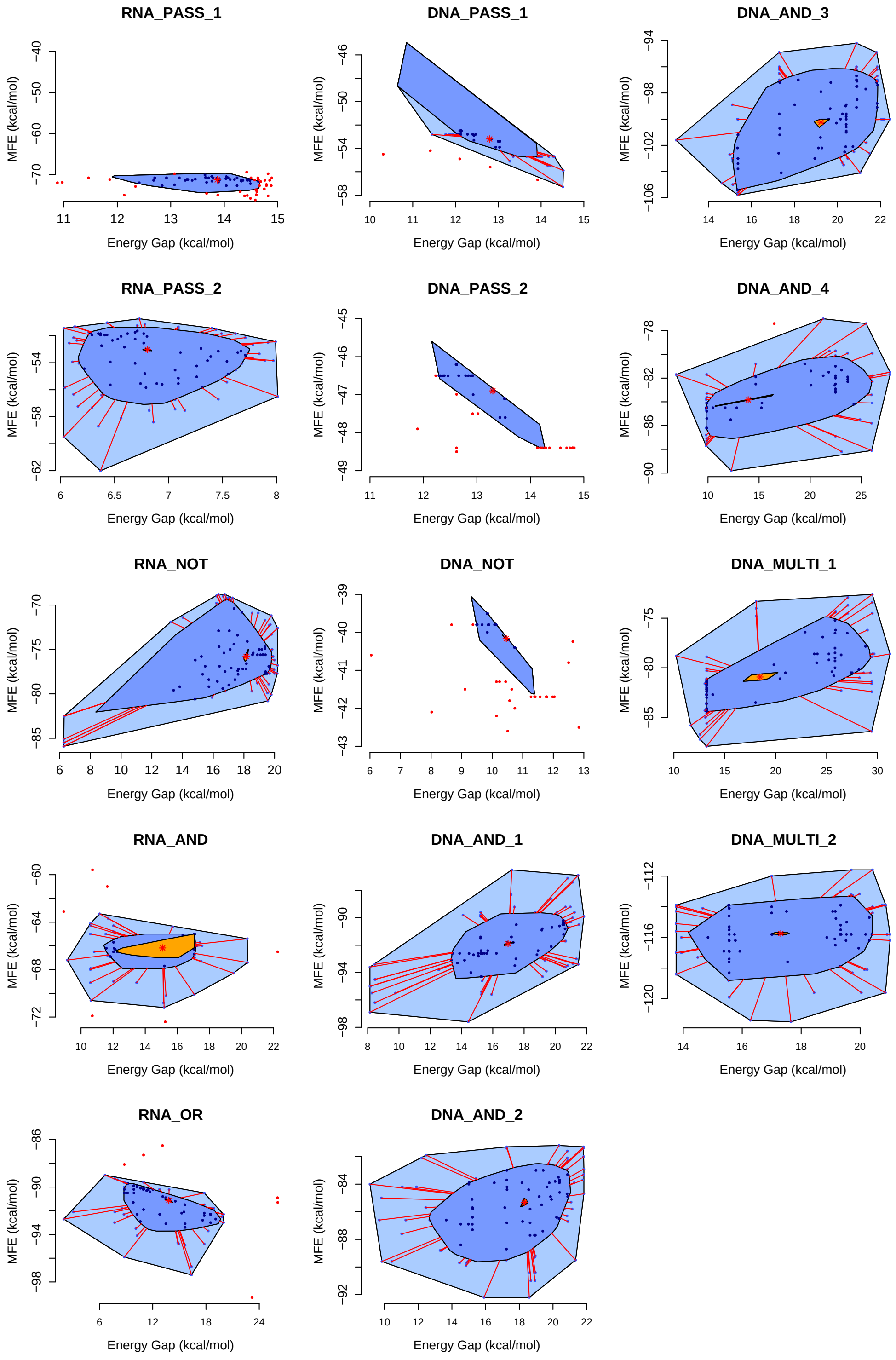


Figure 5.13: The MFE value and energy gap for sequences generated by *multiSrch* for the trans-acting molecules (DS-LG dataset). The DS-LG comprises 14 nucleic acids logic gates from Tab. 5.1 with effector molecule(s) and external base pairings included to the meta-stable target conformations. A set of 100 sequences are generated from *multiSrch* for each logic gate. The quality in terms of E_{gap} for logic gates with single effector are better than the logic gates with two effectors. However, a significantly lower free energy is observed for all logic gates when compared to the simulated value of the published sequences in Tab. 5.4. PASS and NOT gates are two-states gates, AND and OR gates are both four-states gate while DNA_MULTI_1 is equivalent to the $i_1 \wedge i_2 \neg i_3$ gate and DNA_MULTI 2 is equivalent to $i_1 \wedge i_2 \wedge \neg i_3$ gate respectively.

Table 5.5: The thermodynamic characteristics for sequences generated by *multiSrch* using the objective function Eq. 5.8 for the DS-LG dataset. The target values of both ΔG_{opt} and E_{gap} are approximation of the simulated (Mückstein et al., 2006) values from published sequences (cf. Tab. 5.4). Mean of MFE and energy gap over 50 samples with errors for $\sigma = 1$

Type	Parameter Value		*multiSrch*	
	ΔG_{opt}	E_{gap}	MFE	Energy Gap
RNA_PASS 1	-35.00	8.00	-36.46±0.53	10.38±1.08
RNA_PASS 2	-28.00	8.00	-29.23±0.78	10.78±0.73
RNA_NOT	-40.00	6.00	-42.22±0.95	8.16±0.23
RNA_AND	-35.00	17.00	-35.15±0.33	17.58±0.11
RNA_OR	-55.00	17.00	-57.06±1.49	18.57±0.34
DNA_PASS 1	-13.50	1.00	-14.07±0.07	1.00±0.07
DNA_PASS 2	-13.00	1.00	-14.41±0.83	1.19±0.62
DNA_NOT	-16.00	2.00	-17.19±0.52	2.03±0.05
DNA_AND 1	-30.00	3.50	-33.24±0.07	3.99±0.19
DNA_AND 2	-20.00	0.50	-21.39±0.08	0.83±0.09
DNA_AND 3	-27.00	1.00	-27.81±0.05	0.98±0.05
DNA_AND 4	-28.00	5.00	-30.97±0.31	5.43±0.10
DNA_MULTI 1	-15.00	0.50	-18.27±0.29	0.74±0.10
DNA_MULTI 2	-45.00	3.50	-45.97±0.44	3.90±0.10

decrease these value by -1 kcal/mol until -5 kcal/mol. Based on the results of these trial runs, we select the target values for ΔG_{opt} and E_{gap} from the runs where the best set of sequences were generated. These values are listed in Tab. 5.5. The assignment of the target value for ΔG_{opt} and E_{gap} are intended to produce sequences that are closer in term of free energy and energy gap to the simulated values of the published sequences.

The result for the second run with objective function 5.8 is listed in Tab. 5.5. When the target value for ΔG_{opt} and E_{gap} are specified *multiSrch* managed to generate results that are much closer in terms of MFE and E_{gap} to the published sequences. These improvements are significant when compared with the initial results listed in Tab. 5.4. In the default setting, the algorithm would search for the lowest free energy sequences that conform to the specified energy gap value, and if such sequences exist, then it would also be possible to find sequences with higher free energy that still adhere to the energy gap target value. These sequences with higher free energy and marginally energy gap are readily available because *multiSrch* keeps any suboptimal candidates that are within ±5.00 kcal/mol of the minimal energy gap value.

If one desires only sequences that comply to a certain minimum value of the objective function (e.g. KEEP $= \Xi(x) - min(\Xi(x)) = +2.0$), then we can directly restrict the suboptimal list to include only those candidates. By specifying the exact values for the ΔG_{opt} and E_{gap} terms, the time required by the *multiSrch* algorithm would be further reduced because the amount of pruning for the base configuration branches has been increased. Any base combinations that already exceeds the amount of free energy and energy gap can be omitted. The specification of the exact values of the ΔG_{opt} and E_{gap} terms also allows one to increase the number of suboptimal candidates (Λ), because there are less branches to be evaluated as the number of pruning increase.

In order to investigate the quality in terms of MFE and E_{gap} for the generated sequences, we plotted the MFE and E_{gap} for each structure (Fig. 5.14). The boundary lines depicted in the bagplots correspond to the parameter setting for ΔG_{opt} and E_{gap}. As, both sequences with positive or negative deviation from the target value are kept in the suboptimal list, it is desirable to have a bagplot that sits in the middle of the crossing between the two dashed lines. From Fig. 5.14, we observed that the majority of the plots tend to skewed to a larger energy gap. These bagplots are representative of the condition where the value of E_{gap} shifted in order to compensate for the optimisation to lower the free energy (discussed in p. 114). This representation is accumulative for all twenty candidates in the sequence list. The sequences close to the intersection of the dashed-lines are of the best solutions.

Another observation that can be made from the figure is for DNA_PASS 1, DNA_NOT and DNA_AND 3, where all three bagplots sit in the middle of the E_{gap}-axis, but below the MFE line. This is representative of a case where reaching the target free energy is compromised in order to achieved a smaller E_{gap}. In general, for the case where the target MFE is achieved while compromising the target E_{gap} and the case where the target E_{gap} is achieved while compromising the target MFE, all sequences generated are very close to the desired target values for both MFE and E_{gap}. In both cases, we can improve the quality of the sequences by increasing or decreasing the parameter values. This enables the construction of a sequence library consisting of candidate sequences within a specific parameter range that might be a part of a homogeneous network of sequences, where the conserved regions are retained with minor mutations to the remaining position (Ancel and Fontana, 2000) or unique (different sets of sequences that fold to the same target conformation).

Earlier in the chapter, we introduced two types of molecular switching models related to our task of developing a multi-stable sequence designer. We then introduced and discussed the implementation of the optimisation algorithm *StochSrchMulti* that can generate sequences for the design of RNA molecules with multi-stable conformations. By reviewing the current design space and the type of molecules for the design of nucleic acids for information processing tasks, we developed a deterministic approach to tackle the sequence design problem for multi-stable conformations. This deterministic approach is aimed not only at single multi-stable conformation molecule, but also the interaction among these multi-stable states molecule. Using dynamic programming, we are able to inspect all possible base combinations belonging to each pathway element. The pathway elements describe the complete dependency relationship of both internal and external base pairing between the interacting molecules and their meta-stable conformations. We

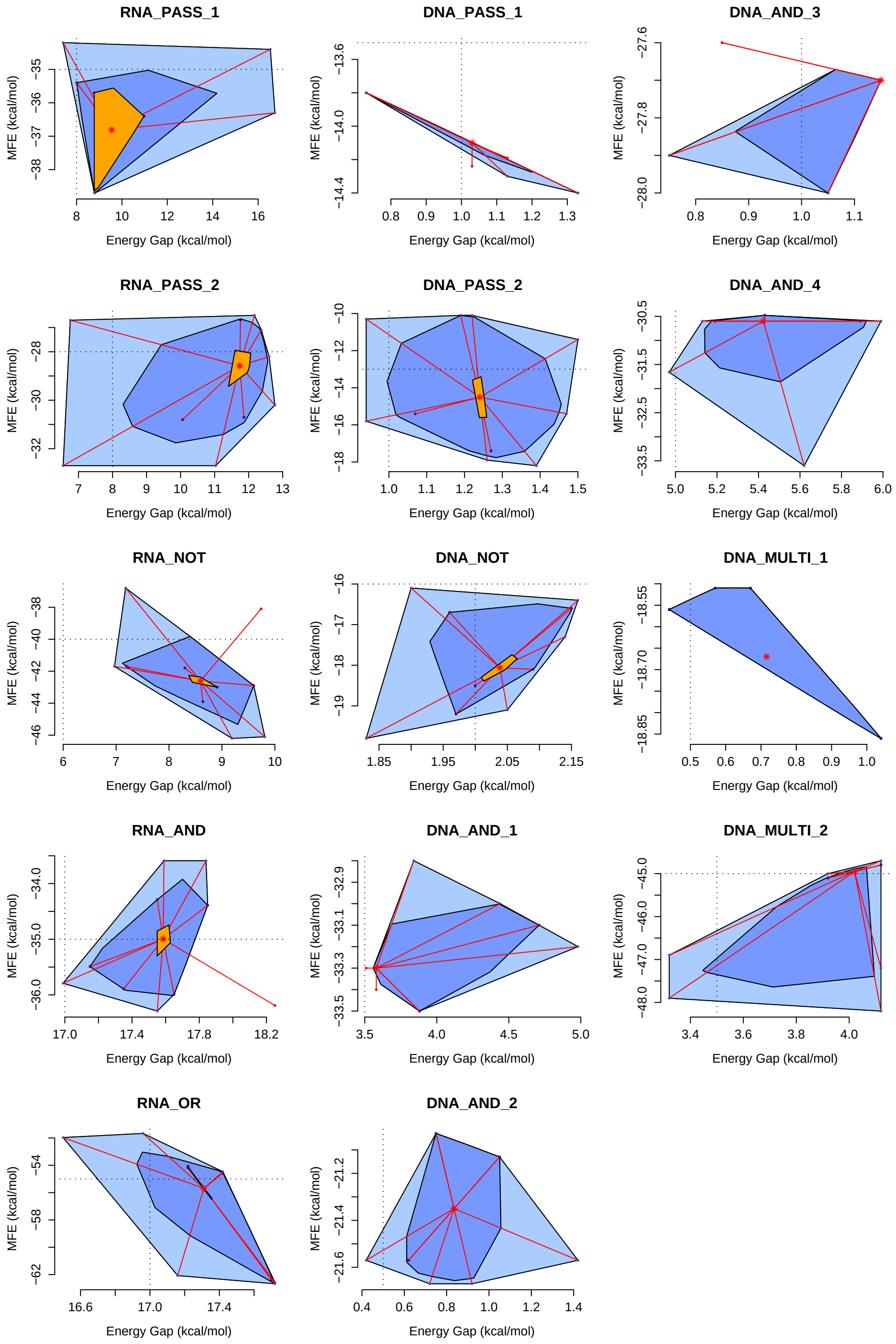


FIGURE 5.14: The MFE and energy gap for sequences generated by *multiSrch*, running with parameter settings tuned to the DS-LG dataset. In each bagplot, the two dash-lines correspond to the parameter setting for ΔG_{opt} (horizontal line) and E_{gap} (vertical line). Bagplot missing the horizontal line (i.e., DNA_AND 1, DNA_AND 2, DNA_AND 3, DNA_AND 4, DNA_AND 5 and DNA_MULTI 1) sit below the ΔG_{opt} value. The illustration of the horizontal line in this cases are excluded to allow for a larger scale of the bagplot itself.

then developed a deterministic algorithm called *multiSrch* and evaluated its performance for both types of switching models. The results indicate that, for the design space of interest, *multiSrch* is a comparatively fast method of generating accurate sequences with desired energy profile.

Chapter 6

Building Computational Nucleic Acids for Molecular Computing

6.1 Computational Design of Molecular PASS Gate

In nature protein molecules which are responsible for carrying out vital cellular functions combine self-assembly and conformational dynamics to achieve their function. A complex three-dimensional structure of the folded protein is required in order to support this function. The structural variability of protein underlies its ability in mediating almost all functions in living cells. Conformation based computing using protein enzymes has been investigated by Zauner and Conrad (2000, 2001). In recent years, with the discovery of small regulatory RNA (sRNAs) (Zamore and Haley, 2005) and the discovery of the catalytic ability of ribozymes (Altman, 1990), RNA emerges as more versatile than previously perceived. Like protein, RNA structure determines its function, but unlike protein, the secondary structure of RNA molecule provides more information regarding its tertiary folding. This allows for the construction of RNA computing units considering only the secondary structure. In contrast to protein, there exist well established computational tools that can aid in the secondary structure prediction and sequence design of RNA molecules.

By combining *RNAfold* for secondary structure prediction, *RNAinverse* for sequence design, *Kinfold* for simulating the kinetic pathway of a secondary structure folding for an RNA sequence, and *RNAcofold* for measuring the efficiency of intermolecular binding (from the Vienna RNA package of Hofacker et al. (1994)), Penchovsky and Breaker (2005) derived a computational protocol to construct RNA logic gates using the allosterically controlled ribozymes architecture. The protocol of Penchovsky and Breaker (2005) however, imposes strict structural constraints during the design process. These structural constraints limit the ability of the protocol to produce different structural designs. In this section, we discuss improvements that are made on Penchovsky and

Breaker (2005) protocol's, in order to enhance its functionality in exploring the design space to generate a diverse set of RNA logic gates. The difference in terms of quality between the revised protocol and the original by Penchovsky and Breaker (2005) is then evaluated by designing a simple PASS gate.

The simplest logic gates have only one bit input. The NOT gate that inverts the input and the PASS gate (sometimes also called "identity" or YES gate) that forwards the input signal. Although, from a purely logic viewpoint PASS gates serve no purpose, in practise they can reform a degraded signal or adjust signal delay (Zauner, 2005b). The molecular pass gates considered in this thesis are more powerful than one-bit logic gates and an essentially arbitrary input sequence of limited length can be recoded into a different output sequence.

The work-flow employed by Penchovsky and Breaker (2005) to construct an RNA PASS gate starts with a design for an approximately 80 nucleotides long RNA molecule which contains a highly sensitive hammerhead ribozyme in its sequence. The bases for the sequence of the molecule are fixed (i.e., this includes the conserved base of the hammerhead ribozyme), except for a region about 10–20 nucleotides long which, however, is crucial for maintaining the active hammerhead conformation. If this region can participate in internal hybridisation the molecule will undergo a large change in conformation to a catalytically inactive state (i.e., with the minimal functional structure of hammerhead ribozyme in Fig. 2.4 distorted). The binding of an effector oligonucleotide to this region will prevent internal hybridisation and thus stabilise the catalytically active conformation of the hammerhead ribozyme. Accordingly, the region acts as an oligonucleotide binding site (OBS, cf. Section 2.3) that exerts allosteric control over the catalytic activity of the ribozyme.

The aim is to design the sequence for this OBS such that it is likely to allow the switching between the active and inactive state in a real RNA molecule. This is achieved by first selecting a candidate sequence for the OBS and inserting it into the fixed sequence of the sensitive hammerhead ribozyme. The sequence is generated by randomly assigning bases to the positions in the sequence while obeying the constraint that no more than three identical consecutive nucleotides can be present in these positions (first row of Tab. 6.1). To judge the plausibility for this RNA sequence design to be practicable and likely to be operative if implemented as a real RNA molecule Penchovsky and Breaker (2005) introduced a filter cascade, the steps of which are summarised in rows two to six in Tab. 6.1.

If a generated sequence passes these five filter steps it is taken as a model design for the secondary structure of the OBS region in the desired gate. This model is specified by the complete secondary structure and a partial sequence which commits to all bases except those located in the the OBS region. By repeatedly running *RNAinverse* with this specification one obtains a set of complete sequences for the logic gate which differ

Table 6.1: Constraints imposed on candidate sequences following (Penchovsky and Breaker, 2005).

Stage	Filter	Condition to satisfy
1	Identical nucleotides	No more than three identical consecutive nucleotides in the oligonucleotide binding site(s)
2	Active state conformation	The formation of an active hammerhead conformation based on the truth table condition (cf. Sec. 6.4 p.128)
3	Base-pairing percentage	In the absence of effector(s) 30%–70% of the oligonucleotide binding region is hybridised
4	Energy gap	Energy gap between the inactive and active state is within -6 kcal/mol to -10 kcal/mol
5	Temperature tolerance	Structure is preserved over a temperature range of 20°– 40°C
6	Ensemble diversity	For neither active nor inactive state the ensemble diversity (cf. (Penchovsky and Ackermann, 2003)) exceeds 9 units
7	Folding efficiency	The RNA molecule must fold, in the absence of the effector, to the inactive conformation within 480 units in *Kinfold* (Flamm et al., 2000).

from each other only in the OBS region. Note that the secondary structure of sequences generated by *RNAinverse* may not strictly conform to the specified conformation, but does not differ by more than two base pairs. Only sequences that have a thermodynamic stability comparable to the model design are maintained. The folding efficiency of these sequences is then verified (last row of Tab. 6.1).

Penchovsky and Breaker (2005) suggest a second stage of processing which derives from the sequences that have successfully passed the filter chain alternative sequences with similar folding and similar thermodynamic stability. In our simulations the generation of different OBS sequences of this second stage was only about 3% better than that of the first stage. The value of the second stage presumably lies in providing sequence alternatives to designs that are already favoured, (e.g., because they have been verified experimentally). In contrast to the highly constrained design protocol of Penchovsky and Breaker (2005) outlined above, we introduce a protocol for designing RNA gates that relaxes the constraints on the secondary structure of the inactive conformation a gate can assume. Yet, in general, the length of functional nucleotide sequences composed of four bases in arbitrary ordering gives rise to a combinatorically large design space in which a random search without appropriate constraints would not be efficient in generating useful designs. However, the design space spanned by the RNA-gate designs found in the literature can be used to narrow the search. Using the properties of DNA and RNA logic gates previously summarised in Tab. 4.2, we deduced a table of parameter ranges (Tab. 6.2) suitable for the design of molecular gates, as described below.

TABLE 6.2: Design space for computational nucleic acids derived from Tab. 4.2.

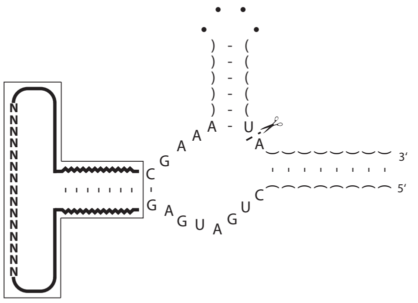
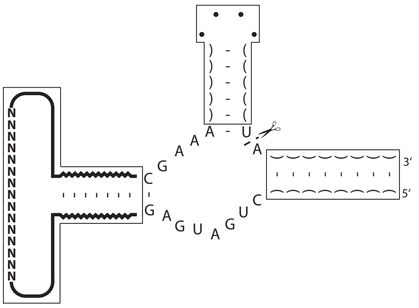
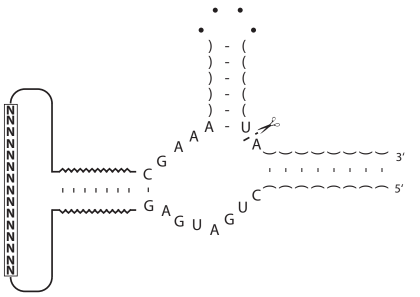
Type	Probability	Maximum no.	Length Range
Helix	0.50	-	4–15
Hairpin Loop	-	0–3	4–15
Internal Loop	0.45	0–3	2–8
Bulge	0.05	0–1	1–8
Junction	-	0–3	4–8
OBS	-	1	15–22
Linker	0.55	2	0–5

Our protocol for designing an allosterically controlled ribozyme comprises three generating steps, outlined in Tab. 6.3, to arrive at a sequence design. This protocol is referred to as P-ER1 for the remainder of the text. Sequence generation is followed by a series of validation steps. In the specific example illustrated for the design of allosterically controlled hammerhead ribozyme imitating the PASS logic operation (cf. left column in Tab. 6.3), in the first generating step the conformation for the catalytically active ribozyme is determined by specifying the secondary structure of an extension to the hammerhead core composed of helix II, two linkers and an OBS region. To this end for each position in the sequence its participation in internal hybridisation is selected by generating a dot-bracket representation for the secondary structure of the molecule. In generating the secondary structure constraints, derived from Tab. 4.2 and detailed in Tab. 6.2, are invoked by a generation algorithm that follows Andronescu et al. (2004).

The second generating step of P-ER1 assigns nucleotides to the positions in the sequence, except the OBS region. This assignment of the nucleotides adheres to the secondary structure generated in the first step. For this task, we selected our *StochSrch* program (cf. Chapter 4), where a rule based initialisation procedure is applied in order to design sequences that conform to the target structure. For the folding tests of the nucleotide assignment, the unassigned OBS region is set to a repetition of a hypothetical non-binding base (labelled `N`) as suggested by Penchovsky and Breaker (2005).

The hammerhead ribozyme can reliably be deactivated by binding to its conserved core-region and thus distorting its secondary structure (Tang and Breaker, 1997; Koizumi et al., 1999; Soukup and Breaker, 1999; Breaker, 2002). Therefore, in the third generating step of P-ER1, first a sequence complementary to the conserved `CUGAUGAG`-region of the hammerhead core is inserted at a random location within the two linkers and the OBS, (i.e., in the hairpin loop attached to helix II). Afterwards the remaining unassigned positions in the sequence are filled by drawing randomly from the four possible

Table 6.3: Proposed computational protocol (P-ER1) for designing allosterically controlled hammerhead ribozyme gates. In contrast to the method described in (Penchovsky and Breaker, 2005) the protocol starts with the conformation of the active ribozyme.

 A A A G C G A G U A G U C U A 3′ 5′ NNNNNNNNNNNNNNN	Design of the secondary structure of a potential gate based on a hammerhead core. First the structure (but not sequence) of the helix II and associated allosteric control domain is generated. This extension of the ribozyme is comprised of four parts: a helix that attaches to the ribozyme core, an effector binding region (NN· · · NN), and two linker sequences connecting binding site and helix. The lengths of the extension (23–62 nt) and the helix (4–15 nt) are chosen randomly. The remaining part of the structure (binding region and linkers) is filled with the constraints listed in Tab. 6.2.
 A A A G C G A G U A G U C U A 3′ 5′ NNNNNNNNNNNNNNN	Assign the conserved nucleotides of the hammerhead core. The remaining sequence positions except the binding site are assigned by searching for a base sequence that will fold into the structure designed in the previous step. This can be achieved with *RNAinverse from Vienna* (Hofacker et al., 1994), *RNAdesigner from RNAsoft* (Andronescu et al., 2003, 2004), or *INFORNA* (Busch and Backofen, 2006). However in our implementation we selected our *RepInit* program for this task. To arrive at the active conformation of the gate, a non-binding pseudo-base (N) is assigned to all positions in the binding region during the search process.
 A A A G C G A G U A G U C U A 3′ 5′ NNNNNNNNNNNNNNN	Replace the pseudo-bases in the binding region with real bases. This is done such that the structural elements initially generated for the linker and binding parts of the extension become manifest. This can be verified with *MFOLD* (Zuker et al., 1999) or *RNAfold* (Hofacker et al., 1994).

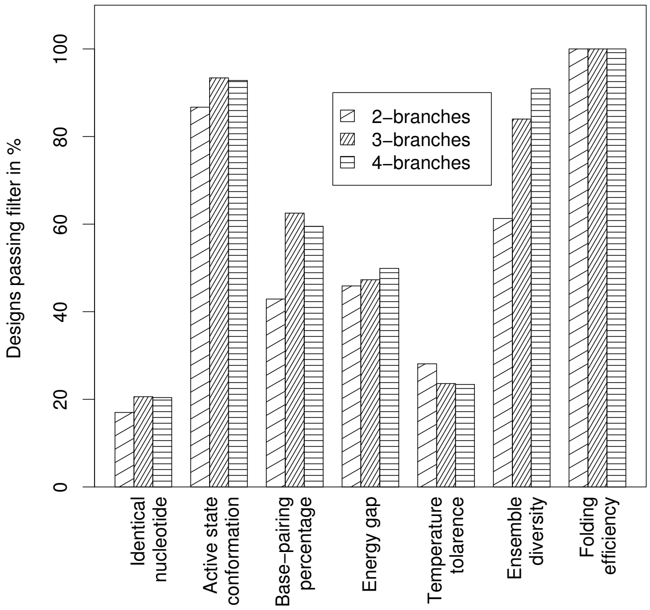


FIGURE 6.1: Permissiveness of each step in the filter chain of Tab. 6.1 when applied to generated candidate sequences for PASS gates. The filter stages are applied consecutively from lefty to right. For each step the sequences adhering to the filter condition is shown as percentage of input sequences supplied to this filter stage.

nucleotides {A,U,C,G}. The resulting sequence is likely to be inactive due to internal hybridisation of a section of the OBS, and possibly a few bases from a linker, to the hammerhead core. In the folding predictions a hybrid molecule composed of ribozyme and substrate is considered, consequently interference of the substrate with the OBS is unlikely.

The subsequent validation of the generated sequence designs involves all steps of the filter cascade in Tab. 6.1. This screening process prunes out 99% of the generated sequence candidates as depicted in Fig. 6.1. Starting with a pool of 50,000 candidates as input to the first filter stage, 586 designs passed the entire filter chain. A manual inspection of the dot-plot graph (Hofacker et al., 1994) for all 586 designs confirmed in every case that the conserved sequence region of the hammerhead core is blocked by hybridisation in the inactive conformation and free in the active conformation. To further evaluate the plausibility of the remaining computational designs we calculate the equilibrium constants for the three possible dimers that can form when ribozyme molecules and effector molecules interact. The calculation is based on the free-energy values provided by *RNAcofold* (Hofacker et al., 1994; Mückstein et al., 2006) and the assumption of a fixed, equal concentration for the monomeric ribozyme (R) and monomeric effector (E) (Bernhart et al., 2006). Any point in the area of the triangle depicted in Fig. 6.2 corresponds to a calculated (cf. Eq. 13 of (Schuster, 2006)) combination of the relative concentrations of the three possible dimers that can form (RR, EE, RE).

The interaction between two RNA molecules that formed complexes is dependent on the level of concentration of each molecule. As example, for hybridisation between the ribozyme (R) and the effector (E) molecule, the two monomers (R and E) can form three different complexes (RE, RR and EE). The partition function of monomers (R and E) and dimers (RE, RR and EE) are generated by the co-folding simulator, to facilitate the calculation of equilibrium constant between dimers and afterwards, the calculation of dimer concentration (Dimitrov and Zuker, 2004; Bernhart et al., 2006). Based on the equilibrium relation defined for the ribozyme and effector co-folding, R + E $\rightleftharpoons$ RE, RR, EE, R and E. In order to have stable hybridisation between ribozyme and substrate strands, the dimer concentration (given as [RE], [RR], and [EE]) should be as follows, [RE] $\gg$ [RR] and [EE].

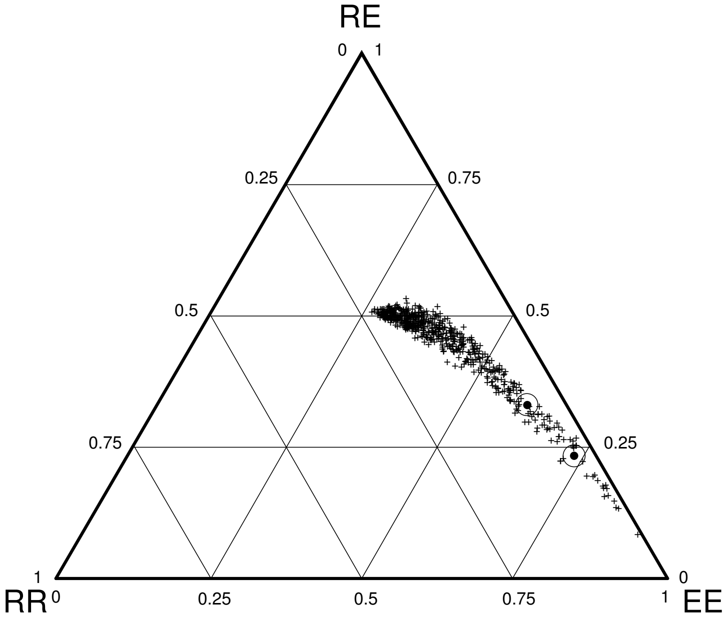


FIGURE 6.2: Estimated binding between ribozyme (R) and effector (E) for different PASS gate designs. The plot indicates the relative concentrations of the complexes formed from ribozyme and effector molecules calculated following (Bernhart et al., 2006) obtained for PASS gates that have been designed with the method of table 6.3 and evaluated with the filter chain in table 6.1 labelled as +, and the two experimentally validated designs from (Penchovsky and Breaker, 2005) labelled ⊙.

From this analysis it appears that the enlarged degrees of freedom in the design protocol (P-ER1) outlined in Tab. 6.3 can yield PASS-gate designs with good ribozyme-effector binding (RE). Note, however, that due to the lack of tools for simulating RNA-DNA hybridisation, the values shown for the designs from Penchovsky and Breaker (2005) have been calculated for an RNA effector, while in (Penchovsky and Breaker, 2005) an experimentally more convenient DNA effector molecule was applied. Samples of structures that were derived with the P-ER1 protocol outlined in table 6.3 and have passed the screening with the filter chain in table 6.1 are shown in Fig. 6.3. The three structures in panel A, B, and C are representative for the classes of molecules that have inactive conformations with 2-branches, 3-branches, and 4-branches, respectively. In each structure the oligonucleotide binding region for the effector molecule is indicated by a bold line section. The structure in panel A bears some resemblance to a design proposed by Porta and Lizardi (1995) (cf. Fig. 2.9A), while the structure in panel B is similar to the designs by Penchovsky and Breaker (2005).

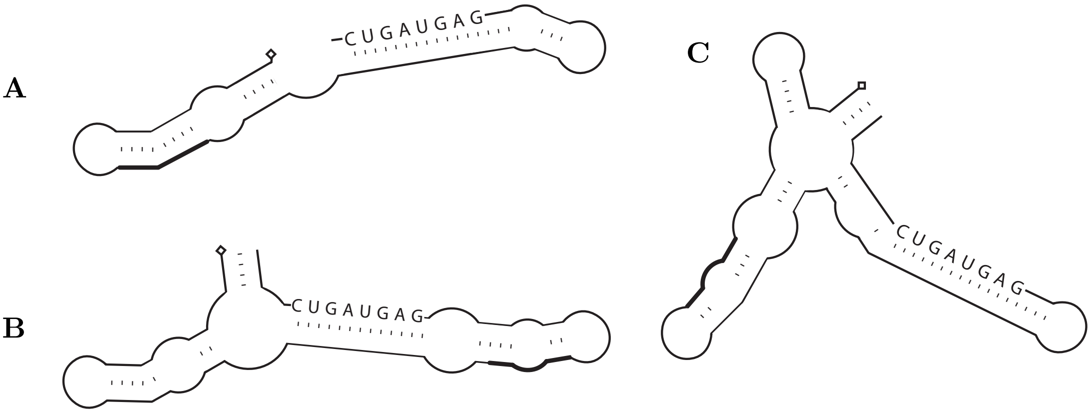


Figure 6.3: Inactive conformation of allosterically controlled hammerhead ribozymes designed to act as PASS gates.

Although our protocol (P-ER1) successfully increases the degree of freedom in designing the molecular PASS gate, however we acknowledge that the variant computational protocol is far from being efficient. It is evident in our approach that the majority (99%) of the candidate sequences corresponding to a suggested structure design are eliminated in the design stage. From Fig. 6.1, one can observe that the passing rates of the sequences are poor in four filter steps (i.e., identical nucleotide, base-pairing percentage, energy gap and temperature tolerance). From these four, the base-pairing percentage, energy gap, and the temperature tolerance can be addressed by *multiSrch* presented Sec. 5.2. The issue of identical nucleotides (i.e., a condition in which only three consecutive nucleotides are allowed in the candidate sequence) can be handled during sequence initialisation. Next, we investigate the ability of this computational protocol to design the complete set of binary logic gates.

6.2 Constructing the Complete Set of RNA Molecular Logic Gates

The construction of conventional logic gates as test case in evaluating the performance of computational protocol here, however, does not imply that logic gates are a viable strategy for implementing computational nucleic acids. Nevertheless, if a nucleic acid unit is to be constructed for the purpose of regulatory control, then, the design of binary logic operators used here as test case has direct application. In this section, we discuss the design of nucleic acid molecular gates that follow the logic of all possible two inputs conventional logic gates. The complete truth table of all the possible two input logic gates are listed in Tab. 6.4. In contrast, Fig. 6.4 illustrate the abstract representation of the nucleic acid logic operators.

Table 6.4: Two-input binary logic gates

Line	Input state					
A	1	0	1	0		
B	1	1	0	0		
Operation	Output state				Name	Symbolic
$A\circ_1 B$	0	0	0	0		0
$A\circ_2 B$	0	0	0	1	NOR	$A \diamond B$
$A\circ_3 B$	0	0	1	0		
$A\circ_4 B$	0	0	1	1	NOT B	$\neg B$
$A\circ_5 B$	0	1	0	0		
$A\circ_6 B$	0	1	0	1	NOT A	$\neg A$
$A\circ_7 B$	0	1	1	0	XOR	
$A\circ_8 B$	0	1	1	1	NAND	$A \mid B$
$A\circ_9 B$	1	0	0	0	AND	$A \wedge B$
$A\circ_{10} B$	1	0	0	1	Equivalence	$A \Leftrightarrow B$
$A\circ_{11} B$	1	0	1	0		A
$A\circ_{12} B$	1	0	1	1		
$A\circ_{13} B$	1	1	0	0		B
$A\circ_{14} B$	1	1	0	1	Implication	$A \Rightarrow B$
$A\circ_{15} B$	1	1	1	0	OR	$A \vee B$
$A\circ_{16} B$	1	1	1	1		1

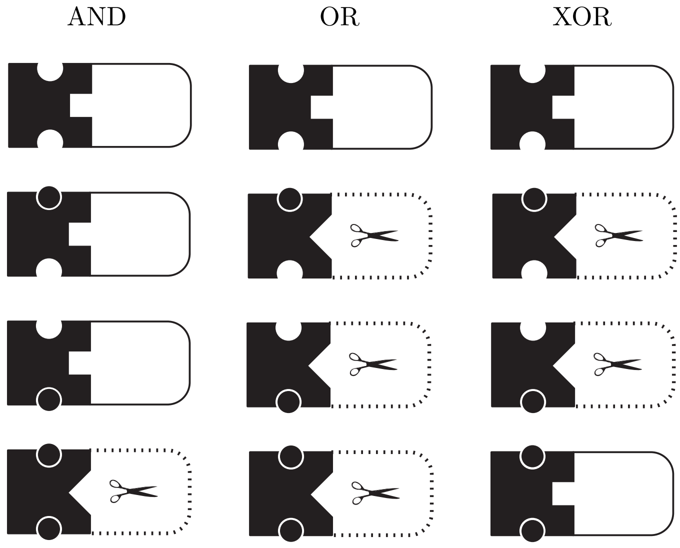


Figure 6.4: The abstract representation of RNA molecular logic gates with (AND, OR, and XOR) operations. The shift in the conformational dynamics is indicated by the change of the rectangle into a triangular shape and the change of the solid line curves into dotted lines. Catalytic activity is marked by the scissors symbol.

Table 6.5: NAND and NOR as universal operators in binary logic

	Equivalent	
Operation	NAND	NOR
$\neg$ A	A | A	A $\diamond$ B
A $\vee$ B	(A | B) | (A | B)	(A $\diamond$ A) $\diamond$ (B $\diamond$ B)
A $\wedge$ B	(A | A) | (B | B)	(A $\diamond$ B) $\diamond$ (A $\diamond$ B)
A $\Rightarrow$ B	A | (A | B)	(B $\diamond$ (A $\diamond$ B)) $\diamond$ (B $\diamond$ (A $\diamond$ B))

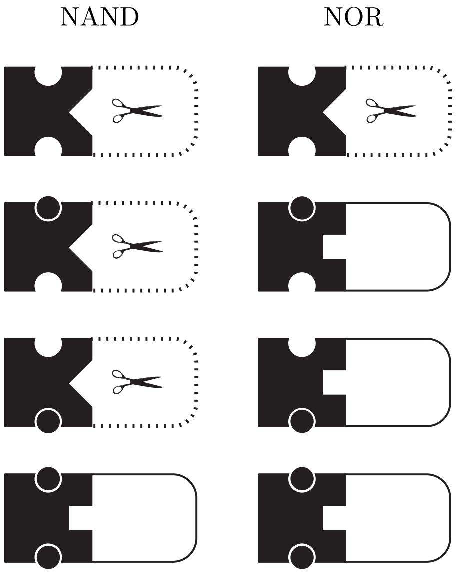
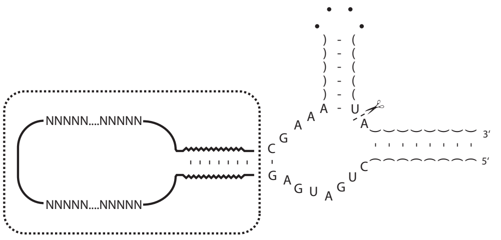
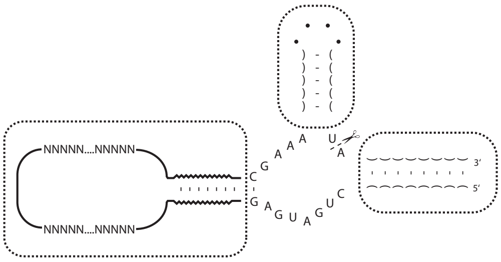
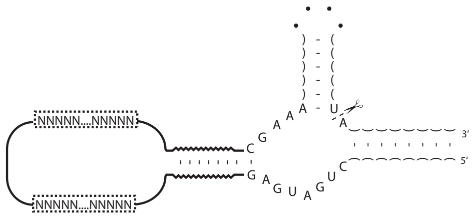


Figure 6.5: The NAND and NOR universal operators as molecular logic gates. See Fig. 6.4 for explanation.

Instead of the binary operation of conventional logic gates, biomolecules offer richer operations. For instance, a molecular gate can be made from an allosterically controlled hammerhead ribozyme, where one can attach different substrate strands to be released as output sequence, and design the receptor sites with different effector molecules as inputs. The substrate is cleaved when the ribozyme is activated, while the effector binds to the receptor site to steer conformation change that activates the ribozyme. However, it is possible to design a substrate strand that can function as effector molecule for another allosterically controlled ribozyme. For instance in the design of a cascade of nucleic acid computers. The substrate and effector molecules may have different sequences, therefore an RNA AND gate, although it follows the logic AND operation (i.e., only releases its output when both effector molecules are present) does not directly correspond to a conventional AND gate. The one-bit input and output signals of the conventional logic gates are represented by essentially arbitrary nucleotide sequences. In principle, there are 4^n base combinations for the input and output molecules, where n denotes the length of the signalling molecule. Two or more molecular gates that are common in their activation mechanism could be completely different in term of their structural design and mechanism (cf. Fig. 2.9). If a cascade of logic gates is to be developed, the dynamics of the molecular logic gates not only allows for the use of output strands as effectors but also as substrates in subsequent processing stages.

Table 6.6: Proposed computational protocol (P-ER2) for designing two-input molecular gates. This revised procedure differs from the P-ER1 protocol to design molecular PASS gates presented in Sec. 6.1, Tab. 6.3 by adding another effector binding site and linker in the extension region of helix II.

	Randomisation of lengths for the extension region that is comprised of a helix that attaches to the ribozyme core, two effectors binding regions (NN· · · NN), and three linker sequences connecting the binding sites with the helix (inside dotted box) are generated. Parameters are restricted to those given in Tab. 6.2.
	Sequence positions except the binding sites are assigned by searching for a base sequence that will fold into the target structure designed in the previous step using either *StochSrch* or *RepInit*, with a non-binding pseudo-base (N) being assigned to all positions in the binding region.
	Replacement of the pseudo-bases in the binding regions with real bases. All possible combinations of effector binding are considered using (NN· · · NN) pseudo-bases to represent an unoccupied binding region.

From the computational protocol (P-ER1) for the design of PASS gates presented in Sec. 6.1, we derived a variant protocol named P-ER2 that will allow the search for the logic operators listed in Tab. 6.4 by adding another effector binding site and linker to the extension region of helix II. As in the P-ER1 protocol, we initially start with an active hammerhead ribozyme configuration, then, search for a sequence combination to be placed in the effector binding region that distorts the active hammerhead motifs. But in the case where two effectors are required, we have to extend the protocol to check for all four possible meta-stable conformations of the molecule (i.e., [no E], [E$_1$], [E$_2$] and

[E_1 and E_2], where E denotes the effector molecules). Table 6.6 depicts the protocol of P-ER2 for the design of two-input molecular logic gates. In order to increase the probability of generating sequences that will disrupt the formation of the hammerhead motifs, the complementary bases of the conserved region are embedded at arbitrary locations within the linker extensions, or at the effector binding sites, or overlapping both (i.e., to improve the performance issue in the base-pairing percentage filter, cf. Tab. 6.1).

There are two conditions that need to be investigated in order to generate sequences for the complete set of logic operators. Based on the binary logic table, represented in Tab. 6.4, firstly, we search starting from the structures that are active with the presence of both effector molecules (i.e., the normal direction of the previously described protocol– cf. Fig. 6.6) and secondly, starting from the reverse direction, where the presence of both effectors does not affect the inactive state of the ribozyme (i.e., the conserved regions of hammerhead ribozyme are bound randomly at the start of the search). The first strategy gives solution for the bottom half of the binary logic function depicted in Tab. 6.4, where in the presence of both effector molecules (input-A and input-B in the table) the catalytic function is activated, and the second strategy gives solutions for the top half of Tab. 6.4, where in the presence of both effectors the catalytic function is always deactivated. For clarity, the logic operators for the first strategy are referred to as LG-B, and the logic operators for the second strategy are referred to as LG-T.

We first investigate the distribution of two-input gates that are generated by the automatic design protocol (P-ER2). For this purpose four runs were conducted, where 12,500 candidate sequences were generated in each run. The results are as shown in Tab. 6.7. Any response pattern of the generated structures to two effector molecules will correspond to a row in Tab. 6.4. However, the top row ($A\circ_1B$) and the bottom row ($A\circ_{16}B$) correspond to the case of a constant OFF output and a constant ON output. These two cases that ignore the effector molecules entirely will not be considered further. Only ≈43% of the total candidates can be classified as imitating the conventional binary logic operation where as the remaining ≈57% fall directly under the constant 0's and 1's logic operators, with the latter forming the majority ≈95% of the constant gates. This indicates that the conformation of the ribozyme core remains active despite the absence of the effector molecules, and subsequently, remain unaffected with the presence of either one or even both of the effector molecules. This is equivalent to the failure of allocating a base pairing region for the conserved bases during the design of the effector binding region (OBS), that is intended to disrupt the catalytic activity of the ribozyme core.

As shown in Tab. 6.7, the P-ER2 protocol, with the first strategy, managed to generate candidate sequences for the logic operators in the bottom half of Tab. 6.4 (LG-B). However, as indicated in the "Success Rate" column, after the filtering process (Tab. 6.7, p.132), we observe a significant decrease in the number of candidates for each classi-

Table 6.7: Distribution of candidate sequence generated by the revised computational protocol (P-ER2) in Tab. 6.6, classified according to the type of binary logic operator depicted in the bottom half of Tab. 6.4. In this case, it is mandatory for the operator to be active, whenever the two inputs are present. The candidates generated at each run will then undergo a filter cascade (cf. Tab. 6.1). The total candidates column represents the total number of candidates for each type of logic gate gathered from run 1 to 4. The percentage of sequences that passed the cascade is listed in the success rate column.

	Input state									
A	1	0	1	0						
B	1	1	0	0						
	Output state				Run 1	Run 2	Run 3	Run 4	Total Candidates	Success Rate
	1	0	0	0	3.0%	3.1%	3.0%	3.1%	1517	5.4%
	1	0	0	1	0.6%	0.6%	0.7%	0.6%	326	26.0%
	1	0	1	0	0.2%	0.2%	0.2%	0.3%	127	11.0%
	1	1	0	1	0.3%	0.4%	0.3%	0.4%	167	0.0%
	1	1	0	0	19.5%	20.2%	20.3%	20.1%	10004	8.9%
	1	1	0	1	15.0%	15.0%	14.9%	14.9%	7470	2.7%
	1	1	1	0	3.7%	3.7%	3.7%	3.7%	1850	16.6%
	Constants				57.7%	56.8%	56.9%	56.9%	28539	-

Table 6.8: Surviving candidates at each filter step, calculated continuously depending on the number of candidates at each previous filter step. Detailed descriptions of each filter step are presented in Tab. 6.1.

Steps	Run 1	Run 2	Run 3	Run 4
Filter 1	79.6%	79.7%	87.0%	85.0%
Filter 2	97.3%	97.0%	97.5%	97.6%
Filter 3	60.8%	60.6%	61.6%	59.9%
Filter 4	15.8%	16.2%	16.0%	16.2%
Filter 5	100.0%	100.0%	100.0%	100.0%
Filter 6	100.0%	100.0%	100.0%	100.0%
Filter 7	100.0%	100.0%	100.0%	100.0%

fied operator. Upon closer inspection, as observed in Tab. 6.8, the passing percentage reduced significantly during the execution of filter steps 3 and 4. The latter indicates the lowest surviving percentage at ≈16%, with ≈60% as the second lowest passing percentage for filter step 3. This problem has been encountered previously, during our design of molecular PASS gate using the P-ER1 protocol (Tab. 6.3) motivated by the protocol suggested by (Penchovsky and Breaker, 2005). As we are removing some of the constraints in the structure specification, in order to increase the degree of freedom and the space of plausible structure configuration, the decrease in candidates occurring in both filters are inevitable, for a procedure that relies on the single state sequence design algorithm in generating its candidate sequences. The ideal solution would be to replace the single state design algorithms with a multi-stable design algorithm minimises the energy gap between the meta-stable states. This is investigated in more details in the next section.

Next we focus on the search for the the LG-T binary logic operators, where the operator would remain inactive, in the presence of both input molecules (cf. top half of Tab. 6.4). For the second design strategy, the protocol (P-ER2) assigns the complementary base pairs of the ribozyme core (conserved `CUGAUGAG`-region) in any random positions within the extension region of helix II (cf. figure panel in Tab. 6.6). The following dot-bracket representation of the allosterically controlled hammerhead ribozyme with $[xxxx \cdots\cdots xxxx]$ symbol indicates the possible region where the base pairs complementary to the conserved `CUGAUGAG` bases can be placed.

5'-((((((((CUGAUGAG...$[xxxx \cdots\cdots xxxx]$CGAAA((((....))))U.))))))))-3'

Because of the high proportion of constant gates from our previous runs (LG-B, Tab. 6.7), for the second design strategy, the occurrences of both gates are excluded from the candidate sequences. To balance our finding with the results for LG-B (for the binary logic operators which are active if both effector molecules are present), only 10,000 candidates were generated, distributed across two runs (5,000 candidate sequences for each run). The results for the LG-T (top half of the binary operator in Tab. 6.4) are listed in Tab. 6.9. The passing percentage calculation for the candidate sequences in each filter step is shown in Tab. 6.10.

Table 6.9: Distribution of candidate sequences generated by the P-ER2 computational protocol in Tab. 6.6. The initial start of these runs requires the molecule to remain inactive despite the presence of both inputs, which is the opposite scenario of the logic gates in Tab. 6.7. See Tab. 6.7 for details.

	Input state							
A	1	0	1	0				
B	1	1	0	0				
	Output state				Run 1	Run 2	Total Candidates	Success Rate
	0	0	0	1	9.3%	10.8%	1006	42.8%
	0	0	1	0	35.6%	29.9%	3276	26.5%
	0	0	1	1	7.9%	7.8%	786	1.7%
	0	1	0	0	28.8%	38.1%	3348	29.3%
	0	1	0	1	13.5%	10.4%	1194	3.9%
	0	1	1	0	2.4%	1.6%	200	18.5%
	0	1	1	1	2.5%	1.4%	200	0.0%

From Tab. 6.9, with the exception of three gates (where the success rates are lower than ≈4%), the success rates of the remaining logic gates slightly better than the success

Table 6.10: Surviving candidates for each filter step for the binary logic operators table in Tab. 6.4.

Steps	Run 1	Run 2
Filter 1	82.6%	80.7%
Filter 2	97.9%	98.2%
Filter 3	60.1%	60.6%
Filter 4	37.8%	38.2%
Filter 5	100.0%	100.0%
Filter 6	100.0%	100.0%
Filter 7	100.0%	100.0%

rate for LG-B shown in Tab. 6.7. Despite the slight increase in the success rates, more than 75% of the candidates are discarded during the filtering stages (cf. Tab. 6.10). However, the candidate sequences of LG-T have a higher passing rate in filter step 4 with ≈38% as compared to ≈16% for the generated candidate sequences of LG-B (cf. Tab. 6.8), although the passing rate in filter step 3 at ≈60%. The results indicate, that the computational protocol (P-ER2) which aims to increase the degree of freedom in generating a diversified structure configuration for the design of molecular logic gates seems to be rather inefficient. The meta-stable conformations of the molecules are not considered during sequence assignment by the single state sequence design algorithm (panel 2 in Tab. 6.6). Instead of the single state sequence design algorithms, for the sequence assignment of computational units with multi-stable conformations, a multi-stable sequence design algorithm is required. Using RNA logic gates as test case, we investigate the implementation of a computational protocol with *multiSrch* in the next section.

6.3 Computational Design of RNA Logic Gates using a Multi-stable Sequence Design Algorithm

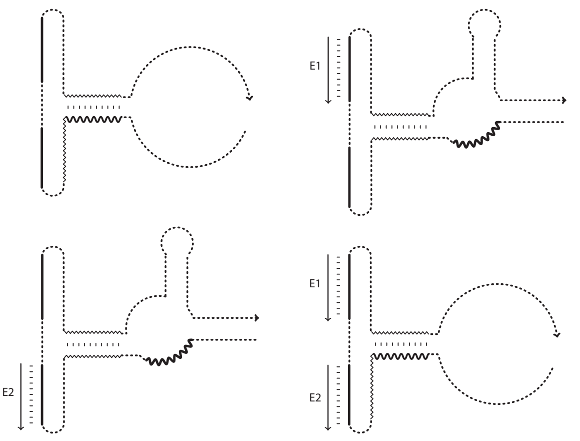
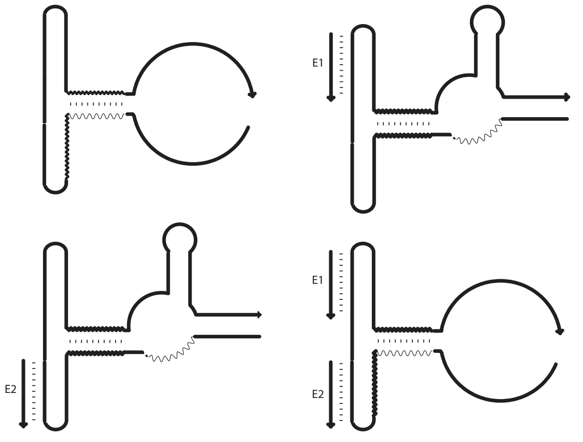
From the engineered DNA and RNA logic gates (Stojanovic and Stefanovic, 2003b; Penchovsky and Breaker, 2005) to the development of synthetic RNA devices (Win and Smolke, 2008; Beisel et al., 2008; Shapiro and Gil, 2008), the type of molecules that are of interest for information processing tasks have a number of meta-stable conformations representing their change of folding in regards to the changes in their environment. For instance, an allosterically controlled hammerhead ribozyme imitating the AND logic operator (Fig. 2.8) has four different meta-stable conformations. These conformations are representative of four conditions, i.e., when no effectors are present, when one effector is present but not the other, and when both effectors are present. For natural occurring riboswitches, there could be a number of possible meta-stable conformations which includes the binding of specific metabolites to their receptor sites and the different stages of conformational shift to triggers their gene control mechanisms, as discussed by Nudler (2006) and Suess and Weigand (2008). Therefore, in order to design nucleic acids computing units, a protocol that includes a multi-stable sequence design algorithm is required. The development of such protocol is discussed in this section, and using RNA logic gate as test case, we evaluate the performance of the new protocol.

In Section. 6.1, using a variant computational protocol developed from the original suggested by (Penchovsky and Breaker, 2005), we have demonstrated the possibility of producing diversified structural configuration for the molecular PASS gates, which are quite unique, compared to the homogeneous configuration implemented in the original protocol. It is evident in our finding that the approach suggested by Penchovsky and Breaker (2005) limits the structural diversity for the construction of computational nucleic acids as the structural space domain is confined to only a set of strictly predefined elements. By expanding the structural space, we managed to find a number of alternative solutions. However, there is an increased in computational time and a declined in the number of solutions suitable for laboratory implementation (based on the filter cascade suggested by Penchovsky and Breaker (2005)). in order to compensate for the increase of the structural space. In the previous section, when we expand the search to include the complete set of binary logic operators, similar issues persisted. From the pool of plausible candidates, more than 75% of the candidates were eliminated in the filtering stage, which further highlights the quality issue of the candidate sequences.

From the results in Tabs. 6.8 and 6.10, the pool of candidates were significantly reduced during filter step 3 (base-pairing percentage) and 4 (energy gap). It appears that the use of a single state sequence designer is insufficient, largely because the multi-stable characteristics (i.e., conformation, free energy and energy gap) are not part of the optimisation objective of the single-state design algorithms. There is a need to substitute the single state sequence designer with a multi-stable sequence design algorithms to take into account the multi-stable characteristics of the molecules. The reduction in filter step 4 is related to the method of constructing the initial structural configurations of the molecule (step 1 in Tab. 6.6). We showed that a less restrictive space is required to produce a diversified set of molecular gates, but accordingly the design space should be constructed by considering a few essential design elements of the molecular gates, e.g., the placement of the receptor sites and the base pairing complementary of the conserved region.

Using the P-ER2 protocol (Tab. 6.6) as a basis, we derive a new protocol named P-ERM which includes *multiSrch* as the multi-stable sequence design algorithm that replaces the single state design algorithm (*StochSrch*) used in P-ER2. Because of the ability of *multiSrch*) to generate sequences for all interacting molecules, we can combine the last two steps of P-ER2 (middle and bottom panels in Tab. 6.6) into a single step. The result is a simplified two-step protocol (Tab. 6.11): firstly we create the plausible structure configuration (known as “partial conformation”) representing the molecules based on their structural and sequence constraints, and secondly the generation of sequences using *multiSrch* with the meta-stable partial conformations (defined in the first step) as input. In P-ERM protocol, the generation of structure in the first step is not entirely

Table 6.11: Computational protocol (P-ERM) based on *multiSrch*.

	Generate all "partial" meta-stable conformations (refer Fig. 6.6 for details) for the molecules. The length of the receptor sites (bold lines), the helix II (short crinkled lines), and the linkers (cf. Tab. 6.6) are randomised within the constraints detailed in Tab. 6.2. For instance, to design an XOR gate, four meta-stable partial conformations are provided. Two inactive states where the hammerhead ribozyme motif is distorted (top left and bottom right) and two active states where the hammerhead ribozyme motif is formed when only one of the effectors is present (top right and bottom left).
	Using *multiSrch*, generate sequences that conform to the meta-stable conformations from the previous step. Bases are assigned for positions in the the bold regions.

random as in the previous protocols (P-ER1 and P-ER2). Only the variable length of the regions is arbitrary selected (cf. Tab. 6.11). To specify a partial conformation, for each state, the regions between which base pairing is derived in the molecule is specified. Figure 6.6 illustrates a partial conformation for a state where the presence of two effector molecules did not activate the catalytic function of the molecule. The conserved bases (indicated by bold wavey lines) are specified to explicitly bind to the helix II region when the two effectors are present, in order to inhibit the catalytic activity of the molecule. Aside from the fixed base pairs (E1 and E2 with their receptor sites, and the conserved `CUGAUGAG`-region with helix II), the remaining positions are not restricted, and can either form base pair remain or unpaired in that particular state. Note that randomisation of

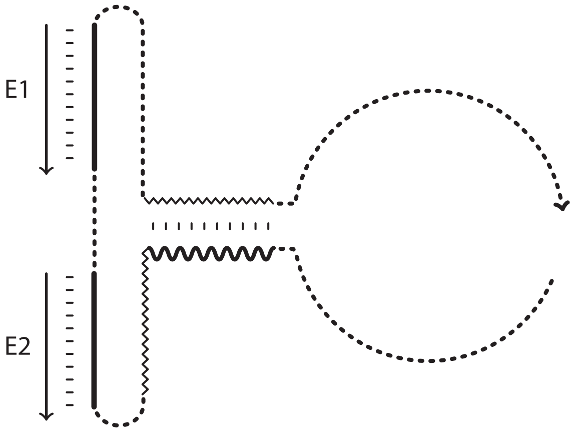


FIGURE 6.6: A sample structure configuration depicting a partial conformation of the binary logic XOR operation, where the presence of two effectors yield no activation of the molecule (cf. XOR in Tab. 6.4). The crinkled lines in the figure represent the helix II base pairing of the active hammerhead ribozyme, while the bold wavey line represent conserved region (CUGAUGAG). The small dashed-lines denote un-fixed conformation, where the base position belonging to this region can be either paired or unpaired as long as other mandatory regions (i.e., the base pairing of CUGAUGAG-region and the complementary half of the helix II, effector 1 (E1) with its binding site and effector 2 (E2) with its binding site) are present.

complementary position are permitted. Using the extended-dot-bracket notation, the partial conformation of the molecule can be written as,

```
*{15}((((((({CUGAUGAG}({15}+E2*{15}({15} +E1)))))))))*{20}
& ){15}+E1 & ){15}+E2
```

where * denotes the dashed-line region with no fixed pairing condition. The base position belonging to this region can either be unpaired or paired, as long as the mandatory base pairing regions are present (e.g., the base pairing of the CUGAUGAG-region and the helix II region of the molecule). Fixing the partial conformation beforehand seems to resemble the protocol suggested by Penchovsky and Breaker (2005). However, since only partial regions in the conformation are fixed, in actual fact, P-ERM protocol still maintains the degree of freedom in generating various structural configurations—rather than stereotyping the molecular structure into a predefined homogeneous conformation.

For the experiment, we created 10 different sets of partial conformations for each type of binary logic operator (T1 to T10). The difference between each set is the length of each element which is randomly selected within the constraints detailed in Tab. 6.2. For each set, the partial conformations corresponding to the binary logic operations are explicitly defined. For instance, the partial conformation for any molecular gate where in one state, the presence of both effectors yields no activation of the ribozyme core is equivalent to the partial conformation presented for the XOR logic gate depicted in Fig. 6.6. Compared to the random base pairing assignment suggested in P-ER2 protocol, the partial conformation contains mandatory base paired regions that must be preserved by the *multiSrch*. In order to promote structural variability, the selection of the complementary pairings are randomised within the specified range (i.e., either a region in the hairpin loop region of helix II, or the helix II region itself).

The parameter settings for *multiSrch* are shown in Tab. 6.12. The settings are largely based on the findings of the evaluation study to generate candidates for DNA and RNA gates described in Section. 5.3. The parameter settings are kept constant for each set of structural configurations. Unlike *StochSrch* in P-ER2, *multiSrch* is a deterministic algorithm and therefore one run is sufficient. To allow for direct comparison with the previous results generated by the P-ER2 protocol, we divided our results into two different categories (LG-T and LG-B). These categories are differentiated by the initial state of its binary logic operations, similar to the initialisation strategy undertaken in the previous section. Tab. 6.13 shows the result for the LG-B logic operators (where without the presence of both effector molecules, the operators are inactive), while Tab. 6.14 shows the result for LG-T operators (where the operator would remain inactive, in the presence of both effector molecules). For each logic gates, 300 candidate sequences were generated based on the sets of partial conformations T1 to T10. These 300 candidate sequences are then filtered using the filter cascade shown in Tab. 6.1. In the inactive state, the conserved region must forms base pair with with any regions from the helical arms II (helix II) until the hairpin loop next to the helix (includes both effector binding sites), and in the active state, this conserved region must be unpaired and the overall structure must have three helices (H1, H2 and H3 in Fig. 2.4) that resembles the conformation of an active hammerhead ribozyme.

Table 6.12: Default parameter setting for *multiSrch*

Parameter	Value
$\mathcal{D}(\Psi^*)$ Order	descending
$\Xi(x)$	Eq. 5.7 with $E_{gap} = -6.0$
Λ	300
KEEP	$\Xi(x) - min(\Xi(x)) = \pm 5.0$

The overall performance of the revised computational protocol with the addition of *multiSrch* is shown in Fig. 6.7. In term of the number of candidates that passed the filter cascade, the P-ERM protocol with *multiSrch* performed significantly better when compared to results from P-ER2 protocol (Sec. 6.2, p.127). Instead of the ≈16% for LG-B and ≈37% for LG-T candidate sequences that pass filter step 4 in P-ER2, most of the candidate sequences generated by *multiSrch* in the P-ERM protocol (for both LG-B and LG-T) pass filter step 4. However, for A$\circ_{11}$B (`1010`), OR (`1110`), A$\circ_3$B (`0010`), and the XOR (`0110`) logic operators (cf. Fig. 6.7), there is drop of more than 20% of the candidate sequences in filter step 2. This indicates that for ≈20% of the candidate sequences for the four logic operators mentioned, the active hammerhead

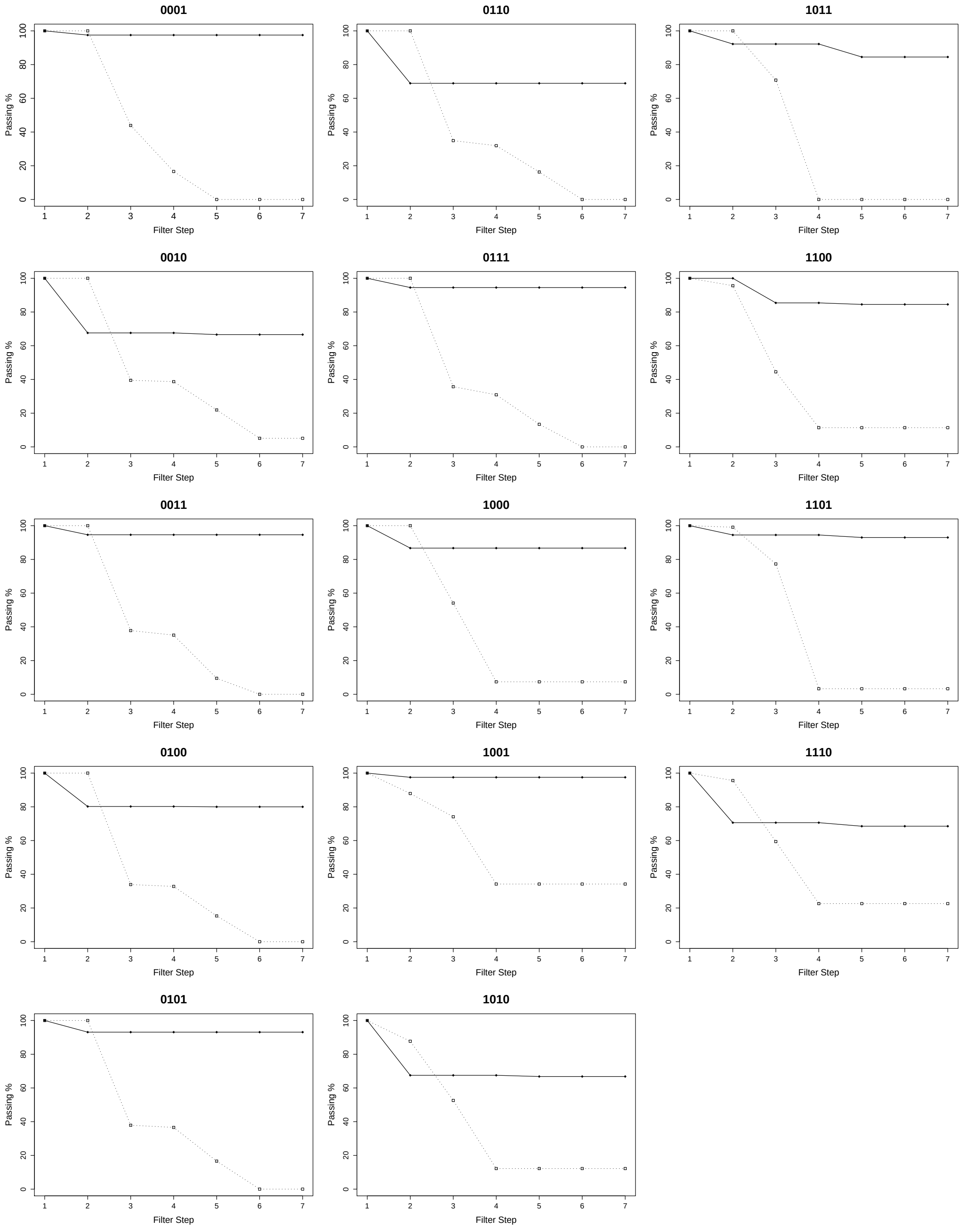


FIGURE 6.7: Quality of the design of nucleic acid logic gates using the single-state and multi-state sequence designers. The graphs show the percentage of candidates that passed the filter for every stage in the filter described in Tab. 6.1. A Solid line represents candidates that were generated by multi-state designer, and a dashed line represents candidates from the single-state designer. The title of each graph represents, from left to right, the output bits of the following input patterns (11, 10, 01, 00), with the inputs in the order of [input-B input-A], see Tab. 6.4. For instance, an OR gate is denoted as 1110 in the figure. Refer Tab. 6.13 and 6.14 for the actual number of candidates.

ribozyme conformation was not obtained. The consistency of the results (for T1 to T10) are also maintained as shown in Tabs. 6.13 and 6.14.

For LG-B logic operators, (cf. Tab 6.13), one can observe a significant improvement in the number of candidate sequences that pass the filtering procedure (indicated by percentage of passing in column Rate in Tab 6.13) when compared to the P-ER2 protocol (cf. Tab 6.8). Note the number of candidate sequences that pass the filter cascade. The worst passing rate across all types (T1 to T10) is recorded at ≈60%, which is still significantly better than the ≈25% we saw in the previous section.

TABLE 6.13: Distribution of filtered candidate sequences generated by the P-ERM (cf. Tab 6.11) protocol with *multiSrch* multi-stable states designer, classified according to the type of binary logic operator (LG-B) shown in the bottom half of Tab. 6.4. Depicted in the table is the number of candidate that passed the seven filter cascades (cf. 6.1) for each respected run. Before the filtering protocol, 300 candidates were generated for each type of logic operation in each run. The mean percentage of success rate is given in the "Rate" column.

	Input state														
A	1	0	1	0											
B	1	1	0	0											
	Output state				T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Rate
	1	0	0	0	210	205	269	300	300	300	209	300	278	200	85.7%
	1	0	0	1	281	300	297	245	298	300	300	285	275	300	96.0%
	1	0	1	0	200	201	188	181	206	204	200	192	201	214	66.2%
	1	0	1	1	264	276	259	231	253	250	239	293	262	273	86.7%
	1	1	0	0	300	300	221	269	277	209	203	269	212	277	84.6%
	1	1	0	1	278	300	251	245	251	300	300	300	265	300	93.0%
	1	1	1	0	242	208	203	208	214	200	204	193	184	201	68.6%

Analysis of the filtering process showed that the reduction of the candidate sequences occurred during filter step 2, where the conformation resembling an active hammerhead ribozyme was not obtained from the four meta-stable conformations. For instance, for $A\circ_{11}B$ gate with the output state of 1010, we found out that the binding of a single effector molecule did not triggered any conformational shift that resembles an active ribozyme conformation. Note that the filtering procedure conducted here is a direct implementation of the filtering model suggested by Penchovsky and Breaker (2005). In order to test the presence of a an active hammerhead ribozyme conformation, Penchovsky and Breaker (2005) (due to the lack of multi-folding prediction programs) suggest to replace the bases for the effector binding sites with "X"s. When one folds the sequence using *RNAfold*, this "X"s-region would represent a mandatory unpaired region. Thus simulating a conformation of a molecule with an effector molecule externally bound to it. Although we adopted the same filter technique, we do not think that this filter accurately predicts the inter-molecular binding between these interacting molecules. The amount of energy release during the formation of external binding is strong enough to break or shift existing internal hybridisation bonds and thus can lead to a change in conformation of the molecule. If we consider the intermolecular binding efficiency between the effector molecule and receptor site instead of the refolding of sequences with "X"s-region, then more than 90% of the generated sequences from *multiSrch* for LG-B have perfect binding between effector and receptor site as simulated by *RNAup* (Mückstein et al., 2006).

The reduction of candidate sequences for A$\circ_{11}$B (1010) and OR (1110) logic gates in filter step 2, can also be contributed by the low free energy generated from these candidate sequences. For a low free energy structure, the base pair composition is highly dominated by either C-G or G-C pairing. Despite the effort to enforce the identical base pairing rules (cf. filter step 1 in Tab. 6.1), there is still an abundance of C-G and G-C pairings occurring (i.e., non-consecutive, but distributed in the group of three or four) in these lower free energy structures. As shown in the Section. 5.3, by specifying the ΔG_{opt} target value in *multiSrch*, the algorithm is then able to generate sequences with higher free energy values which are better suited to the meta-stable molecules to be implemented. For this purpose, ΔG_{opt} can be based on nucleic acids logic gates that have already been engineered in the laboratory.

Table 6.14: Distribution of filtered candidate sequence generated by the P-ERM protocol with *multiSrch* multi-stable designer. The columns (T1 to T10) represent the number of candidate sequences that passed the seven filter cascade in Tab. 6.1. For each logic operator, 300 candidate sequences were generated. The mean percentage of success rate is given in the "Rate" column.

	Input state														
A	1	0	1	0											
B	1	1	0	0											
	Output state				T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Rate
	0	0	0	1	298	300	292	291	296	300	300	275	275	300	97.6%
	0	0	1	0	201	187	200	231	196	198	200	192	191	200	66.5%
	0	0	1	1	278	300	297	245	298	300	300	285	275	259	94.6%
	0	1	0	0	300	199	246	200	300	259	210	212	278	198	80.1%
	0	1	0	1	278	300	297	245	213	300	300	285	275	300	93.0%
	0	1	1	0	209	200	209	194	251	200	183	192	200	192	67.6%
	0	1	1	1	275	300	251	245	300	300	300	300	265	300	94.5%

Table 6.14 shows the overall result of generating candidate sequences for LG-T logic operators. The results are similar to the LG-B logic operators in Tab. 6.13, where there are two types of logic operators (A$\circ_3$B and XOR) where the success rates are below ≈66%. During the filtering process, the reduction in candidate sequences occurs during filter step 2. Upon closer inspection, we found that the filtering procedure for step 2 in Tab. 6.1 that is implemented based on the model of (Penchovsky and Breaker, 2005) might be flawed because the base pairing formation between the effector molecules and its corresponding binding site are present when the generated sequences are simulated using *RNAup* (Mückstein et al., 2006). The inability of the molecules to shift conformation

(with the substitutions of "X"s bases) might be due to the low free energy of these sequences. The possibility to specify a desired MFE (ΔG_{opt}) for the *multiSrch* algorithm was motivated by this issue.

From the results in Tabs. 6.13 and 6.14, the type gates with consistently poor numbers of filtered candidates across all ten structural designs (T1 to T10) were selected. Four gates were identified, the A$\circ_{11}$B (1010), OR (1110), A$\circ_3$B (0010), and the XOR (0110) gates where the success rate is less than ≈69%. The partial conformations for each of these gates can be classified as "unfavourable". For these molecular gates, the presence of any one of the effectors can activate the ribozyme. However, the partial conformation that is arbitrarily selected in this comparison study is inadequate for dealing with this condition, because in the inactive state, some regions belonging to the receptor site always bind together. The self-assembly between the effector and receptor site for either one of the input (A or B), might not be sufficient in triggering a conformational change to activate the ribozyme. The design of A$\circ_{11}$B logic gate would be more "favourable" if the effector representing input B is shorter compared to the effector for input A. The binding of effector B is estimated not to change the conformation as much as the binding of effector A to its receptor site. A "favourable" partial conformation design for each of these four gates would improve the passing rate of the candidate sequences specially for filter step two.

The comparison study in the previous chapter showed that the default setting of the *multiSrch* algorithm usually arrives at sequences with the low MFE. These sequences are therefore quite stable and would require to overcome a high energy barrier to disassociate some of the existing base pairs.

Table 6.15: Design of logic gates using a desired MFE. The number of candidates that pass the filter cascade generated for the four selected binary logic operators using ΔG_{opt} = -40.00 kcal/mol.

	Input state														
A	1	0	1	0											
B	1	1	0	0											
	Output state				T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Rate
	1	0	1	0	281	281	278	273	286	274	266	272	271	284	92.0%
	1	1	1	0	262	278	284	288	272	285	276	283	264	291	92.8%
	0	0	1	0	251	254	240	272	255	243	249	258	255	261	84.6%
	0	1	1	0	221	243	233	245	251	240	232	227	244	253	81.6%

A test to generate sequences for the four gates which were difficult to design using the same initial structures but with target MFE of ΔG_{opt} = -40.00 kcal/mol. There is an increase in the percentage of success rate of all four logic gates, from ≈66% to ≈81% as indicated in Tab. 6.15. For the two logic gates of A$\circ_{11}$B (1010), and OR (1110), the success rate increased up to ≈92%, approximately 10% better than the other two logic gates of $A \circ_5 B$ (0100), and XOR (0110). The likely cause of this issue is the design of the partial conformation itself. For instance, in the design of XOR gates, the design of an external binding for both effector molecules must be strong enough to completely disrupt the formation of an active conformation that needs to be present with a single effector molecule.

It is best, if we can supply the multi-stable states design algorithm (*multiSrch*) with a complete structural details of the molecule, including changes in its meta-stable state. Regardless of the level of details available in the partial conformations, the computational protocol (P-ERM) should still be able to produce a good number of candidate sequences, as shown in this evaluation (Tab. 6.15). The success rate of the “difficult” structures illustrated by the XOR and $A \circ_3 B$ are still above 80%, which is very encouraging because only partial conformations are supplied in the protocol. From the pool of successful candidates sequences, one can simply select any of the workable conformations as inputs for *multiSrch*, and accordingly with the estimation of MFE and energy gap, one would likely be able to produce a good number of candidate sequences. In order to construct a fixed structural configuration, one is required to fix the mandatory base pairing and unpaired region of the molecules, thus creating a partial conformation (Fig. 6.6) of the molecules for each meta-stable state. The undefined regions (i.e., non-mandatory base positions) are allowed to form base-pair or remain unpaired as long as the mandatory base pairings and unpaired regions are preserved. Only the conserved bases are specified as constraint, allowing another level of diversity for the sequences. This approach is suitable if the target conformation is only partially known beforehand.

For designing the complete set of binary logic operators in Sec. 6.3, we only supplied partial conformations as input to *multiSrch* (Ref. Tab. 6.11). In order to promote diversity among the designed sequences, the length for extension region (receptor site–OBS, helix II, and linkers) was randomised. However, the intended base pairing is determined beforehand, in order to confine the generation of sequences to conform to the structural constraint. The design of this partial conformation is depicted in Fig. 6.6. Figures 6.8 and 6.9 show various samples of inactive conformations for allosterically controlled ribozymes that were found during the design of RNA logic gates. For the LG-B logic operators (Ref. Tab. 6.13), the inactive or “off” state conformations can be categorised into three types according to the number of branches formed in the secondary structure of the molecule. Within each of the three classes the structure vary by minor details such as small internal loops and bulges. The positions of the conserved region and oligo binding sites are similar within each class.

Figure 6.9 (for LG-T logic operators) shows the type of conformation found for the inactive state of a molecule, if either one of the effector molecules is present (Fig. 6.9A) or when both effector molecules are present (Fig. 6.9B). Again the conformation encountered during the design of logic gates fall typically in the two classes illustrated by the sample structures. Within the two classes, there are minor structural differences

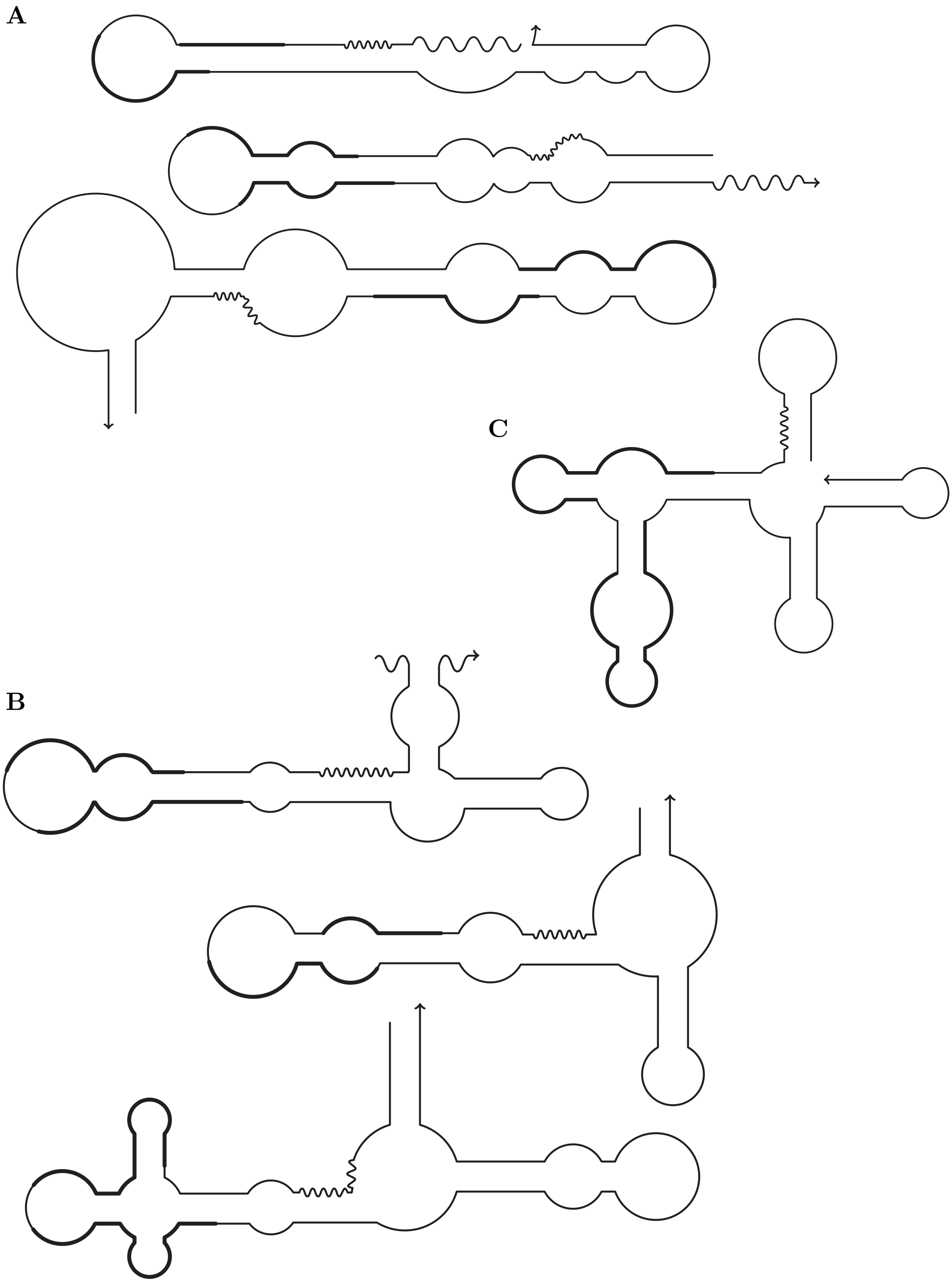


FIGURE 6.8: Possible conformations of nucleic acids logic gates in inactive state when both effector molecules are absent (Ref. to Fig. 6.6 for specific partial conformation). The 1 or 2-branch type (A), the 3-branch type (B), and the 4-branch type (C) are shown. The thick lines represent receptor sites (OBS regions), crinkled lines represent conserved regions (ribozyme core), and the wavey lines indicate dangling ends.

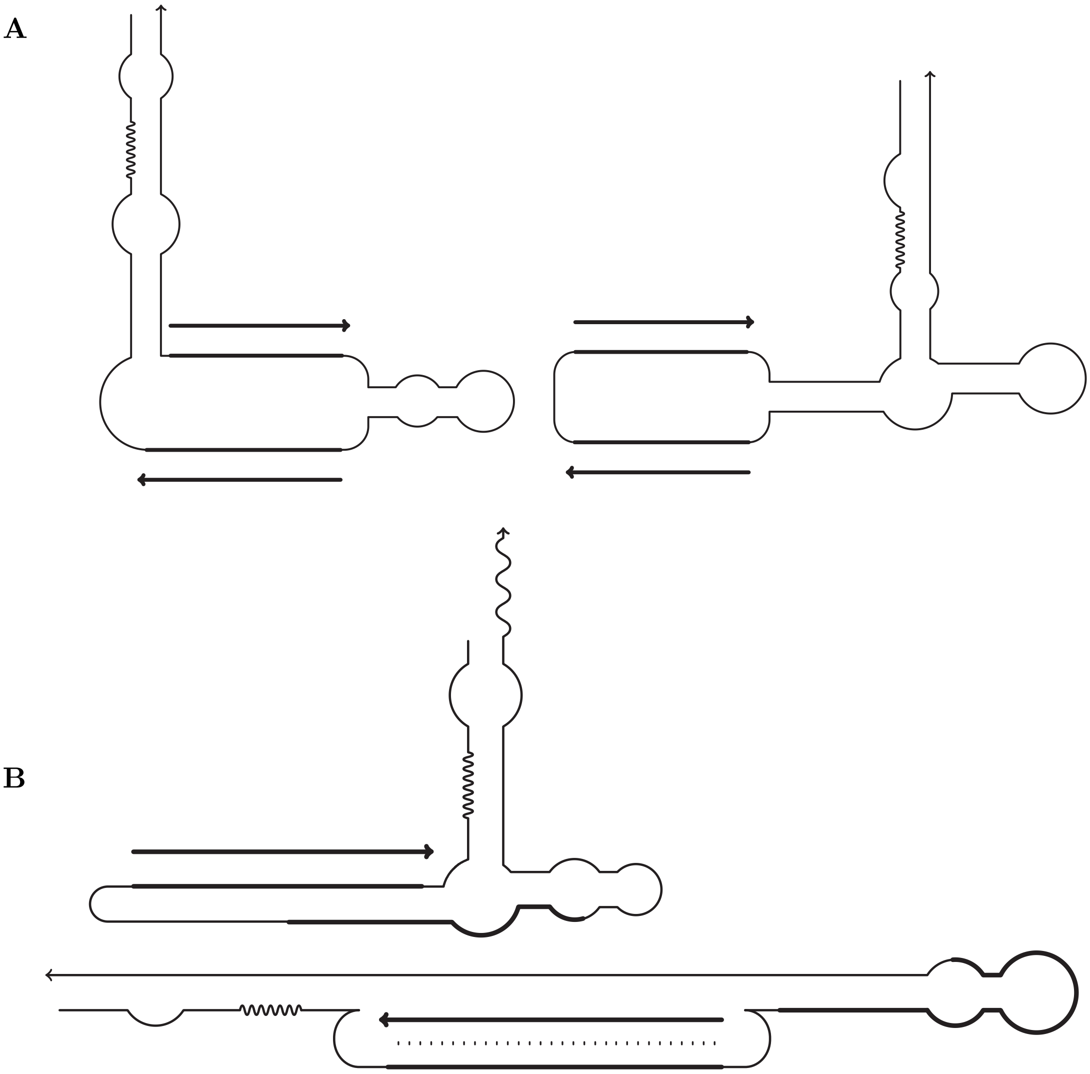


FIGURE 6.9: Inactive conformations of sample nucleic acid logic gates (Ref. to Fig. 6.6 for specific partial conformation) when either both the effector molecules (A) or only a single effector molecule (B) are present. The thick lines represent receptor sites (OBS regions), crinkled lines represent conserved regions (ribozyme core), and the wavey lines indicate dangling ends. In (A), the presence of both effector molecules straighten the OBS binding site region on the molecules and thus triggers a conformational shift. In (B), a single effector molecule steers a conformational change that deactivates the ribozyme.

which are more apparent in the first class Fig. 6.9A than in the second class Fig. 6.9B. A design for a molecular XOR gate was selected to illustrate its operation from the pool of candidate sequences that passed all the filter stages in Tab. 6.1. The four meta-stable conformations of this XOR design are shown in Fig. 6.10. For the logic operation of the molecular XOR gate the presence of a single effector molecule would trigger conformational changes that form the active conformation of the ribozyme. The presence of both effector molecules in this case would stretch the binding site region, disrupting the formation of the hammerhead motif and thus deactivating the ribozyme.

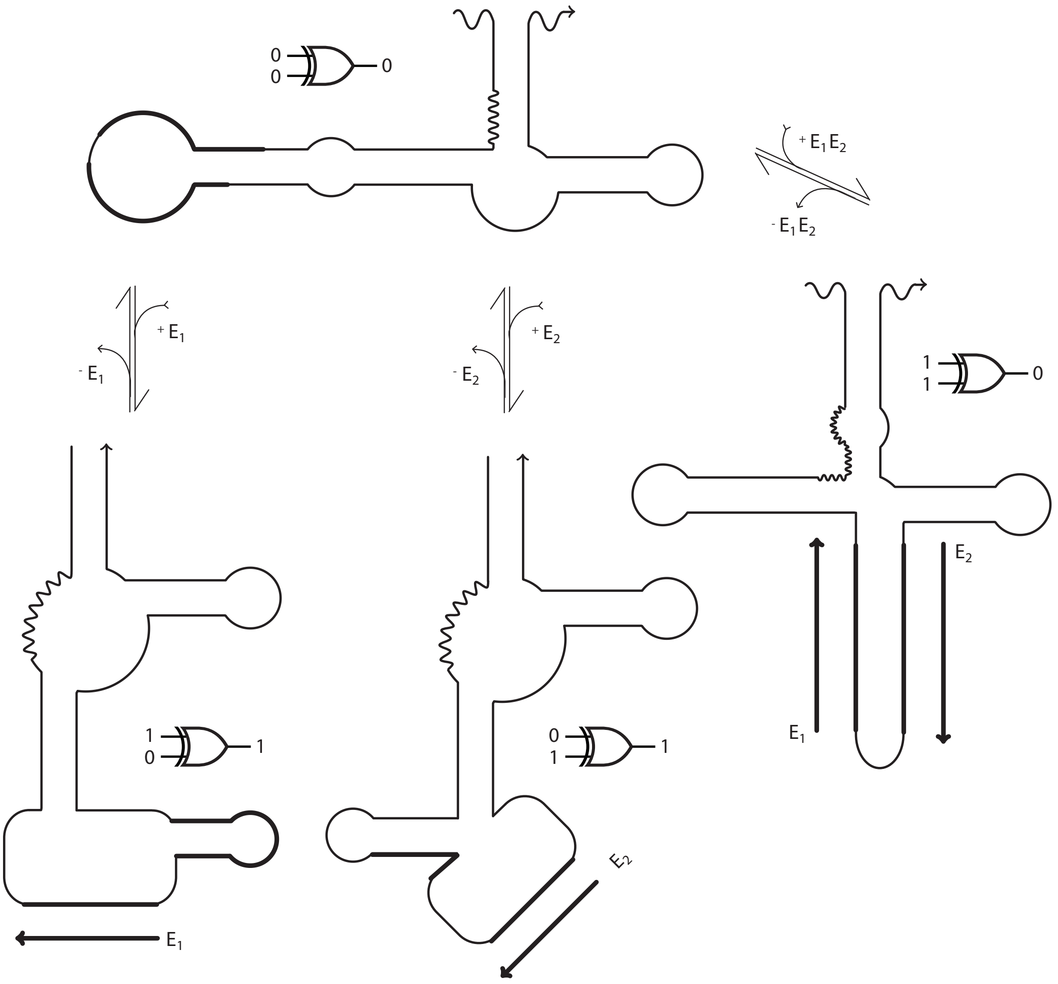


Figure 6.10: Molecular XOR gate generated using P-ERM computational protocol (selected from successful candidates in T1). The inactive conformation of the molecule (top left) changes in the presence of effector molecule E$_1$, to the active ribozyme conformation (bottom right). The presence of effector molecule E$_2$ triggers a different conformation shift, that also activate the ribozymes (middle bottom). The presence of both effectors stretches out both binding regions and disrupts the formation of an active hammerhead motif (right).

Penchovsky and Breaker (2005) suggested a computational protocol to assist in the construction of ribonucleic acids logic gates. The protocol however, is restricted and specifically tuned only for generating sequences for allosterically controlled ribozymes imitating conventional logic gates. This limits the degree of freedom in generating various structural configurations. For constructing computational nucleic acids, structural variability is important because in the laboratory, depending on the physico-chemical environment, only some designs are practicable. By having a set of structural configurations, then it is possible to not only use the configuration that is workable under the given conditions, but at the same time, allows for the best design to be applied. It is also important because it allows one to investigate the type of structural complexity that might be required to solve certain information processing tasks. For instance, the design of a cascade of computing units (to relay or transform a signal) might require a large number of different structural configurations. The protocol of Penchovsky and Breaker (2005) becomes insufficient if one desired to construct a number of computational units.

A revised protocol (P-ER1) was developed primarily to increase the diversity of the structural configurations. The P-ER1 protocol was tested to design molecular PASS gates, and despite its ability to increase structural variability, most of its candidate sequences failed during the filtering process. The P-ER1 protocol was extended to P-ER2 protocol in order to investigate its capability in generating the complete set of binary logic operators (cf. Tab. 6.4). Identical to P-ER1, most of the sequences generated by P-ER2 were eliminated during the filtering process. From the results, we found that the multi-stable conformation characteristic of the molecular computing units has not been considered by the single state sequence design algorithm (*StochSrch*) included in P-ER2. A new protocol called P-ERM was developed to resolve this issue with a multi-stable sequence design algorithm (*multiSrch*). In general, the P-ERM protocol comprises of two phases (Tab. 6.11), the construction of the partial conformation (cf. Fig. 6.6) and the generation of the candidate sequences (using *multiSrch*) that conform to the partial conformation. The partial conformation allows user to specify both the structural and sequence constraints for each state. Using the binary logic operators as a test case, the P-ERM protocol generated a set of structural configurations illustrated in Figs. 6.8 and 6.9, with candidate sequences that have a high success rate during the filtering procedure. The feasibility of generating the complete set of binary logic operators in this chapter indicates both the effectiveness (in term of generating sequences with high success rate) and the efficiency (using only one run of *multiSrch*) of the protocol.

6.3.1 The Design of Sample Nucleic Acid Aptamers

Thus far, we have demonstrated the ability of the P-ERM protocol in generating sequences for the complete binary logic operators (Sec. 6.3). The construction of these conventional logic gates is intended to evaluate the performance of the protocol (P-ERM) in designing simple nucleic acid computers. In order to show the generality of the protocol, in this section, we construct a few designs of nucleic acid aptamers with sticky ends that can self-assemble with another molecule into a chain of blocks. These sticky end regions become available only if an effector molecule is present. For instance, an aptamer resembling the stem-loop structure of a molecular beacon (Stojanovic et al., 2001) is illustrated in Fig. 6.11.

A sample DNA aptamer in Fig. 6.11 has two meta-stable conformations. In one state, the molecule folds into a stem-loop structure and in another state, when the effector molecule is present, the receptor site of the molecule forms base pairs with the effector molecule and straighten the molecule creating two sticky ends. If the length of the

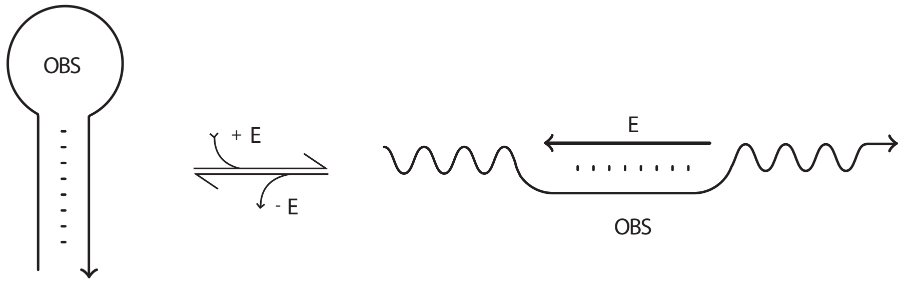


Figure 6.11: A sample DNA aptamer resembling the stem-loop structure of a molecular beacon (Stojanovic et al., 2001). The structure consists of a 8 nt helix and a hairpin loop of 8 nt, which acts as the effector binding site (OBS). The effector molecule (E) binds to the receptor site (OBS), thus creating two dangling (sticky) ends. These dangling ends function as templates for self-assembly with another molecule.

helices and loop are 8 nt, following the complementary base pairs (Sec. 2.4.1.1 p.28) we can assign consecutive `A-U` base pairs to form the helices, consecutive `C` for the receptor site and accordingly consecutive `G` for the effector molecule. However, if the aptamer in Fig. 6.11 is expanded to self-assembles with two more aptamers with identical structural properties as illustrated in Fig. 6.12, then the computational protocol is required to generate the sequences that conform to the design.

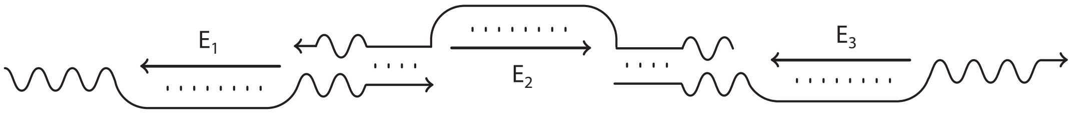


Figure 6.12: A chain of DNA aptamers consists of three basic units introduced in Fig. 6.11. Three effector molecules are required for each unit to bind to the receptor site revealing the sticky ends. The middle unit that links the other two units is called a “bridge”.

Six molecules are required for the design. For the design of the three DNA aptamers (referred as A, B and C), the 4 positions before the $3'$ end of unit A must form base pairs with the 4 positions before the $3'$ end of unit B, and the 4 positions from the $5'$ end of unit B must form base pairs with the 4 positions from the $5'$ end of unit C. Unit B that links unit A and unit C is called a “bridge”. The extended notation representing the chain of molecules in Fig. 6.12 is as follows,

```
State 1  ⟶  (8.8)8 & ........ & (8.8)8 & ........ & (8.8)8 & ........

State 2  ⟶  .8[8(]+1.4[4(]+2 & [)8]+1 & .4[)4]+4[(8]+3[)4]+2.4 &
            [)8]+3 & .4[4(] +4[8(]+5.8 & [)8]+5
```

where, `[)8]+1`, `[)8]+3` and `[)8]+5` denote the effector molecules that form base pair with the receptor sites in molecule A (1), molecule B (3), and molecule C(5). After

Table 6.16: Candidates sequence for the chain of DNA aptamers as illustrated in Fig. 6.12. For each candidate sequence, the base pair regions that link molecule A, molecule B, and molecule C are boxed. The notation of nucleic acid sequences is written from left to right in 5′ to 3′ direction.

Molecule A	Effector	MFE	E_{gap}
GAGGCCCACCGCGACCUGGG $\boxed{\texttt{CCUC}}$	GGTTGCGG	-14.91	0.42
GAGGCCCACCGCUUCCUGGG $\boxed{\texttt{CCUC}}$	GGAGGCGG	-15.10	0.63
GAGGCCCACCUUCAUCUGGG $\boxed{\texttt{CCUC}}$	GATGAAGG	-14.52	2.62
GAGGCCCACCUUGAUCUGGG $\boxed{\texttt{CCUC}}$	GATCAAGG	-14.53	2.60
GAGGCCUACCUUUAUCUGGG $\boxed{\texttt{CCUC}}$	GATAAAGG	-14.27	2.64

Molecule B (Linker)	Effector	MFE	E_{gap}
GGGU $\boxed{\texttt{GAGG}}$ ACGGCGAC $\boxed{\texttt{CCUC}}$ ACCC	GTCGCCGT	-14.92	0.32
GGGU $\boxed{\texttt{GAGG}}$ AUGGCCAC $\boxed{\texttt{CCUC}}$ ACCC	GTGGCCAT	-14.61	0.50
GGCU $\boxed{\texttt{GAGG}}$ ACGGCCAC $\boxed{\texttt{CCUC}}$ AGCC	GTGGCCGT	-14.63	0.92
GGGU $\boxed{\texttt{GAGG}}$ ACGGCCUC $\boxed{\texttt{CCUC}}$ ACCC	GAGGCCGT	-14.60	1.07
GGGU $\boxed{\texttt{GAGG}}$ ACGGCUAC $\boxed{\texttt{CCUC}}$ ACCC	GTAGCCGT	-14.66	1.12

Molecule C	Effector	MFE	E_{gap}
$\boxed{\texttt{GAGG}}$ GGCCGCCCUGACGGCCCUUC	GTCAGGGC	-14.97	0.67
$\boxed{\texttt{GAGG}}$ GGCCGCCCUGACGGCCCUUC	GTCAGGGT	-14.32	0.73
$\boxed{\texttt{GAGG}}$ GGCCCACCUGACGGCCCUUC	GTCAGGTG	-14.29	1.13
$\boxed{\texttt{GAGG}}$ GGCCUCCCUGACGGCCCUUC	GTCAGGGA	-14.83	1.12
$\boxed{\texttt{GAGG}}$ GGCCACCCUGACGGCCCUUC	GTCAGGGT	-15.01	1.26

the preparation of the default conformations, we then proceed to the design of candidate sequences, by inputting the following conformation into *multiSrch*. Table 6.16 shows the candidate sequences for the chain of DNA aptamers in Fig. 6.12 generated by *multiSrch* based on the meta-stable conformations described in p. 148. The target energy gap (E_{gap}) is set to -2.5 kcal/mol with error value of retaining suboptimal candidate (KEEP) is set to -1.5 kcal/mol. Because *multiSrch* generates sequences for a set of interacting molecules with meta-stable conformations, then for the design of the DNA chain aptamers in Fig. 6.12, the candidates sequences for the three DNA aptamers and their effectors are generated in one run.

The base composition among the three aptamers is important to reduce the possibility of these aptamers to bind with each other. The dependency pathways for *multiSrch* is identified for the entire meta-stable conformations across all molecules. For the two meta-stable conformations of the DNA aptamer chain in Fig. 6.12, in one state, each DNA aptamer always folds to itself and in another state, each DNA aptamer has regions that form external binding with the other DNA aptamers. Therefore, if bases `CCUC` is assigned to the last 4 positions before the 3′ end of molecule A, then this region binds with bases `GAGG` in molecule B. Because of the dependency, to ensure that molecule A folds back to itself, the first 4 positions of the 5′ end for molecule A is assigned the same bases (`GAGG`). The base assignment in *multiSrch* always considers both intra and intermolecular folding of each molecules. As a result, in Tab. 6.16, the free energy only slightly differs among the three DNA aptamers.

A sample RNA aptamer with three effector molecules is illustrated in Fig. 6.13. When the three effector molecules bind to their respective receptor sites, a conformational change that releases four binding regions of the molecule. Compared to chain of DNA aptamers in Fig. 6.12, the sample RNA aptamer in Fig. 6.13 focuses on the design of a molecule with multiple effector molecules. When all three of the effectors are present and bind to their receptor sites, the RNA aptamer is straighten to create four possible external binding regions. The sequence design of four molecules is required. Three for the effector molecules, and one for the main aptamer unit. The sample RNA aptamer is 56 nt long, with 4 possible external base pairing regions of 8 nt long each and 3 receptor sites that are also 8 nt long. Table 6.17 shows the candidate sequences for the sample RNA aptamer, with structural configuration illustrated in Fig. 6.13.

To show the accuracy of *multiSrch* in generating sequences with different energy gap target value, we conducted two runs of *multiSrch*. For the first run, the target energy gap (E_{gap}) is set to 15.00 kcal/mol, and for the second run, the target energy gap is lowered to 10.50 kcal/mol. In both runs, five candidate sequences were selected. The diversity among the five candidate sequences is small ($\mathcal{D}$ < 5 nt), where $\mathcal{D}$ is the distance of two sequences calculated using the hamming distance measure (Sec. 4.1 p.57). This indicates that within the pool of candidate sequences generated by *multiSrch*, the quality (i.e., structure, MFE, and energy gap) of the suboptimal solutions are closer to the optimal solution. For the generated sequences in Tab. 6.17, in the first run the same three effector molecules were obtained, with five different aptamer sequences that share the same receptor sites. For the second run, the same scenario is observed.

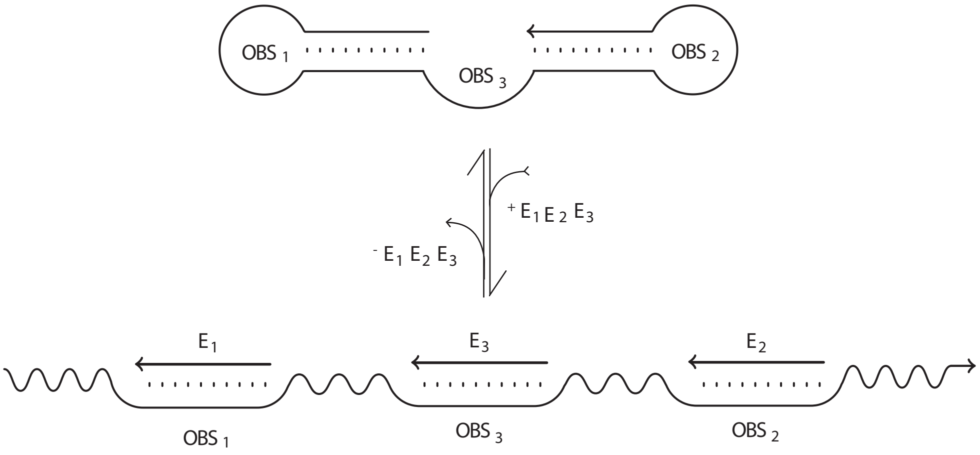


Figure 6.13: A sample RNA aptamer with three receptor sites. Four possible base pairing regions are released upon the presence of three effector molecules that bind to their respective receptor sites. The three receptor sites are labelled as OBS_1, OBS_2 and OBS_3. The wavey lines represent possible external base pairing region.

TABLE 6.17: Candidate sequence for the design of RNA aptamer in Fig. 6.13. The underlined regions represent the three receptor sites for each molecule.

	E_1	E_2	E_3	MFE	E_{gap}
GUGCCCGCAGGGCGGCGUGGGCACGAGUGCACGGUCCGGGAGGGACUCCCUGGACC	GCCGCCCU	GUGUACUC	GAGUCCCU	-46.75	14.45
GUGCCCGCAGGGCGGCGUGGGCACGAGUGCACGGUCCUGGAGGGACUCCCGGGACC	GCCGCCCU	GUGUACUC	GAGUCCCU	-46.73	13.57
GUGCCCGCAGGGCGGCGUGGGCACGAGUGCACGGUCUCGGAGGGACUCCCGGGACC	GCCGCCCU	GUGUACUC	GAGUCCCU	-46.73	13.57
UUGCCCGCAGGGCGGCGUGGGCAAGAGUGCACGGUCCCCCAGGGACUCGGGGGACC	GCCGCCCU	GUGUACUC	GAGUCCCU	-45.34	15.86
GUUCCCGCAGGGCGGCGUGGGAACGAGUGCACGGUCCCCCAGGGACUCGGGGGACC	GCCGCCCU	GUGUACUC	GAGUCCCU	-46.82	15.88
UCCGGGUCGUGGGACCGACCCGGGGAGAGAGAGGUCCGGCCAGGGACCGCCGGACC	GGUCCCAC	UCUCUCUC	GGUCCCUG	-43.76	8.94
GCCGGGUCGUGGGACCGACUCGGCGAGAGAGAGGUCCGGCCAGGGACCGCCGGACC	GGUCCCAC	UCUCUCUC	GGUCCCUG	-44.05	9.55
ACCGGGUCGUGGGACCGACCCGGUGAGAGAGAGGUCCGGCCAGGGACCGCCGGACC	GGUCCCAC	UCUCUCUC	GGUCCCUG	-43.77	10.73
GCCGGGUCGUGGGACCGAUCCGGCGAGAGAGAGGUCCGGCCAGGGACCGCCGGACC	GGUCCCAC	UCUCUCUC	GGUCCCUG	-44.05	10.75
GCCGGUUCGUGGGACCGAACCGGCGAGAGAGAGGUCCGGCCAGGGACCGCCGGACC	GGUCCCAC	UCUCUCUC	GGUCCCUG	-44.05	11.35

The two designs of nucleic acid aptamers discussed in this section are intended to show the generality of the P-ERM computational protocol. Instead of the partial conformation (Fig. 6.6) used in the design of logic gates (Sec. 134), a complete meta-stable conformations is supplied to *multiSrch* for the two nucleic acids aptamers in Figs. 6.12 and 6.13. We demonstrate that the sequence design of multiple molecules with meta-stable conformations can be made in one run. We also demonstrate the ability of *multiSrch* to generate a diverse set of sequences that shares the same set of effector molecules. The computing speed and sequence diversity in which these candidate sequences were generated show the efficiency of *multiSrch*. P-ERM protocol with *multiSrch* allows users to specify the structural constraints of the computing units, tune the characteristic of the candidate sequences (i.e., specifying the MFE or E_{gap} parameters, adjust the sequence variability), and substitute different tools to suite a specific design criteria in generating computational nucleic acids.

Chapter 7

Discussion

7.1 Research Summary

Many published information processing schemes, purportedly intended for use with molecular materials are impractical, because the formal designs of these schemes are ignorant of the micro-physical behaviour of molecular materials. The design of realisable molecular architectures therefore has to take the physics of the molecules into account. In particular the man-made information processing schematics of these computational nucleic acids are driven by free energy minimisation and folding kinetics. These schematics are designed based on the secondary structure folding of the molecule, which includes changes in conformation at certain intervals that are triggered by changes in its physiological environment. Individually, the molecules intended for information processing must be able to kinetically fold into multiple meta-stable conformations. In designing these information processors, one of the conformations is representative of the RNA molecule in equilibrium and therefore has the energy among all conformations. The energy barrier separating these meta-stable conformations must be sufficient to separate conformational states, enable switching to the relevant ones and also deter switching back to the unwanted conformations. Secondary structure prediction forms the basis for the development of a sequence design algorithm for single state molecules (cf. Chapter 4), that was later extended into the design of sequences for multi-stable molecules and interacting molecules (cf. Chapter 5).

The conformational dynamics of nucleic acid that corresponds to the introduction of another molecule, which self-assemble to the receptor site and trigger a structural switching is exemplified in nature through riboswitches and engineered in the laboratory as allosterically controlled ribozymes. If one would design a cascade of nucleic acid computing units, each of which initiates the next unit, then a change that occur in one unit in the cascade would subsequently effect all the remaining units that are apart of the cascade. Because complementary base pairs must be present for the self-assembly

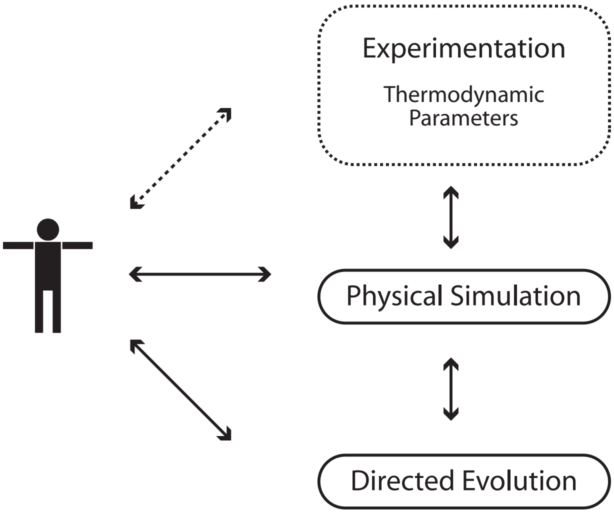


FIGURE 7.1: Design by orchestration. The molecular building blocks for information processing are produced based on the interplay of physical simulation (with measured parameters) and a directed evolution process that adapts the context and properties of each molecule to confer to a particular function, governed by the orchestration of a conductor (user). Adapted from (Zauner, 2005a).

between an effector and a receptor site to occur, then a change of an effector molecule is directly related to its receptor site. The conformational dynamics does not support strict conventional programming. Therefore, in Chapter 6, a computational protocol that considers the multi-stable conformations of nucleic acid molecules were discussed and developed. To facilitate the design of interacting molecules with meta-stable conformation, energy gap is calculated instead of energy barrier as the estimation of energy barrier is computationally expensive. The estimation of energy barrier can be apply later to the set of candidate sequences generated by the computational protocol that has fulfil the necessary filtering procedure.

Based on the concept of orchestrating informed-matter introduced by Zauner (2005a) (c.f. Fig 7.1), the "programming" of molecules in a dynamic environment can be pictured similar to a flight simulator with a realistic physics engine. The user interactively directs the design of molecules. Instead of relying solely on the hybridisation rules (i.e., Watson-Crick or wobble base pairs), the mapping between structure and sequence is based on experimental thermodynamic data (i.e., the free energy measurement of complementary pairing and unpaired bases). In terms of physical simulation, we have adapted and refined the computational tools for predicting secondary structure of both single and interacting molecules which use empirical thermodynamic parameters. With the development of a sequence designer that is capable of generating sequences for molecules with multi-stable conformations, we have created a computational protocol (cf. Chapter. 6), to support the design of nucleic acid computing units. In order to fully realise the "orchestration" concept however, an interactive user interface would be required as part of the protocol to control the integration of these tools.

7.1.1 Interacting Multi-stable States Nucleic Acids

The design of nucleic acid sequences has received a lot of attention in bio-molecular computing (Brenneman and Condon, 2002; Condon, 2003). For instance, the implementation of DNA computers is always hindered by the problem of sequence design, despite the well known complementary base pairing property of DNA (Deaton et al., 1999; Braich et al., 2000; Penchovsky and Ackermann, 2003). The design of sequences that conform to a predefined target structure is essential for the construction of nucleic acid computing units. The mapping between RNA sequences and structures has been investigated thoroughly in (Reidys et al., 1997; Schuster, 2006) and tools such as *RNAinverse*, *RNAdesigner* and *INFO-RNA* are available to solve the inverse prediction problem of a single-state molecule.

Although, there is no tool available for the design of multi-stable conformation molecules, Flamm et al. (2001) suggest a multi-stable extension that can be implemented using any optimisation heuristics that is already available to handle the single state sequence design. Before we implemented the multi-stable model of Flamm et al. (2001) (Chapter 5), we evaluated the performance of single state sequence design algorithms (i.e., *RNAinverse*, *RNAdesigner*, *INFO-RNA*, *StochSrch*, and *INFO-RNA*) for the design space suitable computational nucleic acids (Chapter 4). Two simplified variants of *RNAdesigner* are proposed. First, called *StochSrch*, performed best for sequences with constraints on the choice of bases, assigned to ≈10% of the positions ("constrained setting"). The second, a rule based non-optimisation algorithm *RepInit* performed the best for sequences where all bases are permitted ("unconstrained setting"). In both unconstrained and constrained settings, the rule-based algorithm *RepInit* produced the highest number of accurate sequences with the lowest free energies. The performance of *RepInit* shows that for parts of the sequences the assignment of bases can be determined without iterative optimisation.

Based on the performance of *StochSrch* in the comparison study, we extended the algorithm (*StochSrchMulti*) to enable it to generate sequences for a single molecule with multi-stable conformations using the model suggested by Flamm et al. (2001). In addition, because the design space of interest is restricted to molecules that are at most 200 nt long and the constraints on the length and number of occurrences for their secondary structure elements are already identified, we developed a deterministic algorithm to solve the sequence design problem (Chapter 5 p. 85). A design algorithm, called *multiSrch*, was developed by combining the dynamic programming algorithm and a multi-objective sorting technique adapted from evolutionary programming. The *multiSrch* algorithm comprises four phases. First is the construction of dependency pathways to find the "intersection region" of base pairing and unpaired bases occurring in each conformation for all participating molecules. The "intersection region" represents positions that are paired differently in each conformation and positions with different states (form base pairs or unpaired) in at least one conformation (cf. Sec. 5.1 p. 88). This is followed by the initialisation of a complete base combination assignment based on to the type and length of each element selected from the list of dependency pathways. Next, the execution of the dynamic programming routine that for each pathway element minimises the value of the multi-objective function. This procedure is then coupled with a sorting mechanism that selects suitable sub-optimal candidates for the next iteration. Once the objective function is calculated, backtracking is executed to find both, the optimal sequence and suboptimal sequences for the target molecule.

During the dynamic programming routine, the selection of pathways from the dependency list is critical. In the default setting, the pathways are arranged in a descending order, where the longest path (i.e., path with the highest number of edges) is placed first. Optionally, the list can be shuffled instead to improve the diversity of the generated sequence. Because the algorithm is deterministic, without shuffling the list, for any target structure only a single set of sequences would be generated, as long as kept the number of suboptimal solutions (Λ) identical during each run. If for instance, the number of suboptimal increases (Λ) from 5 to 10, then the first five solutions that appeared earlier would still be present in the next set along with an additional five sequences. With the shuffling of the pathway list, the algorithm produces different sequences during each run. For the 300 candidate sequences that we used in the design of RNA logic gate (cf. Tab. 6.13, 6.14, 6.14 and 6.15), there are approximately 10 to 30 sequences that are present consistently for most of the 10 runs (for each type of logic gate).

Both cases of structural switching molecules (i.e., self-induced and trans-acting) were included for evaluating the performance of *StochSrchMulti* and *multiSrch*. For the self-induced test switches, *multiSrch* managed to produce a better set of sequences when compared to *StochSrchMulti* in terms of the free energy and energy gap (cf. Figs. 5.10 and 5.12). We then extended the test to include the trans-acting structural switches, which is more representative of our structural space of interest. The type of molecules selected as the datasets (DS-MS and DS-LG) consist of mostly logic gates constructed from DNA and RNA molecules that have been engineered and verified in the laboratory (cf. Tab. 5.1). With default settings the generated sequences from *multiSrch* have a significantly lower free energy then published sequences. However, by tuning the target value of ΔG_{opt} and E_{gap}, the generated sequences have free energies comparable with the experimentally verified sequences from the literature (cf. Tab. 5.5).

During the two comparison studies (TTS and TSS in Sec. 5.3 p. 104) of evaluating the performance of *multiSrch*, there is evidence that the implementation of a single aggregate objective function during minimisation of possible base combination assignments produced sequences that always biased the free energy term. For multiple objectives optimisation, no balance was achieved. The problem is visible in designing the trans-acting switches. Instead of only two terms (MFE and energy gap), the intermolecular binding efficiency (ΔG_{int}) is also measured. The algorithm immediately prunes poor candidates and because the intermolecular binding efficiency (ΔG_{int}) only contributes to the objective function in later stages of the iteration, candidates with better ΔG_{int} might have been eliminated early. In *multiSrch*, multiple sorting bins have been implemented to resolve this problem. The suboptimal solutions are ranked separately according to the term allocated for each bins. The top Λ suboptimal solutions from each bin is then selected for the next iteration. Suboptimal candidates that overlap are removed. At most, $n_b \times \Lambda$ number of suboptimal candidates are selected, where n_b denotes number of bins. With the implementation of the sorting bins, the algorithm eliminates the need of the weighting factors in the single aggregate objective function. Aside from the basic sorting option (i.e., arranging then suboptimal solutions in descending order), *multiSrch* implements a tournament ranking selection (Bäck et al., 2000), where the suboptimal candidates are compared in arbitrary pairs, and the candidates with the highest number of wins are ranked higher. The top Λ suboptimal candidates from each bins are then selected for the next iteration.

7.1.2 Computational Protocol for Constructing Computational Nucleic Acids

Secondary structure prediction tools have also been used in the design of nucleic acid computing units such as the DNA logic gates by Stojanovic et al. (2005) and Macdonald et al. (2006) and recently, in the construction of synthetic RNA devices by Win and Smolke (2008) and Beisel et al. (2008). Secondary structure folding simulators are required for most of the work related to the design of computational nucleic acids because these simulators provide the basic structural representation of the molecules. Without this structural representation, nucleic acids can only be seen as a linear string of bases, which conceals the properties of the molecules. However, only the work of Penchovsky and Breaker (2005) discuss the practicality of using computational tools to aid in the construction of these molecular information processors.

The protocol of Penchovsky and Breaker (2005) is confined to the construction of the four logic gate RNA gates (PASS, NOT, OR and AND). The protocol works in an automated manner, with fixed constraints being applied prior to the search procedure. By relaxing these constraints, we have created a more general protocol (P-ER1 in Tab. 6.3) that generate sequences with diverse structural conformations. Structural and sequence diversity is important because only some designs are applicable depending on the physico-chemical environment. The diverse set of solutions allow for the best design to be selected depending on the given conditions thus reducing the risk and cost of construction in the laboratory. Using the design of molecular PASS gates as test case, P-ER1 generated a more diverse set of structures and sequences compared to the protocol of Penchovsky and Breaker (2005). However the passing rate of the candidate sequences is low (Fig. 6.1). Although the design of computational nucleic acids is not intended for developing conventional logic gates, the design of conventional logic gates allows the performance of the protocol to be evaluated. P-ER2 (cf. Tab. 6.6) was then developed to design the complete binary logic operators.

The computational protocol of P-ER2 relies on prediction and sequence design of only single molecules. Therefore, the protocol failed to consider the multi-stable conformation characteristics of the binary logic gates. These binary logic gates have four meta-stable conformations. Despite generating a diverse set of sequences and structures, the passing rate of P-ER2 is identical to P-ER1 (cf. Tabs. 6.7 and 6.9). The availability of a multi-stable sequence design algorithm (*multiSrch*) allows the derivation of a new protocol named P-ERM (Tab. 6.11). The P-ERM protocol deals with the design of multiple molecules where each molecule has multi-stable conformations. In the P-ERM protocol, one is required to specify the partial conformations (cf. Fig. 6.6) of the molecule as input for *multiSrch*. A partial conformation represents the mandatory base pairing and unpaired base positions (i.e., structural constraints) that must be preserved during the generation of the candidate sequences. For constructing the complete set of binary logic operators, the P-ERM protocol significantly improved the quality of the candidate sequences. From the 300 candidate sequences that were generated for the least favourable design of logic gates, a minimum of ≈66% of the candidates still survived the filtering procedure. For favourable design of logic gates, the success rate is more than 82% (cf. Tabs. 6.13 and 6.14).

As discussed in Chapter. 6, there are “unfavourable” design of binary logic gates. For these unfavourable structural designs, the protocol would require a specific meta-stable conformations as input instead of the partial conformations. However, if only the partial conformations are available, one can steer the direction of the sequence design algorithm by tuning the target values of MFE and energy gap. This tuning facility allows the user to change the generation of candidate sequences into a different part of the sequence space (i.e., higher MFE or higher energy gap) which would be more suited to the target structures. The ability to tune the generation of candidate sequences is important in constructing computational nucleic acids. The generality of the protocol is briefly evaluated by designing a few samples of nucleic acid aptamer (Sec. 6.3.1). From the performance study, the P-ERM protocol is efficient because diversified set of sequences and structures were obtained in the binary logic gates study. The passing rate of the candidate sequences is good across all gates. The protocol is also effective, since only a single run is required to generate these solutions.

7.2 Conclusion

For decades, RNA molecules have been perceived as merely an intermediary in the translation of genetic information from DNA molecules into messenger RNA (mRNA) that later makes protein molecules. From the discovery of catalytic RNAs (Altman, 1990), noncoding RNAs (ncRNAs)[1] which perform various cellular functions (Eddy, 2001) and until recently the discovery of small regulatory RNAs (sRNA)[2] and riboswitches (Mandal and Breaker, 2004; Winkler and Breaker, 2005) that are responsible in the expression of gene regulation, interest in RNA molecules has steadily increased over the past two decades. This interest also led to the emergence of synthetic RNA devices with novel catalytic functions (Joyce, 2004) and functions not found in nature (Isaacs et al., 2006; Win and Smolke, 2008; Beisel et al., 2008). The construction of synthetic RNA-devices also drives the development of computational tools to aid the design of functional RNA while minimising the cost and complexity of handling these molecules in the laboratory.

The development of a computational protocol which enables the construction of functional nucleic acids that can act as a substrate for information processing is the focus in this thesis. In addition, a tool that is capable of generating nucleic acid sequences for the design of multi-stable conformation molecules was developed as part of the protocol. Many of the existing computational tools for nucleic acids focus on the mapping of nucleotide sequences to their secondary structure. The inverse which is the mapping of secondary structures to nucleotide sequences is of major importance in constructing computational nucleic acids. The length of nucleotide sequences composed of four bases in arbitrary order gives rise to a combinatorically large design space, in which random search without appropriate constraint would not be efficient in generating useful designs. The design space comprised of both, natural occurring and engineered small catalytic RNAs and DNAs enzymes was identified for the construction of computational nucleic acids. For the design space, existing single-state sequence design algorithms varied in performance, but generally the accuracy of sequences generated by these single-state design algorithms in the design space of interest is unsatisfactory. We then derived two design algorithms that are specifically tuned to the design space of interest. Results from our comparison study showed that the quality, in terms of accuracy and MFE of the sequences generated by *StochSrch* and *RepInit* is significantly better than the existing design algorithms.

The design of computational nucleic acids however, must consider molecules with multi-stable conformations. Furthermore, the design of a cascading network of computational nucleic acids must take the interactions among multiple molecules with multi-stable conformations into consideration. Tools for such tasks have not yet been developed. To fill this gap, we developed a deterministic sequence design algorithm (*multiSrch*)

[1] RNA molecules that do not encode protein.

[2] Distinct classes of sRNA includes small interfering RNA (siRNA) and micro-RNAs (miRNA) (Couzin, 2002; Novina and Sharp, 2004; Zamore and Haley, 2005).

for sets of molecules with multi-stable conformations. The algorithm is efficient in generating sequences that are not only accurate, but conform to the thermodynamic requirements (i.e., MFE and energy gap) specific for the design. The *multiSrch* algorithm is also effective as only a single run is needed to generate a set of sequences consist of both the optimal and suboptimal solutions. With the addition of *multiSrch*, we have demonstrated (in Chapter 6) the simplicity with which one can construct a set of nucleic acid computing units using our new computational protocol. To accomplish the task of constructing these nucleic acid computers, the protocol managed to produce a diverse set of structures and sequences based on the design constraints. Although, the *in-silico* construction of computational nucleic acids using the protocol does not guarantee their success in the laboratory, the protocol contributes in identifying possible candidate solutions for the actual implementation.

7.3 Future Directions

A concept of molecular computing using nucleic acid molecules that combines both the self-assembly and conformational dynamic paradigms is presented in this thesis. Nucleic acid computing units have been engineered by others using functional DNA molecules (Stojanovic and Stefanovic, 2003a; Stojanovic et al., 2002; Stojanovic and Stefanovic, 2003b) and RNA molecules (Penchovsky and Breaker, 2005) with the latter, constructed using a computational protocol that integrates some of the common RNA prediction tools. Although the majority of the existing work demonstrates the functionality of computational nucleic acids as logic operators, in recent applications, the concept has been extended to built synthetic RNA devices to be applied *in-vivo*, thus allowing the creation of RNA-based systems that regulate gene expression events (Isaacs et al., 2006; Beisel et al., 2008; Suess, 2005; Suess and Weigand, 2008) and a regulatory system for combinatorial gene regulation (Rinaudo et al., 2007).

As a direction for future work, an integration of the computational tools used in the protocol is required in order to create an interactive software that enables the user to monitor and provides immediate responses during the design process. For instance, creating an interactive user interface that would allow users to monitor the fitness of the generated sequences and change the target value of free energy or energy gap to steer the generated sequences into a more desirable part of the sequence space. This would allow for direct interruption to be made to reduce the processing time. This would also functions as a means of measuring the possibility of generating sequences for a particular conformation of the computing units. If a conformation is unfavourable, then a different strategy can be applied to the design.

A secondary structure prediction tool for RNA structure with pseudoknot motifs (Condon and Jabbari, 2009) can be considered in future work. This should allow for an increase in the design space of interest to include the design of natural occurring regulator (e.g., riboswitches) to be made. Accordingly, this would enable the construction of more complex RNA synthetic devices tailored to function in the intra cellular environment. For instance, a development of RNAs that bind specifically to sets of codon in messenger RNA, so called siRNAs.

Equally important is the prediction of secondary structure folding involving multiple molecules and the folding prediction for RNA-DNA interactions. An expansion of the single molecule folding prediction into multiple molecules is possible. However the recalculation of the thermodynamic parameters to include inter-molecular binding and the combinatorial aspect of determining the hybridisation between these molecules are some of the issues that need to be resolved, in order to implement the multiple molecules prediction tools. DNA effector is more rigid as compared to the RNA effector. The binding of DNA effector to its receptor site has been shown to trigger conformational change that activated the catalytic reaction of a number of allosterically controlled functional nucleic acids (Porta and Lizardi, 1995; Burke et al., 2002; Komatsu et al., 2000; Wang et al., 2002; Stojanovic and Stefanovic, 2003b; Penchovsky and Breaker, 2005). The availability of thermodynamics data for RNA and DNA binding is important in designing the computing units. This would allow more allosteric control strategies (as shown in Fig. 2.9) to be implemented.

For *multiSrch*, a more efficient multi-objective optimisation technique would be beneficial to resolve the pre-mature pruning of base assignments. The current method of employing a single aggregate objective function and sorting-bins are not yet effective. Using the findings from Chapter 4, there is a possibility of implementing a rule based approach to help assign and prune bases combinations in *multiSrch*. For this, an analysis of sequences for the interacting molecules with multi-stable conformation is required to find common base combinations for structural motifs. Rules can then be derived from this analysis. These rules aid during base assignment. The implementation of this rule based approach would eliminate the need for the dynamic programming optimisation, and could potentially reduce the processing time of the algorithm.

The application of nucleic acids in bioimmersive computation has the potential to open up interesting possibilities. For instance, using the strands of noncoding RNAs (ncRNA), one could try to develop regulatory units that harness their conformational switching (triggered after the introduction of an effector molecule, i.e., in this case, a short ncRNA) to create sticky ends that binds to a particular codon of mRNAs. The development of regulatory control points, such as a set of riboswitches or allosterically controlled nucleic acids are possible. For instance, using short ncRNAs as input, the set of allosterically controlled nucleic acids can be activated when a specific effector molecules are present and releases short RNA strand that binds to a specific codon in mRNA to block the production of harmful protein. Our computational protocol can support the design of nucleic acid computers that function as detection units, or the design of a network of regulatory control points. Smart drugs that can sense the internal state of cell and intervene in the intracellular regulatory mechanisms may come within reach (Benenson et al., 2004) and engineered molecular control mechanisms that can be integrated into cells would be a powerful tool for life-science research (Simpson, 2004).

Bibliography

J. P. Abrahams, M. v. d. Berg, E. v. Batenburg, and C. Pleij. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Research*, 18(10):3035–3044, 1990.

A. Adamatzky, B. De Lacy Costello, and T. Asai. *Reaction-Diffusion Computers.* Elsevier Science, Amsterdam, 2005.

L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell.* Garland Science, New York, fourth edition, 2002.

S. Altman. Enzymatic cleavage of RNA by RNA (Nobel lecture). *Angewandte Chemie International Edition*, 29:749–758, 1990.

M. Amarzguioui and H. Prydz. Hammerhead ribozymes design and application. *Cellular and Molecular Life Sciences*, 54:1175–1202, 1998.

L. W. Ancel and W. Fontana. Plasticity, evolvability and modularity in RNA. *Journal Experimental Zoology*, 288:242–283, 2000.

M. Andronescu. Algorithms for predicting the secondary structure of pairs and combinatorial sets of nucleic acid strands. Master's thesis, University of British Columbia, Vancouver, 2003.

M. Andronescu, R. Anguirre-Hernández, A. Condon, and H. H. Hoos. RNAsoft: A suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Research*, 31(13):3461–3422, 2003.

M. Andronescu, A. Condon, H. H. Hoos, D. H. Mathews, and K. Murphy. Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, 23 (13):i19–i28, 2007.

M. Andronescu, A. P. Fejes, F. Hutter, A. Condon, and H. H. Hoos. A new algorithm for RNA secondary structure design. *Journal of Molecular Biology*, 336(3):607–624, 2004.

M. Andronescu, Z. C. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *Journal of Molecular Biology*, 345:987–1001, 2005.

A. Avihoo and D. Barash. Shape similarity measures for the design of small RNA switches. *Journal of Biomolecular Structure & Dynamics*, 24(1):17–23, 2006.

T. Bäck, D. B. Fogel, and T. Michalewicz. *Evolutionary Computation 1.* Institute of Physics Publishing, Bristol and Philadelphia, 2000.

A. R. Banerjee, J. A. Jaeger, and D. H. Turner. Thermal unfolding of a group I ribozyme: the low-temperature transition is primarily disruption of tertiary structure. *Biochemistry*, 32(1):153–163, 1993.

R. Baron, O. Lioubashevski, E. Katz, T. Niazov, and I. Willner. Elementary arithmetic operations by enzymes : A model for metabolic pathway based computing. *Angew. Chem. Int. Ed.*, 45:1572–1576, 2006.

C. L. Beisel, T. S. Bayer, K. G. Hoff, and D. Smolke. Model-guided design of ligand-regulated RNAi for programmable control of gene expression. *Molecular Systems Biology*, 4(224):1–14, 2008.

G. Benedetti and S. Morosetti. A genetic algorithm to search for optimal and suboptimal RNA secondary structures. *Biophysical Chemistry*, 55:253–259, 1995.

Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro. An autonomous molecular computer for logical control of gene expression. *Nature*, 429:423–429, 2004.

J. M. Berg, J. L. Tymoczko, and L. Stryer. *Biochemistry.* W. H. Freeman and Company, New York, fifth edition, 2003.

S. H. Bernhart, H. Tafer, U. Mückstein, C. Flamm, P. F. Stadler, and I. L. Hofacker. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms for Molecular Biology*, 1(3), 2006. doi: 10.1186/1748-7188-1-3.

C. K. Biebricher and R. Luce. In vitro recombination and terminal elongation RNA by Qβ replicase. *EMBO Journal*, 11(13):5129–5135, 1992.

K. R. Birikh, P. A. Heaton, and F. Eckstein. The structure, function and application of the hammerhead ribozyme. *Eur. J. Biochemistry*, 245:1–16, 1997.

B. J. Blencowe and M. Khanna. Molecular biology: RNA in control. *Nature*, 447: 391–393, 2007.

V. A. Bloomfield, D. M. Crothers, and I. Tinoco, Jr. *Nucleic Acids: Structures, Properties and Functions.* University Science Books, California, first edition, 2000.

R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, and L. M. Adleman. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 296:499–502, 2002.

R. S. Braich, C. Johnson, P. W. K. Rothemund, D. Hwang, N. Chelyapov, and L. M. Adleman. Solution of a satisfiability problem on a gel-based DNA computer. In A. Condon and G. Rozenberg, editors, *6th International Workshop on DNA-Based Computers: DNA Computing*, volume 2054 of *LNCS*, pages 27–42. Springer, 2000.

R. R. Breaker. DNA enzymes. *Nature Biotechnology*, 15:427–431, 1997.

R. R. Breaker. Engineered allosteric ribozymes as biosensor components. *Current Opinion in Biotechnology*, 13:31–39, 2002.

R. R. Breaker and G. F. Joyce. A DNA enzyme that cleaves RNA. *Chemistry & Biology*, 1:223–229, 1994a.

R. R. Breaker and G. F. Joyce. Inventing and improving ribozyme function: Rational design versus iterative selection methods. *Trends in Biotechnology*, 12:268–275, 1994b.

R. R. Breaker and G. F. Joyce. A DNA enzyme with Mg^{2+}-dependent RNA phosphoesterase activity. *Chemistry & Biology*, 2:655–660, 1995.

A. Brenneman and A. Condon. Strand design for biomolecular computation. *Theoretical Computer Science*, 287:39–58, 2002.

P. Brion and E. Westhof. Hierarchy and dynamics of RNA folding. *Annual Review of Biophysics and Biomolecular Structure*, 26:113–137, 1997.

D. H. Burke, N. D. S. Ozerova, and M. Nilsen-Hamilton. Allosteric hammerhead ribozyme TRAPs. *Biochemistry*, 41:6588–6594, 2002.

A. Busch and R. Backofen. INFO-RNA—a fast approach to inverse RNA folding. *Bioinformatics*, 22(15):1823–1831, 2006.

T. R. Cech. The chemistry of self-splicing RNA and RNA enzymes. *Science*, 236(4808): 1532–1539, 1987.

D. M. Chadalavada, S. M. Knudsen, S. Nakano, and P. C. Bevilacqua. A role for upstream RNA structure in facilitating the catalytic fold of the genomic Hepatitis Delta Virus ribozyme. *Journal of Molecular Biology*, 301:349–367, 2000.

K. B. Chapman and J. W. Szostak. In vitro selection of catalytic RNAs. *Current Opinion in Structural Biology*, 4:618–622, 1994.

A. Condon. Problems on RNA secondary structure prediction and design. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Automata, Languages and Programming: 30th International Colloquium, ICALP 2003 Eindhoven, The Netherlands, June 30–July 4, 2003 Proceedings*, volume 2719 of *LNCS*, pages 193–204. Springer, 2003.

A. Condon, B. Daby, B. Rastegari, S. Zhao, and F. Tarrant. Classifying RNA pseudoknotted structures. *Theoretical Computer Science*, 320(1):35–50, 2004.

A. Condon and H. Jabbari. Computational prediction of nucleic acid secondary structure: Methods, applications, and challenges. *Theoratical Computer Science*, 410: 294–301, 2009.

M. Conrad. Self-assembly as a mechanism of molecular computing. In *Proceedings of the 11th Annual International IEEE-EMBS Conference*, pages 1354–1355, Piscataway, NJ, 1989. IEEE.

M. Conrad. Molecular computing: The lock-key paradigm. *Computer (IEEE)*, 25(11): 11–20, 1992.

M. Conrad. Emergent computation through self-assembly. *Nanobiology*, 2:5–30, 1993.

J. Couzin. Small RNAs make big splash. *Science*, 298:2296–2297, 2002.

D. M. Crothers, P. E. Cole, C. W. Hilbers, and R. G. Shulman. The molecular mechanism of thermal unfolding of *Escherichia coli* formylmethionine transfer RNA. *Journal of Molecular Biology*, 87(1):63–88, 1974.

R. Deaton, C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr. Good encodings for DNA-based solutions to combinatorial problems. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 44:247–258, 1999.

A. Dietrich and W. Been. Memory and DNA. *Journal of Theoratical Biology*, 208(2): 145–149, 2001.

R. A. Dimitrov and M. Zuker. Prediction of hybridization and melting for double-stranded nucleic acids. *Biophysical Journal*, 87:215–226, 2004.

R. M. Dirks and N. A. Pierce. A partition function algorithm for nucleic acids secondary structure including pseudoknots. *Journal of Computational Chemistry*, 24(13):1664–1677, 2003.

E. A. Doherty and J. A. Doudna. Ribozyme structures and mechanisms. *Annual Review of Biochemistry*, 69:597–615, 2000.

J. A. Doudna and T. R. Cech. The chemical repertoire of natural ribozymes. *Nature*, 418:222–228, 2002.

D. E. Draper. Strategies for RNA folding. *Trends in Biochemistry Sciences*, 21:145–149, 1996.

S. R. Eddy. Non-coding RNA genes and the modern RNA world. *Nature Review Genetics*, 2:919–929, 2001.

S. R. Eddy. What is dynamic programming? *Nature Biotechnology*, 22(7):909–910, 2004.

A. D. Ellington and J. W. Szostak. In vitro selection of RNA molecule that bind specific ligands. *Nature*, 346:818–822, 1990.

G. M. Emilsson and R. R. Breaker. Deoxyribozymes: New activities and new applications. *Cellular and Molecular Life Sciences*, 59:596–607, 2002.

M. Famulok and J. W. Szostak. In vitro selection of specific ligand-binding nucleic acids. *Angewandte Chemie International Edition*, 31:979–988, 1992.

M. J. Fedor. Structure and function of the hairpin ribozyme. *Journal of Molecular Biology*, 297:269–291, 2000.

M. J. Fedor and J. R. Williamson. The catalytic diversity of RNAs. *Molecular Cell Biology*, 6:399–412, 2005.

A. Fernández, T. Burastero, R. Salthú, and A. Tablar. Energy-level statistics in the fine conformational resolution of RNA folding dynamics. *Physical Review E*, 60(5): 5888–5893, 1999.

A. R. Ferré-D'Amaré, K. Zhou, and J. A. Doudna. Crystal structure of a hepatitis delta virus ribozyme. *Nature*, 395:567–574, 1998.

C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.

C. Flamm, Ivo. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, 2001.

A. C. Forster and R. H. Symons. Self-cleavage of plus and minus RNAs of a virusoid and a structural model for the active sites. *Cell*, 49:211–220, 1987.

T. Franch, A. P. Gultyaev, and K. Gerdes. Programmed cell death by *hok/sok* of plasmid R1: Processing at the *hok* mRNA 3-end triggers structural rearrangements that allow translation and antisense RNA binding. *Journal of Molecular Biology*, 273(1):38–51, 1997.

R. Geigerich, D Haase, and M. Rehmsmeier. Prediction and visualization of structural switches in RNA. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klien, editors, *Proceedings of the Pacific Symposium on Biocomputing*, volume 4, pages 126–137. World Scientific Press, 1999.

R. Giegerich and D. J. Evers. RNA movies: visualizing RNA secondary structure spaces. *Bioinformatics*, 15(1):32–37, 1999.

W. Gilbert. Origin of life: The RNA world. *Nature*, 319:618, 1986.

J. Goodchild. Enhancement of ribozyme catalytic activity by a contiguous oligodeoxynucleotide (facilitator) and by 2'-O-methylation. *Nucleic Acids Research*, 20(17):4607–4612, 1992.

A. P. Gultyaev, T. Franch, and K. Gerdes. Programmed cell death by *hok/sok* of plasmid R1: Coupled nucleotide covariations reveal a phylogenetically conserved folding pathway in the *hok* family of mRNAs. *Journal of Molecular Biology*, 273(1):26–37, 1997.

R. R. Gutell, J. C. Lee, and J. J. Cannone. The accuracy of ribosomal RNA comparative structure models. *Current Opinion in Structural Biology*, 12(3):301–310, 2002.

N. Hampp. Bacteriorhodopsin as a photochromic retinal protein for optical memories. *Chem. Rev.*, 100:1755–1776, 2000.

G. J. Hannon. RNA interference. *Nature*, 418:244–251, 2002.

C. E. Heitsch, A. Condon, and H. H. Hoos. From RNA secondary structure to coding theory: A combinatorial approach. In *Proceedings of the 8th International Workshop on DNA Based Computers, LNCS 2568*, pages 215–218, 2003.

P. Hendry and M. McCall. Unexpected anisotropy in substrate cleavage rates by asymmetric hammerhead ribozymes. *Nucleic Acids Research*, 24(14):2679–2684, 1996.

P. Hendry, M. J. McCall, and T. J. Lockett. Influence of helix length on cleavage efficiency of hammerhead ribozymes. *Australian Journal of Chemistry*, 58:851–858, 2005.

K. J. Hertel, D. Herschlag, and O. C. Uhlenbeck. A kinetic and thermodynamic framework for the hammerhead ribozyme reaction. *Biochemistry*, 33:3374–3385, 1994.

K. J. Hertel, D. Herschlag, and O. C. Uhlenbeck. Specificity of hammerhead ribozyme cleavage. *EMBO Journal*, 25(14):3751–3757, 1996.

P. G. Higgs. RNA secondary structure: a comparison of real and random sequences. *Journal de Physique I*, 3:43–59, 1993.

P. G. Higgs. Thermodynamic properties of transfer RNA: A computational study. *J. Chem. Soc. Faraday Trans*, 9(16):2531–2540, 1995.

P. G. Higgs. RNA secondary structure: physical and computational aspects. *Quarterly Reviews of Biophysics*, 33(3):199–253, 2000.

P. G. Higgs and S. R. Morgan. *Advances in Artificial Life*, chapter Thermodynamics of RNA folding. When is an RNA molecule in equilibrium?, pages 852–861. Lecture Notes in Computer Science. Springer, Berlin / Heidelberg, 1995.

I. L. Hofacker. *A statistical characterization of the sequence to structure mapping in RNA*. PhD thesis, University Wien, Wien, 1994.

I. L. Hofacker. Vienna RNA secondary structure package. `http://www.tbi.univie.ac.at/~ivo/RNA/`, 2007. Last Accessed: 16 May 2007.

I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Chemical Monthly*, 125 (2):167–188, 1994.

S. Hohng, T. J. Wilson, E. Tan, R. M. Clegg, D. M. J. Lilley, and T. Ha. Conformational flexibility of four-way junctions in RNA. *J. Mol. Biol.*, 336:69–79, 2004.

F. J. Isaacs, D. J. Dwyer, and J. J. Collins. RNA synthetic biology. *Nature Biotechnology*, 24:545–554, 2006.

J. A. Jaeger, D. H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. In *Proceedings of the National Academy of Sciences, USA*, volume 86, pages 7706–7710, 1989.

G. F. Joyce. Directed molecular evolution. *Scientific American*, 267(6):48–55, 1992.

G. F. Joyce. Directed evolution of nucleic acid enzymes. *Annual Review of Biochemistry*, 73:791–836, 2004.

K. Kawasaki. Diffusion constants near the critical point for time-tependent ising models. I. *Physical Review*, 145(1):224–230, 1966.

M. Koizumi, G. A. Soukup, J. N. Q. Kerr, and R. R. Breaker. Allosteric selection of ribozymes that respond to the second messengers cGMP and cAMP. *Nature Structural Biology*, 6(11):1062–1071, 1999.

D. M. Kolpashchikov and M. S. Stojanovic. Boolean control of aptamer binding states. *Journal of the American Chemical Society*, 127:11348–11351, 2005.

Y. Komatsu, S. Yamashita, N. Kazama, K. Nobuoka, and E. Ohtsuka. Construction of new ribozymes requiring short regulator oligonucleotides as a cofactor. *Journal of Molecular Biology*, 229:1231–1243, 2000.

D. A. M. Konings and R. R. Gutell. A comparison of thermodynamic foldings with comparatively derived structures of 16S and 16S-like rRNAs. *RNA*, 1:559–574, 1995.

T. Kuwabara, M. Warashina, and K. Taira. Allosterically controllable ribozymes with biosensor functions. *Current Opinion in Chemical Biology*, 4:669–677, 2000.

D. A. Lafontaine, D. G. Norman, and D. M. J. Lilley. Structure folding and activity of the VS ribozyme: Importance of the 2-3-6 helical junction. *EMBO Journal*, 20(6): 1415–1424, 2001.

H. Lederman, J. Macdonald, D. Stefanovic, and M. Stojanovic. Deoxyribozyme-based three input logic gates and construction of a molecular full adder. *Biochemistry*, 45: 1194–1199, 2006.

E. A. Liberman. Analog-digital molecular cell computer. *BioSystems*, 11:111–124, 1979.

A. Lieber and M. Strauss. Selection of efficient cleavage site in target RNAs by using a ribozyme expression library. *Molecular and Cellular Biology*, 15(1):540–551, 1995.

D. M. J. Lilley. Structure, folding and catalysis of the small nucleolytic ribozymes. *Current Opinion in Structural Biology*, 9:330–338, 1999.

D. M. Long and O. C. Uhlenbeck. Self-cleaving catalytic RNA. *FASEB Journal*, 7: 25–30, 1993.

R. B. Lyngsø, M. Zuker, and C. N. S. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999a.

R. B. Lyngsø, M. Zuker, and C. N. S. Pedersen. Internal loops in RNA secondary structure prediction. In *Proc. 3rd Int. Conf. Computational Molecular Biology (RECOMB '99)*, pages 260–267, 1999b.

J. Macdonald, Y. Li, M. Sutovic, H. Lederman, K. Pendri, W. Lu, B. L. Andrews, D. Stefanovic, and M. N. Stojanovic. Medium scale integration of molecular logic gates in an automaton. *Nano Letters*, 6(11):2598–2603, 2006.

M. Mandal and R. R. Breaker. Gene regulation by riboswitches. *Molecular cell biology*, 5:451–463, 2004.

C. Mao, T. H. LaBean, J. H. Reif, and N. C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, 407:493–496, 2000.

N. R. Markham and M. Zuker. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acid Research*, 33:W577–W581, 2005. doi:10.1093/nar/gki591.

D. H. Mathews. Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*, 10:1178–1190, 2004.

D. H. Mathews. Predicting RNA secondary structure by free energy minimization. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling*, 116:160–168, 2005.

D. H. Mathews, M. E. Burkard, S. M. Freier, J. R. Wyatt, and D. H. Turner. Predicting oligonucleotide affinity to nucleic acid targets. *RNA*, 5:1458–1469, 1999a.

D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. In *Proceedings of the National Academy of Sciences, USA*, volume 101, pages 7287–7292, 2004.

D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999b.

D. H. Mathews, S. J. Schroeder, D. H. Turner, and M. Zuker. Predicting RNA secondary structure. In R. F. Gesteland, T. R. Cech, and J. F. Atkins, editors, *RNA world*, pages 631–657. Cold Spring Harbor Laboratory Press, New York, third edition, 2006.

O. Matzura and A. Wennborg. RNAdraw: an integrated program for RNA secondary structure calculation and analysis under 32-bit microsoft windows. *Computer Applications in the Biosciences (CABIOS)*, 12(1):247–249, 1996.

M. J. McCall, P. Hendry, and P. A. Jennings. Minimal sequence requirements for ribozyme activity. In *Proceedings of the National Academy of Sciences, USA*, volume 89, pages 5710–5714, 1992.

J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6): 1087–1091, 1953.

K. U. Mir. A restricted alphabet for DNA computing. In L. F. Landweber and E. B. Baum, editors, *DNA Based Computers II, DIMACS Workshop, June 10-12, 1996*, volume 44 of *DIMACS Series of Discrete Mathematics and Theoratical Computer Science*, pages 243–246. American Mathematical Society, 1996.

J. Møller-Jensen, T. Franch, and K. Gerdes. Temporal translational control by a metastable RNA structure. *Journal of Biological Chemistry*, 276(38):35707–35713, 2001.

P. B. Moore. Structural motifs in RNA. *Annual Review of Biochemistry*, 68:287–300, 1999.

S. R. Morgan and P. G. Higgs. Evidence for kinetics effects in the folding of large RNA molecules. *Journal of Chemical Physics*, 105(16):7152–7157, 1996.

S. R. Morgan and P. G. Higgs. Barrier heights between ground states in a model of RNA secondary structure. *Journal of Physics A: Mathematical & General*, 31:3153–3170, 1998.

U. Mückstein, H. Tafer, J. Hackermüller, S. Bernhard, P. F. Stadler, and I. L. Hofacker. Thermodynamics of RNA-RNA binding. *Bioinformatics*, 22(10):1177–1182, 2006. doi:10.1093/bioinformatics/btl024.

K. Nagai and L. W. Mattaj. *RNA-Protein Interactions*. Oxford University Press, New York, first edition, 1994.

J. H. A. Nagel and C. W. A. Pleij. Self-induced structural switches in RNA. *Biochimie*, 84:913–923, 2002.

P. Nelson, M. Kiriakidou, A. Sharma, E. Maniataki, and Z. Mourelatos. The microRNA world: small is mighty. *Trends in Biochemical Sciences*, 28(10):534–540, 2003.

C. D. Novina and P. A. Sharp. The RNAi revolution. *Nature*, 430:161–164, 2004.

J. Nowakowski and I. Tinoco, Jr. RNA structure and stability. *Seminars in Virology*, 8: 153–165, 1997.

E. Nudler. Flipping riboswitches. *Cell*, 126:19–22, 2006.

E. Nudler and A. S. Mironov. The riboswitch control of bacterial metabolism. *Trends in biochemical sciences*, 29(1):11–17, 2004.

R. Nussinov and B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. In *Proceedings of the National Academy of Sciences, USA*, volume 77(11), pages 6309–6313, 1980.

J. Pan, M. Deras, and S. A. Woodson. Fast folding of a ribozyme by stabilizing core interactions: evidence for multiple folding pathways in RNA. *Journal of Molecular Biology*, 296(1):133–144, 2000.

J. Pan and S. A. Woodson. Folding intermediates of a self-splicing RNA: mispairing of the catalytic core. *Journal of Molecular Biology*, 280(4):597–609, 1998.

T. Pan, X. Fang, and T. Sosnick. Pathway modulation, circular permutation and rapid RNA folding under kinetic control. *Journal of Molecular Biology*, 286(3):721–731, 1999.

N. Paul, G. Springsteen, and F. Joyce. Conversion of a ribozyme to a deoxyribozyme through in vitro evolution. *Chemistry & Biology*, 13(3):329–338, 2006.

R. Pei, S. K. Taylor, D. Stefanovic, S. Rudchenko, T. E. Mitchell, and M. N. Stojanovic. Behaviour of polycatalytic assemblies in a substrate-displaying matrix. *Journal of the American Chemical Society*, 128:12693–12699, 2006.

R. Penchovsky and J. Ackermann. DNA library design for molecular computation. *Journal of Computational Biology*, 10(2):215–229, 2003.

R. Penchovsky and R. R. Breaker. Computational design and experimental validation of oligonucleotide-sensing allosteric ribozymes. *Nature Biotechnology*, 23(11):1424–1433, 2005.

C. P. Petersen, J. G. Doench, A. Grishok, and P. A. Sharp. The biology of short RNAs. In R. F. Gesteland, T. R. Cech, and J. F. Atkins, editors, *RNA world*, pages 535–565. Cold Spring Harbor Laboratory Press, New York, third edition, 2006.

M. C. Petty, M. R. Bryce, and D. Bloor, editors. *An Introduction to Molecular Electronics*. Oxford University Press, Oxford, 1995.

D. Pörschke. Model calculations on the kinetics of oligonucleotide double helix coil transitions. Evidence for a fast chain sliding reaction. *Biophysical Chemistry*, 2(2): 83–96, 1974.

D. Porschke, J. M. Burke, and N. G. Walter. Global structure and flexibility of hairpin ribozymes with extended terminal helices. *Journal of Molecular Biology*, 289:799–813, 1999.

H. Porta and P. M. Lizardi. An allosteric hammherhead ribozyme. *Bio/Technology*, 13: 161–164, 1995.

E. Puerta-Fernández, C. Romero-López, A. Barroso-delJesus, and A. Berzal-Herranz. Ribozymes: Recent advances in the development of RNA tools. *FEMS Microbiology Reviews*, 27:75–97, 2003.

J. D. Puglisi, J. R. Wyatt, and I. Tinoco, Jr. RNA pseudoknots. *Accounts of Chemical Research*, 25(5):152–158, 1991.

E. I. Ramlan and K.-P. Zauner. An extended dot-bracket-notation for functional nucleic acids. In E. Csuhaj-Varjú, R. Freund, M. Oswald, and K. Salomaa, editors, *International Workshop in Computing with Biomolecules, Wien, Austria, August 27, 2008*, pages 75–86. Österreichische Computer Gesellschaft, 2008. ISBN 978-3-85403-244-1.

E. I. Ramlan and K.-P. Zauner. Nucleic acid enzymes: The fusion of self-assembly and conformational computing. *International Journal of Unconventional Computing*, 5(2): 165–189, 2009.

J. Reeder, M. Höchsmann, M. Rehmsmeier, B. Voss, and R. Giegerich. Beyond Mfold: recent advances in RNA bioinformatics. *Journal of Biotechnology*, 124:41–55, 2006.

M. Rehmsmeier, P. Steffen, M. Höchsmann, and R. Giegerich. Fast and effective prediction of microRNA/target duplexes. *RNA*, 10:1507–1517, 2004.

J. Reichert, A. Jabs, P. Slicker, and J. Sühnel. The IMB Jena image library of biological macromolecules. *Nucleic Acids Research*, 28:246–249, 2000.

J. Reichert and J. Sühnel. The IMB Jena image library of biological macromolecules: 2002 update. *Nucleic Acids Research*, 30:253–254, 2002.

C. Reidys, P. F. Stadler, and P. Schuster. Generic properties of combinatory maps: Neutral networks of RNA secondary structures. *Bulletin of Mathematical Biology*, 59 (2):339–397, 1997.

K. Rinaudo, L. Bleris, R. Maddamsetti, S. Subramanian, R. Weiss, and Y. Benenson. A universal RNAi-based logic evaluator that operates in mammalian cells. *Nature Biotechnology*, 25(7):795–801, 2007.

E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285(5):2053–2068, 1999.

P. J. Rousseeuw, I. Ruts, and J. W. Tukey. The bagplot: a bivariate boxplot. *The American Statistician*, 53:382–387, 1999.

R. Russell and D. Herschlag. Probing the folding lanscape of the *Tetrahymena* ribozymel commitment to form the native conformation is late in the folding pathway. *Journal of Molecular Biology*, 308:839–851, 2001.

J. SantaLucia, Jr. and D. Hicks. The thermodynamics of DNA structural motifs. *Annual Review of Biomolecular Structure*, 33:415–440, 2004.

J. SantaLucia, Jr. and D. H. Turner. Measuring the thermodynamics of RNA secondary structure formation. *Biopolymers*, 44(3):309–319, 1997.

S. W. Santoro and G. F. Joyce. A general purpose RNA-cleaving DNA enzyme. In *Proceedings of the National Academy of Sciences, USA*, volume 94, pages 4262–4266, 1997.

M. Schmitz and G. Steger. Description of RNA folding by "Simulated Annealing". *Journal of Molecular Biology*, 255:254–266, 1996.

E. Schultes, P. T. Hraber, and T. H. Labean. A parameterization of RNA sequence space. *Complexity*, 4(4):61–71, 1998.

E. A. Schultes and D. P. Bartel. One sequence, two ribozymes: Implications for the emergence of new ribozyme folds. *Science*, 289(5478):448–452, 2000.

P. Schuster. Prediction of RNA seconday structures: from theory to models and real molecules. *Reports on Progress in Physics*, 69:1419–1477, 2006.

P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker. From sequence to shapes and back: A case study in RNA secondary structures. In *Proceedings of the Royal Society B: Biological Sciences*, volume 255, pages 279–284, 1994.

W. G. Scott. Biophysical and biochemical investigations of RNA catalysis in the hammerhead ribozyme. *Quaterly Reviews of Biophysics*, 32(3):241–284, 1999.

W. G. Scott, J. B. Murray, J. R. P. Arnold, B. L. Stoddard, and A. Klug. Capturing the structure of catalytic RNA intermediate: the hammerhead ribozyme. *Science*, 274: 2065–2069, 1996.

N. C. Seeman. DNA in a material world. *Nature*, pages 427–431, 2003.

B. A. Shapiro, D. Bengali, W. Kasprzak, and J. C. Wu. RNA folding pathway functional intermediates: Their prediction and analysis. *Journal of Molecular Biology*, 312:27–44, 2001.

E. Shapiro and B. Gil. RNA computing in a living cell. *Science*, 322:387–388, 2008.

S. K. Silverman. Rube Goldberg goes (ribo)nuclear? Molecular switches and sensors made from RNA. *RNA*, 9:377–383, 2003.

M. L. Simpson. Rewiring the cell: synthetic biology moves towards higher functional complexity. *Trends in Biotechnology*, 22(11):555–557, 2004.

G. A. Soukup and R. R. Breaker. Nucleic acid molecular switches. *Trends in Biotechnology*, 17:469–476, 1999.

P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4): 500–503, 2006.

M. N. Stojanovic, P. dePrada, and D. W. Landry. Catalytic molecular beacons. *Chembiochem*, 2(6):411–415, 2001.

M. N. Stojanovic, T. E. Mitchell, and D. Stefanovic. Deoxyribozyme-based logic gates. *Journal of the American Chemical Society*, 124:3555–3561, 2002.

M. N. Stojanovic, D. B. Nikic, and D. Stefanovic. Implicit-OR tiling of deoxyribozymes: Construction of molecular scale OR, NAND and four-input logic gates. *J. Serb. Chem. Soc.*, 68(4–5):321–326, 2003.

M. N. Stojanovic, S. Semova, D. Kolpashchikov, J. Macdonald, C. Morgan, and D. Stefanovic. Deoxyribozyme-based ligase logic gates and their initial circuits. *Journal of the American Chemical Society*, 127:6914–6915, 2005.

M. N. Stojanovic and D. Stefanovic. Deoxyribozyme-based half-adder. *Journal of the American Chemical Society*, 125:6673–6676, 2003a.

M. N. Stojanovic and D. Stefanovic. A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21(9):1069–1074, 2003b.

B. Suess. Engineered riboswitches control gene expression by small molecules. *Biochemical Society Transactions*, 33:474–476, 2005.

B. Suess and J. E. Weigand. Engineered riboswitches: overview, problems and trends. *RNA Biology*, 5:24–29, 2008.

J. Sühnel. Jena library of biological macromolecules. `http://www.fli-leibniz.de/cgi-bin/ImgLib.pl?CODE=URX057`, 2008. Last Accessed: 22 December 2008.

R. H. Symons. Small catalytic RNAs. *Annual Review of Biochemistry*, 61:641–671, 1992.

R. H. Symons. Plant pathogenic RNAs and RNA catalysis. *Nucleic Acids Research*, 25 (14):2683–2689, 1997.

E. t. Dam, K. Pleij, and D. Draper. Structural and functional aspects of RNA pseudoknots. *Biochemistry*, 31(47):11665–11676, 1992.

J. Tang and R. R. Breaker. Rational design of allosteric ribozymes. *Chemistry & Biology*, 4:453–459, 1997.

J. Tang and R. R. Breaker. Structural diversity of self-cleaving ribozymes. In *Proceedings of the National Academy of Sciences, USA*, volume 97, pages 5784–5789, 2000.

N. K. Tanner. Ribozymes: The characteristics and properties of catalytic RNAs. *FEMS Microbiology Reviews*, 23:257–275, 1999.

I. Tinoco, Jr. and C. Bustamante. How RNA folds. *Journal of Molecular Biology*, 293: 271–281, 1999.

D. K. Treiber, M. S. Rook, P. P. Zarrinkar, and J. R. Williamson. Kinetic intermediates trapped by native interactions in RNA folding. *Science*, 279(5358):1943–1946, 1998.

D. K. Treiber and J. R. Williamson. Exposing the kinetic traps in RNA folding. *Current Opinion in Structural Biology*, 9(3):339–345, 1999.

D. Tulpan, M. Andronescu, S. B. Chang, M. R. Shortreed, A. Condon, H. H. Hoos, and L. M. Smith. Thermodynamically based DNA strand design. *Nucleic Acids Research*, 33(15):4951–4964, 2005.

T. Tuschl, J. B. Thomson, and F. Eckstein. RNA cleavage by small catalytic RNAs. *Current Opinion in Structural Biology*, 5:296–302, 1995.

N. Usman, L. Beigelman, and J. A. McSwiggen. Hammerhead ribozyme engineering. *Current Opinion in Structural Biology*, 1:527–533, 1996.

F. H. D. van Batenburg and C. W. A. Pleij. An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *J. theor. Biol*, 174:269–280, 1995.

G. Varani and W. H. McClain. The G·U wobble base pair. *EMBO reports*, 1(1):18–23, 2000.

N. G. Walter and J. M. Burke. The hairpin ribozyme: Structure, assembly and catalysis. *Current Opinion in Chemical Biology*, 2:24–30, 1998.

D. Y. Wang, H. Y. Lai, A. R. Feldman, and D. Sen. A general approach for the use of oligonucleotide effectors to regulate the catalysis of RNA-cleaving ribozymes and DNAzymes. *Nucleic Acids Research*, 30(8):1735–1742, 2002.

M. S. Waterman. Secondary structure of single-stranded nucleic acids. *Studies in Foundations and Combinatorics, Advances in Mathematics, Supplementary Studies*, 1:167–212, 1978.

M. S. Waterman and T. H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences*, 77:179–188, 1985.

M. S. Waterman and T. F. Smith. RNA secondary structure: A complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

J. E. Wedekind and D. B. McKay. Crystal structure of lead-dependent ribozyme revealing metal binding sites relvant to catalysis. *Nature Structural Biology*, 6(3):261–268, 1999.

M. N. Win and C. D. Smolke. Higher-order celular information processing with synthetic RNA devices. *Science*, 322:456–460, 2008.

E. Winfree. Algorithmic self-assembly of DNA: Theoretical motivations and 2d assembly experiments. *J. of Biomolecular Structure & Dynamics*, 11(2):263–270, 2000.

W. C. Winkler and R. R. Breaker. Regulation of bacterial gene expression by riboswitches. *Annual. Review of Microbiology*, 59:487–517, 2005.

M. Wu and I. Tinoco, Jr. RNA folding causes secondary structure rearrangement. In *Proceedings of the National Academy of Sciences, USA*, volume 95, pages 11555–11560, 1998.

S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structure. *Biopolymers*, 49:145–165, 1999.

J. Wuyts, Y. V. de Peer, T. Winkelmans, and R. D. Wachter. The european database on small subunit ribosomal RNA. *Nucleic Acids Research*, 30(1):183–185, 2002.

T. Xia, J. SantaLucia, Jr., M. E. Bunkard, R. Kierzek, S. J. Schroeder, X. Jiao, C. Cox, and D. H. Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37:14719–14735, 1998.

H. Zamora, R. Luce, and C. K. Biebricher. Design of artificial short-chained RNA species that are replicated by Qβ replicase. *Biochemistry*, 34(4):1261–1266, 1995.

P. D. Zamore and B. Haley. Ribo-gnome: The big world of small RNAs. *Science*, 309: 1519–1524, 2005.

K.-P. Zauner. From prescriptive programming of solid-state devices to orchestrated self-organisation of informed matter. In J.-P. Banâtre, J.-L. Giavitto, P. Fradet, and O. Michel, editors, *Proceedings of UPP 2004, Unconventional Programming Paradigms, 15–17 September, Le Mont Saint-Michel, France*, volume 3566 of *LNCS*, pages 47–55. ERCIM, Springer, 2005a.

K.-P. Zauner. Molecular information technology. *Critical Reviews in Solid State and Material Sciences*, 30(1):33–69, 2005b.

K.-P. Zauner and M. Conrad. Enzymatic pattern processing. *Naturwissenschaften*, 87: 360–362, 2000.

K.-P. Zauner and M. Conrad. Enzymatic computing. *Biotechnology Progress*, 17(3): 553–559, 2001.

M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.

M. Zuker and A. B. Jacobson. 'Well-determined' regions in RNA secondary structure prediction: analysis of small subunit ribosomal RNA. *Nucleic Acids Research*, 23(14): 2791–2798, 1995.

M. Zuker, J. A. Jaeger, and D. H. Turner. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Research*, 19(10):2707–2714, 1991.

M. Zuker, D. H. Mathews, and D. H. Turner. *RNA Biochemistry and Biotechnology*, chapter Algorithm and Thermodynamics for RNA Secondary Structure Prediction: A Practical Guide, pages 11–43. NATO ASI. Kluwer Academic Publishers, Dordrecht, 1999.

M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 1984.

M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamic and auxiliry information. *Nucleic Acids Research*, 9(1):133–148, 1981.