

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **ABSTRACT**

### **AMINORMOTIFFINDER – A GRAPH GRAMMAR BASED TOOL TO EFFECTIVELY SEARCH A MINOR MOTIFS IN 3D RNA MOLECULES**

**by**  
**Ankur Malhotra**

RNA Motifs are three dimensional folds that play important role in RNA folding and its interaction with other molecules. They basically have modular structure and are composed of conserved building blocks dependent upon the sequence. Their automated in silico identification remains a challenging task. Existing motif identification tools does not correctly identify motifs with large structure variations. Here a “graph rewriting” based method is proposed to identify motifs in real three dimensional structures. The unique encoding of A Minor Searcher takes into consideration the non canonical base pairs and also multipairing of RNA structural motifs. The accuracy is demonstrated by correctly predicting A minor motifs across many PDB files with zero false positives.

There is a huge demand of a good well developed RNA Motif identification algorithm that would successfully identify both canonical / non canonical and isomorphic motifs. In this thesis, a novel encoding algorithm is demonstrated that successfully identifies RNA A Minor Motifs from 3D RNAs. The algorithm encodes the three dimensional RNA Data into one dimension without losing any tertiary information during the transition. A Minor motif is then searched in this one dimensional string using exhaustive search technique with linear time complexity. The efficiency is demonstrated by the comparison of AMinorSearcher with benchmark tool FR3D. FR3D lacked in both precision and recall while AMinorSearcher did not.

**AMINORMOTIFFINDER – A GRAPH GRAMMAR BASED TOOL TO  
EFFECTIVELY SEARCH A MINOR MOTIFS IN 3D RNA MOLECULES**

**by  
Ankur Malhotra**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Bioinformatics**

**Department of Computer Science**

**January 2011**

Blank Page

## **APPROVAL PAGE**

### **AMINORMOTIFFINDER – A GRAPH GRAMMAR BASED TOOL TO EFFECTIVELY SEARCH A MINOR MOTIFS IN 3D RNA MOLECULES**

**Ankur Malhotra**

---

Dr. Jason T.L. Wang, Thesis Advisor Professor of Bioinformatics and Computer Science, NJIT	Date
---	------

---

Dr. Michael A. Baltrush, Committee Member Associate Professor of Computer Science, NJIT	Date
--	------

---

Dr. Guiling Wang, Committee Member Assistant Professor of Computer Science, NJIT	Date
---	------

## **BIOGRAPHICAL SKETCH**

**Author:** Ankur Malhotra  
**Degree:** Master of Science  
**Date:** January 2011

### **Undergraduate and Graduate Education:**

- Master of Science in Bioinformatics  
New Jersey Institute of Technology, Newark, NJ, 2011
- Bachelor of Technology in Bioinformatics  
Vellore Institute of Technology University, Vellore, India, 2007

**Major:** Bioinformatics

### **Presentations and Publications:**

Malhotra Ankur, Using FTIR Imaging to do Automated Protein Sequencing, Indian Patent Journal number 31/2007.

Sarkar R, Malhotra A, Phylogenetic analysis of *Oryza L* species based on seed protein markers, Advanced Biotech vol. 9, Issue 1, pp 10-12.

**Dedicated to  
Professor Jason Wang,  
My family and  
Ms Tripti Kingra (very soon Mrs Tripti Malhotra😊)**



## **ACKNOWLEDGEMENT**

I would like to thank Professor Jason Wang, my thesis advisor who has been a guiding light throughout and without whose support this thesis would have never been possible. I would also like to thank Dr. Michael A Baltrush and Dr. Guiling Wang for being in my thesis committee. My sincere appreciation goes to my fellow student James Slocum for coding this algorithm in Java and making the idea into a real-time tool. Last but not the least I would like to thank my family for their immense support throughout.

## TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION.....	1
1.1 What is RNA Motif.....	1
1.2 What is A Minor Motif .....	2
1.3 Other Characteristics of A Minor Motif.....	3
1.4 The Four Versions of A Minor Motif.....	3
2 MATERIALS AND METHODS.....	6
2.1 Data.....	6
2.2 Graph Grammar.....	7
2.3 Overview.....	7
2.4 Encoding .....	8
2.5 Forming 1D String.....	12
2.6 Searching A Minor Motifs.....	15
2.7 The XML Converter.....	16
3 EXPERIMENTS.....	25
3.1 FR3D.....	25
3.2 Experiments Performed With FR3D.....	26
4 DISCUSSION.....	31
5 REFERENCES.....	35

## LIST OF TABLES

Table	Page
1.5 Various types of A Minor interactions.....	4
2.4 L and W Non Watson and Crick notation.....	11
3.2.1 Summary of experiment on 1FFK as candidate structure and 1NJP and 2J00 as target structure .....	29
3.2.2 Summary of experiment on 1VQ0 as candidate structure and 1JJ2 and 1FFK as target structure .....	30
4.1 New encoding scheme to mine all A Minor Motifs.....	33

## LIST OF FIGURES

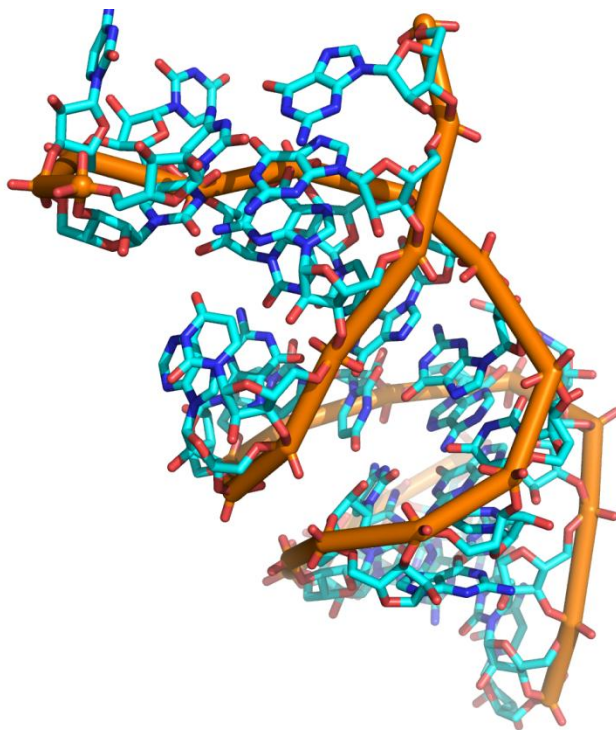
Figure	Page
1.1 Pymol snapshot of bases of one helix crossing with the other stem forming a motif as seen in 2ZJB .....	1
2.1 Algorithm in a nutshell.....	6
2.3 Flowchart describing the overall algorithm.....	8
2.4 Example to show various encoding protocols used in our algorithm.....	10
2.5.1 Encoding a stem region.....	12
2.5.2 Encoding a hairpin loop region.....	13
2.5.3 Flowchart to show the rules of encoding in case of stem or loop regions.....	14
2.5.4 The direction of flow of Type1 and Type2 string transformations .....	15
2.6 Flowchart to explain the mining of A Minor motifs using exhaustive search.....	16
2.7 Graph grammar describes a simple A-minor motif found in 1CX0.....	24
3.1 FR3D Interface .....	26
3.2.1 Stick model of C:1214, G:1048, C:1209 of 2J00 examined in PyMol.....	27
3.2.2 Stick model of G:1947, A:1920, C:1917 of 1NJP examined in Pymol.....	28
3.2.3 True A Minor Motif discovered by AMinorMotiFinder at G: 988, C: 997, A:1012 not discovered by FR3D .....	28
3.2.4 1JJ2 positions mined by FR3D G: 1489, C: 1456 and A: 1659 shows no A Minor interactions. Drawing tool: Pymol .....	29
3.2.5 No A Minor Interactions discovered in 1FFK C: 763, G: 901 and A: 643, Drawing Tool: Pymol .....	30
4.1 Integration with MC Fold .....	34

# CHAPTER 1

## INTRODUCTION

### 1.1 What is RNA Motif

RNA Motifs are chain like three dimensional biological molecules and are regarded as combination of tertiary structures. A Particular type of motif will have a similar spacial elements in x,y and z axis. These motifs are often formed at a particular temperature and pH in order to give increased stability to mostly unstable RNA structure. Motifs usually occur across the base pairs in the same/different stem/ loop and can be of various types based on the types of interactions they exhibit.

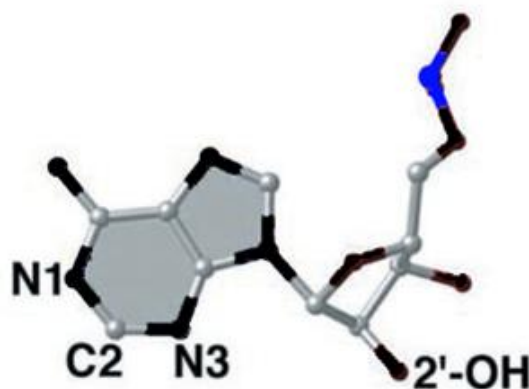


**Figure 1.1** Pymol snapshot of bases of one helix crossing with the other stem forming a motif as seen in 2ZJB.

## 1.2 What is A Minor Motif?

A minor motif (12) is the most common RNA Motif found in RNA's. It involves insertion of smooth minor groove edges of Adenine into minor groove helices (usually at a WC GC Pair) where they form Hydrogen bond with one or both OH's of those pairs resulting in more stabilization of contact between those helices. The examination of A residues of 50s rRNA indicated that adenine is by far the most ubiquitous and conserved nucleotide in the RNA. The theory was verified when 23s rRNA was sequenced and was found that A residues were the most abundant residues not involving helices and many of these non base A's are conserved. (17) It was later justified that it is these conserved A residues that form these motifs via N1-C2-N3 edges. (Figure 1.2).

RNA residues are also conserved (just like in proteins) because they are critical for function or they stabilize the tertiary structure of RNA. A residue is most abundant, conserved and most which are involved in tertiary interactions. It had been realized earlier, when the first 23S rRNAs were sequenced, that A residues are more abundant than other bases in 23S rRNA sequences not involved in regular helix formation and that many of these nonbase-paired As are conserved. (17). These conserved A's form A Minor interactions.



**Figure 1.2** The smooth minor groove face of A packs into the minor groove of helix. Its N1,N3, and 2'-OH atoms usually forms H interactions. Drawn using Chems sketch.

### 1.3 Other Characteristics of A Minor Motif

- a.) It usually stabilizes RNA-RNA interactions, loop helix, loop - loop interactions, GAAA tetra loop and helix junctions where there is a sharp change in the direction of the backbone (12, 13)
- b.) The third base interaction with WC pairs in a minor groove (12, 14)
- d.) The most common A Minor interaction is A-GC (12)
- e.) The donor adenosine may be present on single strand or on variety of non canonical pairings (12,16)




### 1.4 The Four Versions of A Minor Motif

In A Minor motif, the ribose phosphate of one strand is closer to the other. Four versions of the motif can be identified that differ with respect to the position of the O2' and N3 atoms of the A residue relative to the O2' atoms of the base pair in the receptor helix.(12)

In Type 1 A Minor motif, The O2' and N3 atoms of the A residue are inside the minor



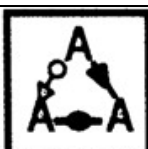

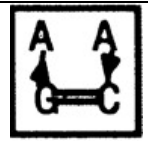
groove of the receptor helix. The inserted base for the Type I interaction must be an adenine. In type2 A Minor Motif, The O2' of the A residue is outside the near strand O2' of the helix and the N3 of the A residue is inside the minor groove. The inserted base for the Type II interaction must be an adenine. Type3 A Minor Motif, The O2' and N3 of the A (or other) residue are outside the near strand O2' of the receptor helix. A rare Type 0 A minor motif, The N3 of the A (or other) residue is outside the O2' of the far strand of the receptor helix. Type 0 and Type III are neither A specific nor A selective. Type 0 is not base ribose of the inserted residue fills the minor groove of the receptor helix, not the base, and because the Watson–Crick faces of all bases include groups that can hydrogen-bond to 2'OH groups and same with type III interaction of A Minor. In contrast, Type 1 and II are way more specific to adenine residues. Only A's can form this kind of interactions and they are highly biased to C-G type of bonding. (Table 1.5)

**Table 1.5** Various types of A Minor interactions

Type of A Minor	Snapshot of the graph structure	PDB example
Type 1		1FFK
Type 2		IFFK
Type 3		IFFK



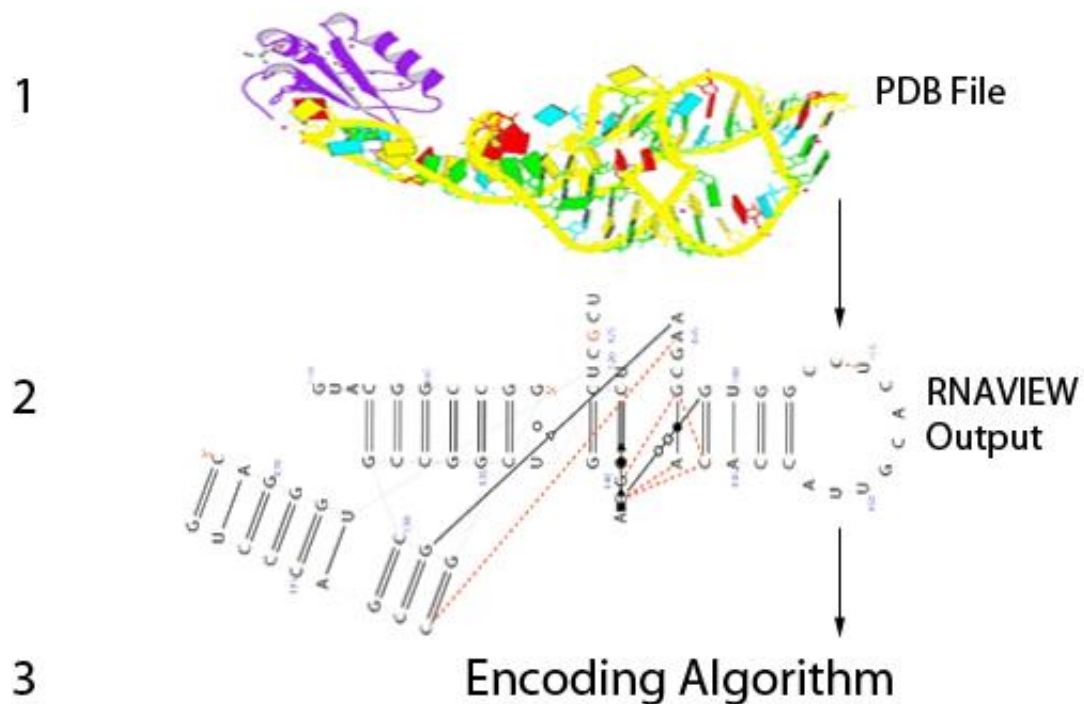
**Table 1.5 (continued)** Various types of A Minor interactions

Type of A Minor	Snapshot of the graph structure	PDB example
Type 2		1FFK
Type 3		1VQ0
Type 3		2J00
Type 3		2G1S
Type 0		2GCV

## MATERIALS AND METHODS

## 2.1 Data

Data was collected from Protein Data Bank (PDB). (10) Annotation program, RNAVIEW (11) was used to produce the corresponding RNA Graphs. All interactions were encoded with our unique encoding scheme. (Figure 2.1) These interactions include the phosphodiester (backbone) link, the canonical WC pairing GC and AU and the 12 non canonical WC base pairs defined by the LW nomenclature (18).



**Figure 2.1** Algorithm in a nutshell.

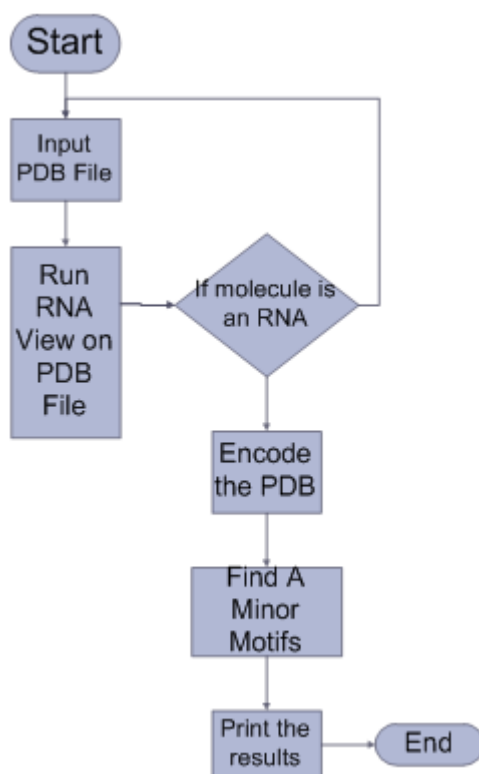
## 2.2 Graph Grammar

The encoding scheme was based on the study called "graph grammar". (The word "graph grammar" would be used as our encoding technique from here forth). (19) Graph grammar or graph rewriting is the technique of writing a new graph out of an existing graph  $p : L \rightarrow R$  using an automatic program (here our algorithm). It uses a set of rules (described below) from L(Left) to form a pattern graph R (Right). R is also called a replacement graph. The rule is applied to L by mining the instances on R.

## 2.3 Overview

The algorithm takes one or more RNA as input. RNA VIEW (21) engine is executed on the RNA PDB data. The output of the RNA VIEW file is an XML file (based on a language called RNAML) (20), a Post Script file and a text output file. XML file was preferred over others as it contains the most structured and complete information of the three dimensional molecule. The algorithm consists of three stages (Figure 2.3)

- a.) Encode the RNAVIEW output into a one dimensional string
- b.) Do an exhaustive search to mine A Minor motifs
- c.) Report the positions of the motif mined.



**Figure 2.3** Flowchart describing the overall algorithm.

## 2.4 Encoding

The following steps are implemented Every B1-B2 edge is encoded using the following notation.

**$Q(x1,x2)=B1$ ; 2D coordinate (B1);Pos(B1); B2; 2D coordinate (B2);Pos(B2); A; B;  
C; D; E**

Where

**B1** is the first base

**B2** is the second base

**2D coordinate (Bx)** is the 2D coordinate space in which base is located

**Pos (Bx)** is the base number of Bx in 2D space

**A=1** for single dash lines (tertiary interactions the pair has a single hydrogen bond or the pair has more than one hydrogen bond but with bad geometry) (Figure 2.4 (a))

**A=0** for non single dash lines (Figure 2.4 (a))

**B=0** for non modified (upper case nucleotides) (Figure 2.4 (b))

**B=1** for modified lower case nucleotides (Figure 2.4 (b))

(The value of B is determined by the target nucleotide. So if we are going from 5' to 3' and B1(5') is connected to B2(3'), and if B2 is lower case then B=1 and vice versa) (Figure 2.4 (b))

**C=1** for parallel lines (or GC pair) (Figure 2.4 (c))

**C=2** for single line (AU/AT pair) (Figure 2.4 (c))

**C=3** for a circle in between B1 and B2 (GU wobble pair) (Figure 2.4 (c))

**C=0** for none of above (Figure 2.4 (c))

**D=0** for a base pair (Figure 2.4 (d))

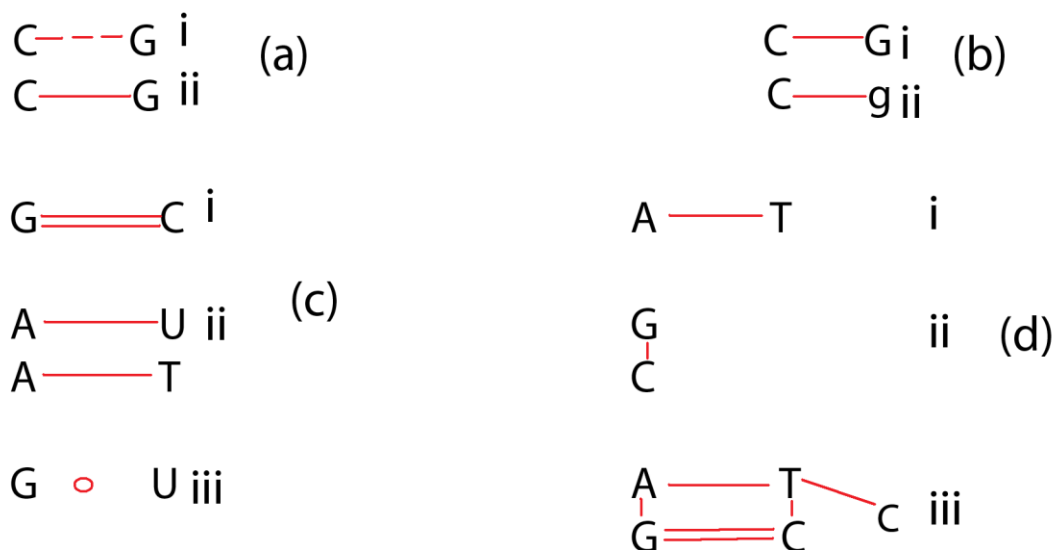
**D=1** for a base stack (Figure 2.4 (d))

**D=2** for a diagonal (Figure 2.4 (d))

**D=3** for a loop (for example tRNA loop) (Figure 2.4 (d))

**E=0** for Watson and Crick bases













**E=1 -12** for the bases according to the Table 2.4



(a) i and ii are the examples of  $A=1$  and  $A=0$  respectively  
 (b) i and ii are the examples of  $B=0$  and  $B=1$  respectively. Notice that in (b) ii, the target nucleotide is lowercase "g".  
 (c) i, ii and iii are the examples of  $C=1$ ,  $C=2$  and  $C=3$  respectively. The GU wobble pair (c) iii is denoted by a circle in between G and U  
 (d) i, ii and iii are examples of  $D=0$ ,  $D=1$  and  $D=2$  respectively. In (d) i, A and T form a base stack (horizontal), in (d) ii, G and C forms base stack (vertical) and in d(iii) T and C forms a diagonal interaction

**Figure 2.4** Example to show various encoding protocols used in our algorithm.

**Table 2.4** L and W Non Watson and Crick Notation (18)

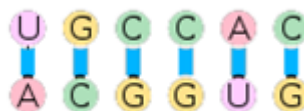
No	Glycoside Orientation	Bond	Interacting Edges	Symbol	Default local standard orientation
1	Cis		WC/WC		Anti Parallel
2	Trans		WC/WC		Parallel
3	Cis		WC/Hoogsteen		Parallel
4	Trans		WC/Hoogsteen		Anti Parallel
5	Cis		WC/Sugar Edge		Parallel
6	Trans		WC/Sugar Edge		Anti Parallel
7	Cis		Hoogsteen/Hoogsteen		Anti Parallel
8	Trans		Hoogsteen/Hoogsteen		Parallel
9	Cis		Hoogsteen/Sugar Edge		Anti Parallel
10	Trans		Hoogsteen /Sugar Edge		Parallel
11	Cis		Sugar Edge / Sugar Edge		Anti Parallel
12	Trans		Sugar Edge / Sugar Edge		Anti Parallel

## 2.5 Forming a 1D string

As RNA is a long molecule, the following rules apply in determining the order by which the base pairs would be encoded. The mining algorithms, start with 5' and ends at 3' region. Rules (described below) are different in case of stems and loops (Figure 2.5.3)

**Stem Region:** (Type 1) (22) A Stem region (Figure 2.5.1) occurs when two regions of the same strand, usually complementary in nucleotide sequence when read in opposite directions, base-pair to form a double helix that ends in an unpaired loop. The resulting lollipop-shaped structure is a key building block of many RNA secondary structures. The following rules apply while encoding the stem region.

- a.) Start from 5' region
- b.) Encode the upper base pair first encountered from the 5' region.
- c.) Encode the two base stacks just below the base pair
- d.) Encode the lower base pair
- e.) Look for the diagonal bases (if any) present in all the regions of this stem. If present, encode the diagonal base pair; otherwise mark it as "N"

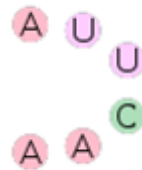


**Figure 2.5.1** Encoding a stem region.



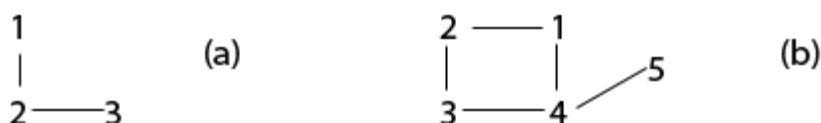
**Hairpin Loop:** (Type 2) (22) A hairpin loop is an unpaired loop of messenger RNA (mRNA) that is created when a RNA strand folds and forms base pairs with another section of the same strand. The resulting structure looks like a loop or a U-shape. The following rules apply when encoding a hairpin loop

- a.) Start from 5' region
- b.) Encode the upper base pair first encountered from the 5' region to the very next base pair.
- c.) Check for any diagonal base pairs connected to upper or lower base pairs
- d.) If present encode the diagonal base pairs, if not write N.



**Figure 2.5.2** Encoding a Hairpin Loop region.





**Type 1 Strig transformation** Type 2 String Transformation  
 In (a), 1 and 2 are encoded using "graph grammar" notation. Then it is checked whether there is another base following 2. If yes, then 2 and 3 is encoded using "graph grammar" notation. The flow continues till there are no more bases following this notation.

The output of (a) would be

Q(1-2);Q(2-3)

In (b) 1 and 2 are encoded using "graph grammar" notation. Then it is checked whether there is another base needed to complete the box. If yes then 2 and 3 are encoded. The condition is rechecked.

Once the box is completed, diagonal base pairs are checked. If present the "graph grammar" string is produced between the diagonal base and the base present in the stem. If not "N" is written instead of the string. So the output of (b) would be Q(1-2);Q(2-3);Q(3-4);N;N;N;Q(4-5).

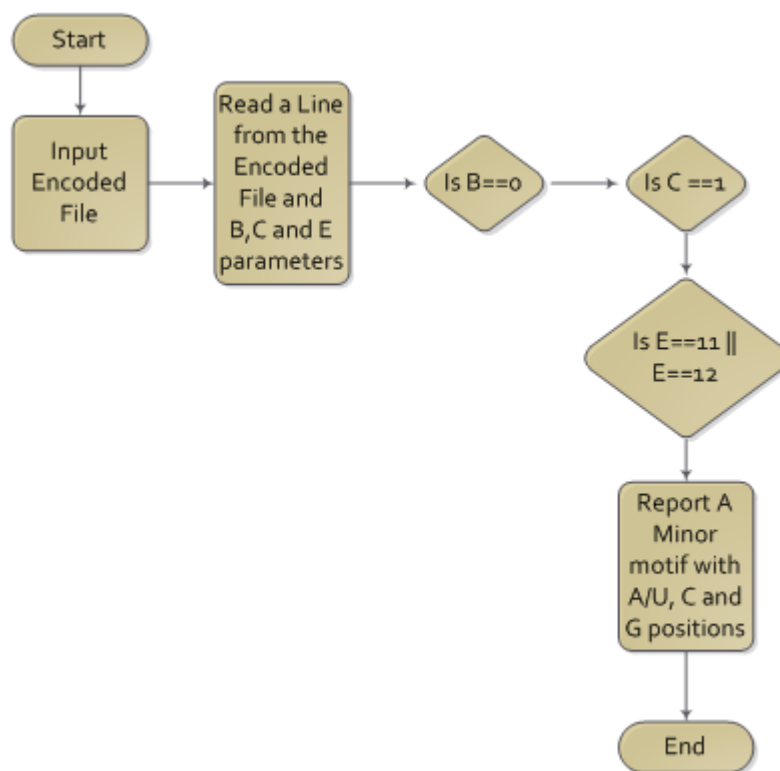
where Q(A-B) means to "graph grammar" string between A and B)

**Figure 2.5.4** The directional flow of Type 1 and Type 2 String transformation.

## 2.6 Searching A Minor Motifs

The A-Minor searcher uses the file in the special graph grammar format to deterministically search out A-Minor motifs. Because of the robust, and easy to read nature of the encoding scheme, the search is made trivial. By simply searching the encoded file for a line where B is equal to 0, and C is equal to 1. Then looking at the next line to see if B is equal to 0 and E is equal to 11 or 12. We can then report this location as

an A minor motif. This encoding format lends itself to be searched in linear time.(Figure 2.6)



**Figure 2.6** Flowchart to explain the mining of A Minor Motif using exhaustive search technique.

## 2.7 The XML Converter

The XML file produced by RNAView contains a lot of data about the RNA molecule that needs to be evaluated. The first and most important task is to gather all of the relevant information from the document and build the molecule with special data structures. The first data structure that was created was called a BaseNode. The BaseNode type contains all of the information necessary to build a single nucleotide. The data structure records the physical information of the base, such as the base type,

index, and whether it is modified. It also stores whether it belongs to a single strand or loop, and the previous and next base nodes in the RNA strand. It also records positional data such as the X and Y coordinates for rendering the graphics. Each node acts as a link in a linked list that can be traversed back and forth as necessary.

The base node also keeps an array of another special data type called Connections. Connections hold a 5' and 3' BaseNode and the details of how they are connected to each other. These connections are very important for mining motifs, and provide valuable information. The possible connection types are

- a) GU Wobble pairs
- b) Tertiary interactions (the pair has a single hydrogen bond or the pair has more than one hydrogen bond but with bad geometry)
- c) Standard Watson-Crick base pairs
- d) Watson-Crick / Watson-Crick Interacting edges
- e) Watson-Crick / Hoogsteen Interacting edges
- f) Watson-Crick / Sugar Interacting edges
- g) Hoogsteen / Hoogsteen Interacting edges
- h) Hoogsteen / Sugar Interacting edges
- i) Sugar / Sugar Interacting edges

The Glycosidic bond orientation is also recorded within the data structure. This data structure also acts as a linked structure to build the RNA strand into a graph in memory. Using the XML file from RNAView, The XMLConverter first reads in the necessary

data by looking for the important data tags. Below are the tags that contain the necessary data, along with descriptions of each of the tags.

<rnaml>

<molecule>

<sequence>

<seq-data>

<structure>

<model>

<base>

<position>

<base-type>

<str-annotation>

<base-pair>

<base-id-5p>

<base-id>

<position>

<base-id-3p>

<base-id>

<position>

<edge-5p>

<edge-3p>

<bond-orientation>

<single-strand>

<segment>

<base-id-5p>

```

    <base-id>
      <position>
    <base-id-3p>
      <base-id>
        <position>

```

The tags are shown at their proper levels, and were used to quickly build up the data structure described above. Below is a description of each tag. If two tags follow each other then that means one means little without the other. (20)

**<rnaml>** This is the root tag that begins an RNAmL XML document. This document can be verified with rnaml.dtd.

**<molecule>** This tag begins the description of a single RNA molecule. A single XML file can have more than one molecule, each beginning with the <molecule> tag.

**<sequence> <seq-data>** This contains a quick preview of the entire RNA sequence.

**<structure> <model>** This begins the description of the physical structure of the RNA sequence. It starts with each base one by one, then follows all of the connection data, followed by special single strand data.

**<base>** This begins the description of a single base element.

**<position>** This is the index used inside the XML document to reference this base.

**<base-type>** this is the actual base type A, C, G, or U. If it is modified it will be listed in lower case.

**<str-annotation>** This begins the description of the connections between the bases.

**<base-pair>** This begins a description of a connection between two base elements.

**<base-id-5p>** This is the 5' base element in the pair

**<base-id> <position>** This is the index of the base being referenced. This is the same as the position shown above and is used to match up two bases to create a pair.

**<base-id-3p>** This is the 3' base element in the pair

**<base-id> <position>** This is the index of the base being referenced. This is the same as the position shown above and is used to match up two bases to create a pair.

**<edge-5p>** This is the edge type for the 5' side. This can be W, H, S, +, or -

**<edge-3p>** This is the edge type for the 3' side. This can be W, H, S, +, or -

**<bond-orientation>** This is the Glycosidic Bond Orientation, it can be c, or t.

**<single-strand> <segment>** This begins the description of a range of bases indexes that make up a single strand segment.

Using a simple XML parser, this document is easy to go through and build up the RNA strand graph. Once the RNA graph is built, the next step is to convert it into the graph grammar form. This graph grammar format is special because it describes the two dimensional graph in a one dimensional text encoding. Once this is done, the entire RNA can be re-built from this file, or can be searched for special criteria that would be difficult if dealing with only the XML file, or a visual representation of the RNA and encoded as the following string.

*base-id1; index1; xCoordinate, yCoordinate; base-id2; index2; xCoordinate,  
yCoordinate; A; B; C; D; E; m*



This format allows for easy parsing and splitting of the important data. All data is separated by a single “;” (semi-colon) character. The format also has single lines with the letter N on them. These lines also have a special meaning. It means that there are no diagonal connections belonging to any of the surrounding base elements. Below is a breakdown of the graph grammar, and a description of each element.

**base-id1** This is the base type of the 5' base. It can be A, U, C, G, a, u, c, or g

**index1** This is the index of this base as it appears in the original PDB file. This is obtained from the XML document.

**xCoordinate, yCoordinate** These are the coordinates used for rendering the graphics. These positions are obtained from the XML document.

**base-id1** This is the base type of the 3' base. It can be A, U, C, G, a, u, c, or g

**index1** This is the index of this base as it appears in the original PDB file. This is obtained from the XML document.

**A** This can have the values of 0, or 1. It is 1 if and only if there is a tertiary edge between these bases

**B** This can have the values of 0, or 1. It is 1 if and only if one of the bases is modified (shown as lower case letter)

**C** This can have the values of 0, through 3. It is 0 if and only if the base pair is non-Watson and Crick. It is 1 if and only if the base pair is a Watson and Crick G-C edge. It is 2 if and only if the base pair is a Watson and Crick A-U edge. It is 3 if and only if the base pair is a G-U wobble pair.

**D** This can have the values of 0, through 3. It is 0 if and only if the two bases make a base pair. It is 1 if and only if the two bases make a base stack. It is 2 if and only if the two bases make a diagonal connection (not base pair or base stack). It is 3 if and only if the bases are part of a single strand loop.

**E** This can have the values of 0, through 12. It is 0 if and only if the bases make do not make up a diagonal connection. Otherwise, 1 – 12 are shown in the table 2.4

Below is an example of an encoded piece of an RNA graph using the graph grammar defined above, and a visual representation of it

C;118;8.13,162.21;G;129;56.79,181.71;0;0;1;0;0;m

N

G;129;56.79,181.71;A;166;225.42,377.32;0;0;0;2;12;m

C;119;0.00,182.49;A;165;212.32,377.32;1;0;0;2;0;m

N

G;129;56.79,181.71;C;130;64.91,161.43;0;0;0;1;0;m

C;118;8.13,162.21;G;129;56.79,181.71;0;0;1;0;0;m

C;118;8.13,162.21;G;117;16.25,141.94;0;0;0;1;0;m

G;117;16.25,141.94;C;130;64.91,161.43;0;0;1;0;0;m

G;129;56.79,181.71;A;166;225.42,377.32;0;0;0;2;12;m

N

N

N

The pseudocode of the process is as follows.

**Algorithm** scanForAMinorMotif(motifs: List):

**motifs:** This is a list that will be populated with the positions of the A-Minor motifs as they are found in the graph grammar document

```
motifs.clear()
```

```
int numberOfMotifsFound = 0
```

```
int i ← 0
```

```
for each Line l in graphGrammar file:
```

```
    i ← i + 1
```

```
    if l.equals("N") then:
```

```
        continue
```

```
String[] tokens ← l.split(";") Split the line based on ";" character
```

```
int b ← parseInt(tokens[7]) The 7th and 8th tokens now have each
```

```
int c ← parseInt(tokens[8]) important piece of data for A-minor
```

```
if (b = 0 AND c = 1) then:    the base is not modified, and the connection  
is a C-G edge
```

```
l ← nextLineFromFile()
```

```
if (l.equals("N")) then:
```

```
    continue
```

```
tokens = l.split(";")
```

```

    b = parseInt(tokens[7])

    int e = parseInt(tokens[10])

    if (b = 0 AND e = 11 OR e = 12) then:      the base is not
modified, and the connection is a sugar-sugar edge

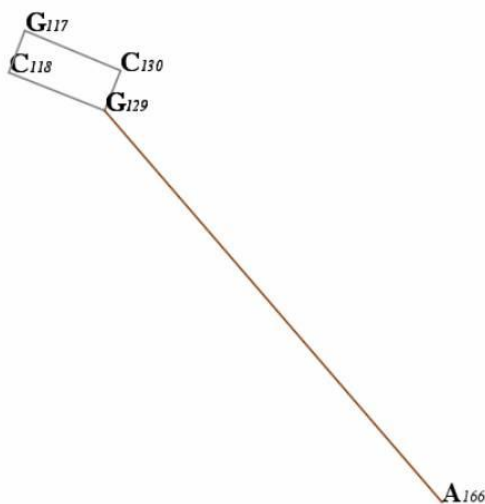
        motifs.add(i)

        numberOfMotifsFound++

return numberOfMotifsFound

```

This graph grammar describes a simple A-minor motif found in 1CX0. (Figure 2.7)



**Figure 2.7** Graph grammar describes a simple A-minor motif found in 1CX0

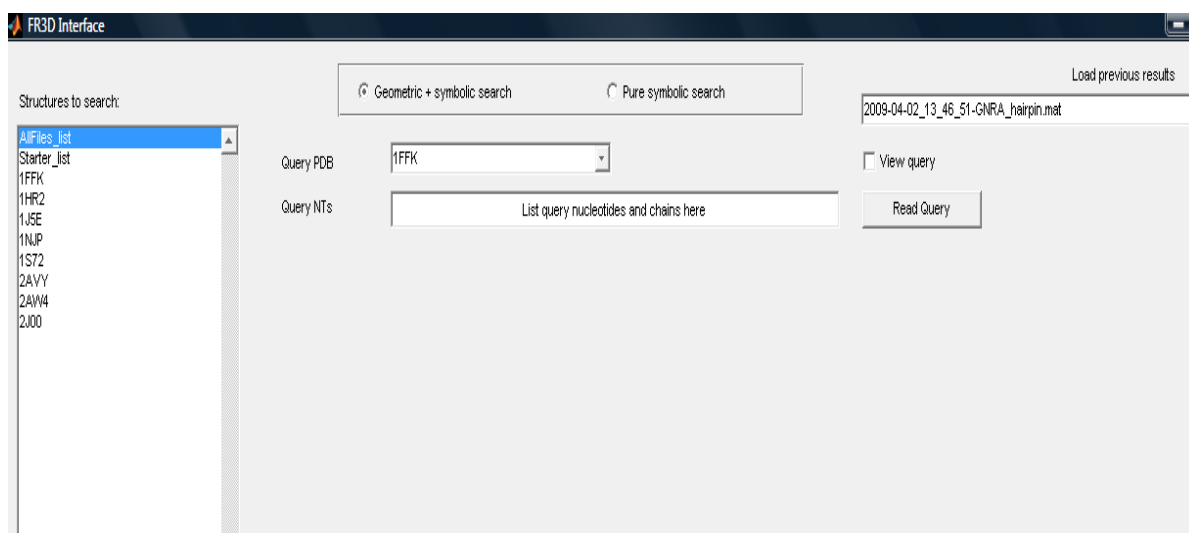
## **CHAPTER 3**

### **EXPERIMENTS**

#### **3.1 FR3D**

FR3D (commonly pronounced as “Fred”) (Figure 3.1) (23) is a Matlab based tool to find small RNA motifs (upto 20 nucleotides) in a PDB file. Through geometric and symbolic searches a user specifies a PDB file with the known motif as a query PDB. The user also specifies the position at which the query nucleotides are located (For example in case of A Minor, the user would specify the position where A/U,G,C nucleotides are located). The user then tells FR3D to read this crystal structure. After the crystal structure is read, the software interprets whether there are more than one chains present in the molecule (for example 1s72 has 6 chains). If present, the user can specify the chain he is concentrating on. The software also displays an interaction matrix in which the user specifies more constraints like Armstrong distances between each nucleotides of the motif. The user then specifies a parameter called guaranteed cutoff. The search algorithm is guaranteed to find all candidates whose geometric discrepancy with the Query motif is less than this number. The discrepancy is roughly comparable to RMS discrepancy. The user must specify the Relaxed Cutoff discrepancy, using the text-box labeled Relaxed Cutoff. The algorithm is not guaranteed to find all candidates whose discrepancy from the Query motif is between the guaranteed cutoff and the relaxed cutoff. User then specifies the target PDB files in which he wants to search for pattern matching his query. This approach can be effective in mining 3D patterns like motifs. It however has two drawbacks.

- a.) A Similar 3D structure may or may not mean a similar motif
- b.) Since the program is program is threshold dependent, it can exhibit false positives / false negatives



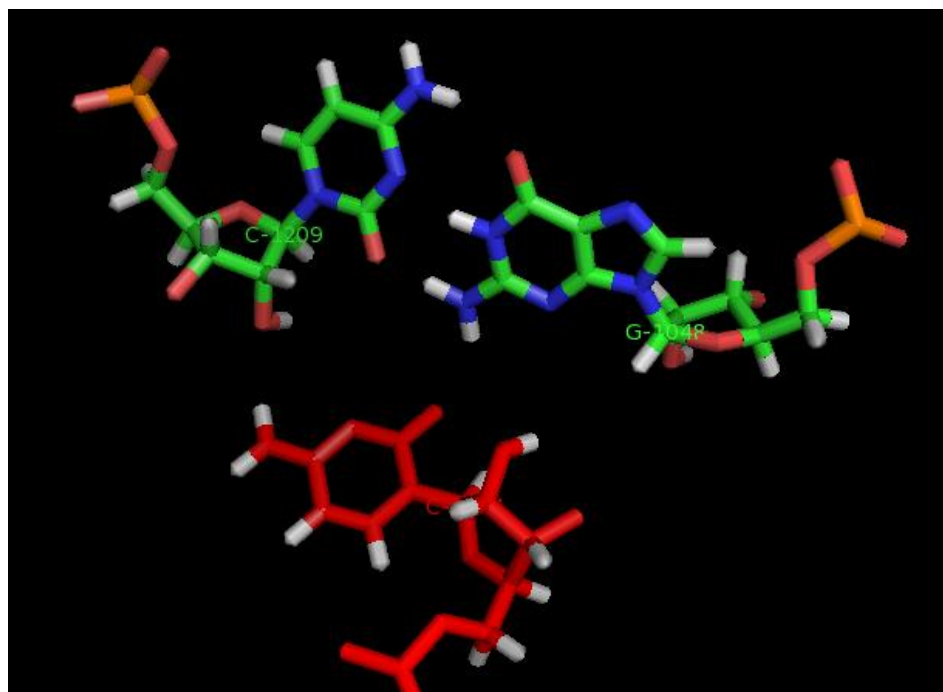
**Figure 3.1** FR3D interface.

### 3.2 Experiments Performed with FR3D

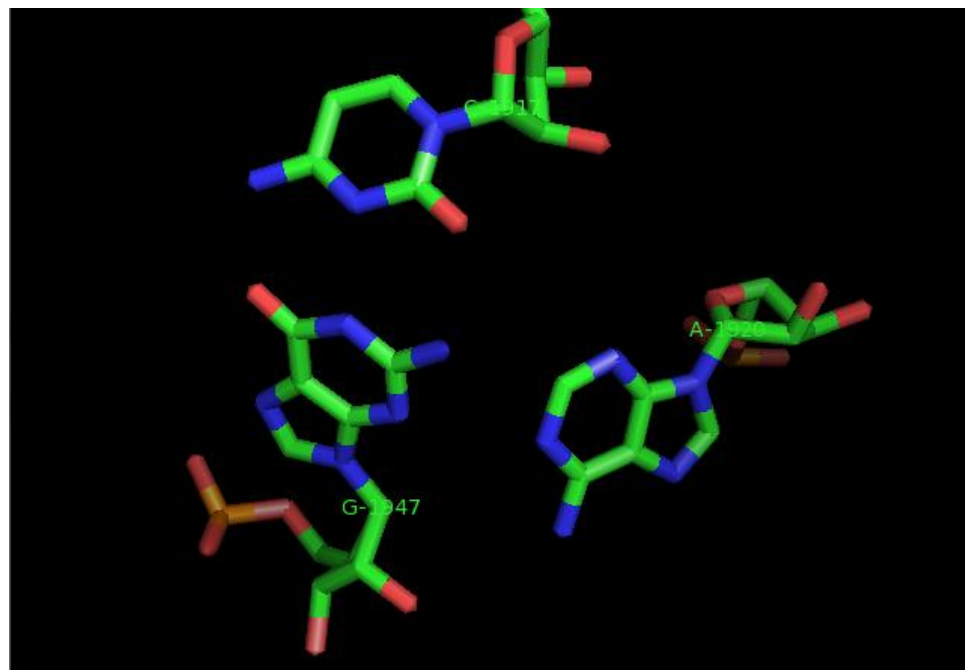
Two independent cases were investigated through FR3D to mine A Minor Motifs. Supplementary for Yurong Xin et al (24) was taken as a reference for candidate structures for A Minor Type 1 Motifs.

In first case, 1FFK was taken as a candidate structure and 0:A521 0:G1364 0:C637 were taken as candidate positions. A Minor motifs were then searched (with default parameters, guaranteed cutoff =0.5 and relaxed cutoff =0.5) in 1NJP and 2J00. The search resulted in 111 motif structures; several were false positives. FR3D also missed false negatives. When 2J00 C:1214, G:1048, C:1209 was examined closely in

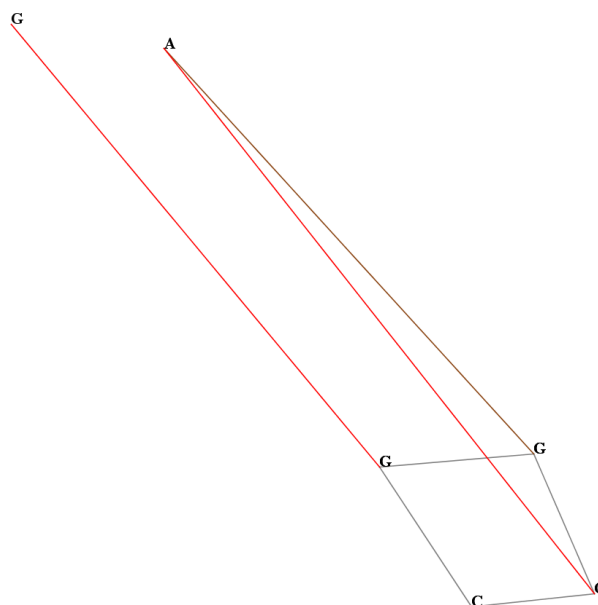
PyMol (molecule viewing tool) (25) , no signs of A minor interactions were found. There was no hydrogen bonding between these three bases. This clearly demonstrates a false positive predicted by FR3D. (Figure 3.2.1). The overall results are summarized in Table 3.2.



**Figure 3.2.1** Stick model of C:1214, G:1048, C:1209 of 2J00 examined in PyMol.



**Figure 3.2.2** Stick model of G:1947, A:1920, C:1917 of 1NJP examined in Pymol.



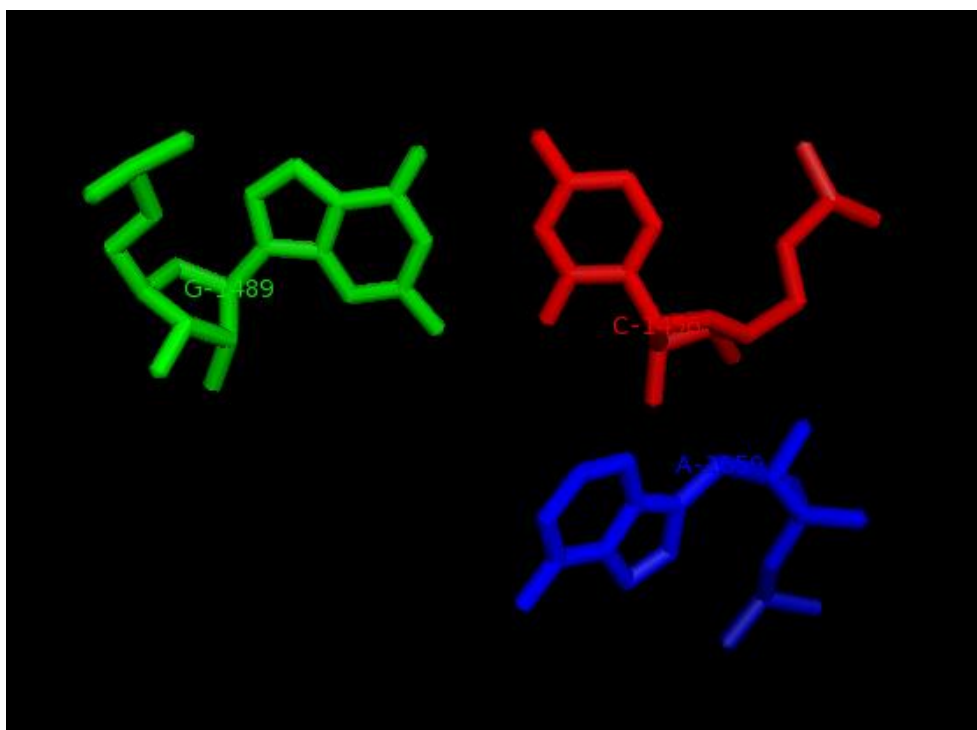
**Figure 3.2.3** True A Minor Motif discovered by AMinorMotiFinder at G: 988, C: 997, A: 1012 not discovered by FR3D.



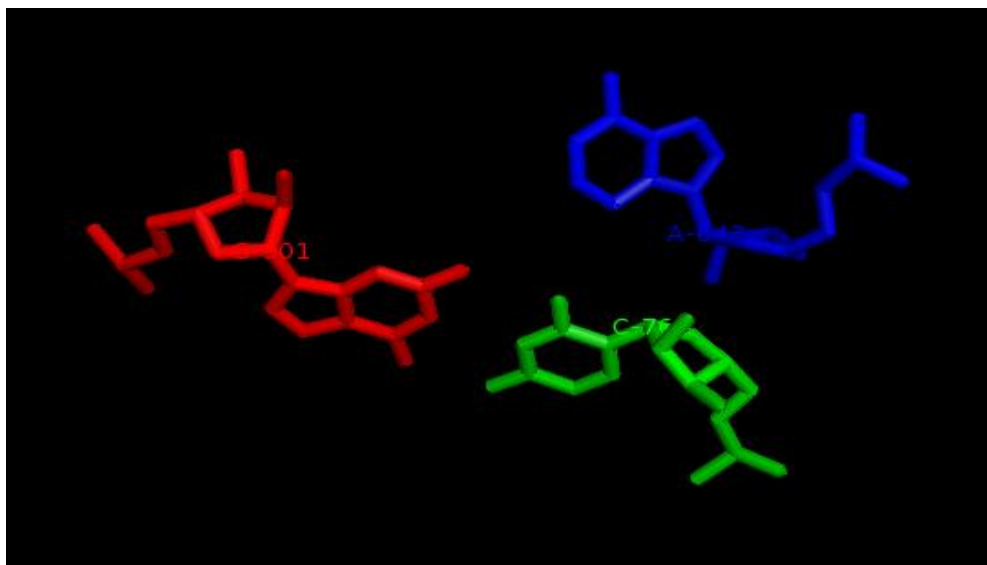
**Table 3.2.1** Summary of experiment on 1FFK as candidate structure and 1NJP and 2J00 as target structure

Molecules Mined through FR3D	Number of False Positives found in FR3D	Number of false Negatives	Number of false positives Found from A Minor Searcher	Number of false negatives found from A Minor searcher
111	43	5	0	14

In another experiment 1VQ0 A:1458, U:862, A:784 was taken as a candidate structure and matched with 1FFK and 1JJ2 as target. Both False positives and false gatives were seen in the experiment. When closely examined in PyMol, false positives were found. (Figure 3.2.4, Figure 3.2.5)



**Figure 3.2.4** 1JJ2 positions mined by FR3D G: 1489, C: 1456 and A: 1659 shows no A Minor interactions. Drawing tool: Pymol.



**Figure 3.2.5** No A Minor Interactions discovered in 1FFK C: 763, G: 901 and A: 643, Drawing Tool: Pymol.

**Table 3.2.2** Summary of experiment on 1VQ0 as candidate structure and 1JJ2 and 1FFK as target structure

Molecules Mined through FR3D	Number of False Positives found in FR3D	Number of false negatives from FR3D	Number of False Positives from A Minor Searcher	Number of false negatives from A minor searcher
154	35	6	0	10

## CHAPTER 4






### DISCUSSION

It is clearly shown from the above experiments that A Minor Searcher is a deterministic tool with zero false positives. The higher number of false negatives is because A Minor Searcher is currently not built to mine Type 0 and Type 3 Motifs (12, 15). The study also shows the powerful nature of graph grammar technique. This technique is not limited to A Minor and can be used to mine other motifs like Sarcin Ricin, Pseudoknots, K Turn, and C Loop etc as every motif has a unique signature which can be encoded and mined through graph grammar techniques. Since “graph grammar” encodes a three dimensional structure into a one dimensional string, this string can be easily applied to algorithms like “Suffix Trees” and techniques like Support Vector Machines and Neural Networks for data mining and pattern finding. Patterns can be found on the entire PDB RNA molecules through suffix trees and novel motifs can be discovered. Graph matching algorithms can also be applied to this graph grammar technique. This graph matching algorithm can detect motifs with far better accuracy than exhaustive searching. One such algorithm (26) is planned to be applied to the existing graph grammar in order to search for the false negatives missed by the exhaustive search technique. It is also planned to encode other A Minor variations and build a complete A Minor Searcher program (Table 4.1). A web server is also planned to be constructed for this tool. The web server would allow a user to upload / specify a PDB file. The user would then specify which motifs he or she wants to find in the PDB file through checkboxes. RNAVIEW engine (11) would run in the

Blank Page

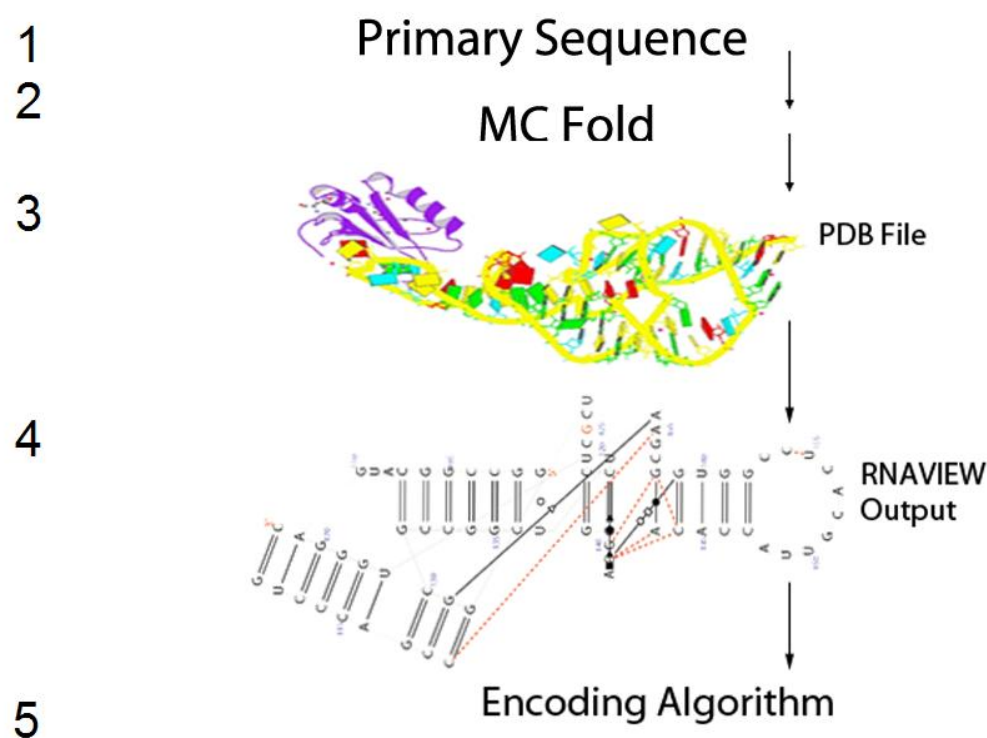
background and convert the PDB into a XML file. This XML file would be parsed to the “graph grammar” technique. Graph searching, suffix trees and exhaustive search would be applied to this graph grammar to mine motifs accurately. The results would be displayed on the web browser as PNG files.

**Table 4.1** New encoding schemes to mine all A Minor Motifs

Type of A Minor	Snapshot of the graph structure	PDB example	Encoding Scheme
Type 3		IFFK	A-A and A-U and C=2
Type 3		1VQ0	C=1, U-C and E=11/12
Type 3		2J00	A-A and E=10 or E=11/12 or E=1
Type 3		2G1S	C=1 and E=11/12
Type 0		2GCV	A-G/C E=11 or E=12

Recently Marc Parisien and François Major published a very powerful tool called MC fold (27) which can convert a primary RNA sequence into a tertiary RNA structure.

The discovery of this tool adds a new dimension to our research. Since this tool converts a primary sequence to a tertiary structure and our input is a tertiary structure, our research is no longer limited to a tertiary structure present in PDB. Upon integration of this tool in the first step (before a RNA VIEW graph is made), the user no longer would have to specify a PDB file as an input. A primary sequence would be enough to find motifs / patterns across the RNA sequence. This is the next step of research by our group. (Figure 4.1)



**Figure 4.1** Integration with MC Fold

## REFERENCES

1. Hendrix,D., Brenner,S. and Holbrook,S. (2005); RNA structural motifs: building blocks of a modular biomolecule. *Q. Rev.Biophys.*, 38, 221–243.
2. Alesker,V., Nussinov,R. and Wolfson,H. (1996) Detection of non-topological motifs in protein structures. *Protein Eng.*, 9,1103–1119.
3. Duarte CM, Wadley LM, Pyle AM. (2003); RNA structure comparison, motif search and discovery using a reduced representation of RNA conformational space. *Nucleic Acids Res.*;31:4755-4761.
4. Wadley LM, Pyle AM (2004);. The identification of novel RNA structural motifs using COMPADRES: an automated approach to structural discovery. *Nucleic Acids Res.*; 32:6650-6659.
5. Eddy S (2001). Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.* 2:919-929.
6. Storz G (2002). An expanding universe of noncoding RNAs. *Science*; 296:1260-1263.
7. Leontis NB, Lescoute A, Westhof E. (2006) The building blocks and motifs of RNA architecture. *Curr. Opin. Struct. Biol.* ;16:279-287.
8. Moore P. (1999) Structural motifs in RNA. *Annu. Rev. Biochem.* ; 68:287-300.
9. Djelloul M, Denise A. (2008) Automated motif extraction and classification in RNA tertiary structures. *RNA*;14:2489-2497.
- 10 Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. (2000) The Protein Data Bank. *Nucleic Acids Res.* ;28:235-242.
11. Yang H, Jossinet F, Leontis N, Chen L, Westbrook J, Berman H, Westhof E. (2003) Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.*;31:3450-3460.
12. Nissen,P., J.A.Ippolito, N.Ban, P.B.Moore, and T.A.Steitz. (2001). RNA tertiary interactions in the large ribosomal subunit: The A-minor motif. *PNAS USA* 98:4899-4903.
13. Carl C. Cornell and Karren Swinger; (2003) Common and distinctive features of GNRA tetraloops based on a GUAA tetraloop structure at 1.4 Å resolutions *RNA*. March; 9(3): 355–363.
14. Harry F. Noller; (2005) RNA structure: reading the ribosome. *Science*. September 2; 309(5740): 1508–1514

18. N. B. Leontis and E Westhof Geometric nomenclature and classification of RNA base pairs. (2001); RNA. April; 7(4): 499–512.
19. Graph rewriting Wikipedia [http://en.wikipedia.org/wiki/Graph\\_rewriting](http://en.wikipedia.org/wiki/Graph_rewriting) retrieved on December 6th 2010
20. Allison Waugh, Patrick Gendron, Russ Altman, James W Brown, David Case, Daniel Gautheret, Stephen C Harvey, Neocles Leontis, John Westbrook, Eric Westhof, Michael Zuker, and François Major (2002 ); RNAML: a standard syntax for exchanging RNA information. RNA. June; 8(6): 707–717.
21. Yang, H., Jossinet, F., Leontis, N., Chen, L., Westbrook, J., Berman, H.M., Westhof, E. (2003). Tools for the automatic identification and classification of RNA base pairs. Nucleic Acids Research 31.13: 3450-3460
22. Stem Loop RNA Regions Wikipedia <http://en.wikipedia.org/wiki/Stem-loop> retrieved on December 6th 2010
23. Michael Sarver, Craig L. Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B. Leontis (2008); FR3D: finding local and composite recurrent structural motifs in RNA 3D structures J Math Biol
24. Yurong Xin, Christian Laing, Neocles B. Leontis Annotation of tertiary interactions in RNA (2008); Structures reveals variations and correlations, RNA 14: 2465-2477
25. The PyMOL Molecular Graphics System, Version 1.2r3pre, Schrödinger, LLC.
26. András Frank (2004). On Kuhn's Hungarian Method - A tribute from Hungary. Technical Report Egerváry Research Group
27. Marc Parisien, François Major, (2008) The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data, Nature 452, 51-55