

STtructure Analysis of **RNA**

TUTORIAL for **STAR**

by

F.H.D. van Batenburg
A.P.Gulyaev
C.W.A.Pleij

© Copyright 1993 F.H.D.van Batenburg. This manual may be cited at will and parts of it may be copied for non-commercial purposes after a previous permission in writing by the copyright-holder.

Group Bio-informatika of the Institute of Theoretical Biology
Theoretical Biology of Leiden University
Kaiserstraat 63, 2311GP Leiden,
The Netherlands

e-mail: Batenburg@Rulsfb.LeidenUniv.NL

<http://wwwbio.LeidenUniv.nl/~Batenburg/STAR.html>

Acknowledgements

STAR is developed by:

©dr.F.H.D. van Batenburg ¹⁾

(1989) J.P. Abrahams (1990) M. van den Berg

(1990) V.W.A. Bos dr. S. Gulyaev ^{1 2)}

(1999) J. Ng dr. C.W.A. Pleij ²⁾

(2001) M.J. Schepers (1991) J.J.M. Riethoven

(1993) J. Th. Voskuilen (1992) M. Weynen

We appreciate it if you refer to *STAR* and the theories behind it (greedy folding ³⁾, stochastic folding ⁴⁾ genetic algorithm folding ⁵⁾) whenever you use *STAR* predictions.

Every effort has been made to ensure the accuracy of this document, but no responsibility can be accepted for any errors or omissions. *STAR* is licensed on an as-is basis; no warranties or liabilities as to any incidental or consequential damages, loss of revenues, reputations, grants or data apply.

We think that more than one copy of *STAR* is not unreasonable as long as only one copy of *STAR* is actively used at a time. Moreover, it may be used on more than one machine for educational purposes only and if the machines are at a single location (classroom). Reselling or redistribution of *STAR*, or its parts, in another program system is forbidden. We have no intention of enforcing our copyrights by legal procedures, but appeal instead to your honesty to comply with these rules.

This software was developed using APL.68000, a proprietary product of MicroAPL Ltd, which has given permission for a runtime version of APL.68000 to be included with the software.

The PC version is implemented in APL2000/W, owned by Cognos which has given permission for a runtime version to be included with the software.

Typing: Mw.Y.Zitman

Manual version 1.0 (1994), version 2.0 (1995), version 3 (2001), version 5 (2003).

Program version 3.0 (March 1994); version (4.0 March 1995); version 5 (dec.2002).

¹ Institute of Evolutionary and Ecological Sciences, Department of Theoretical Biology, University of Leiden, the Netherlands.

² Department of Biochemistry, Gorlaeus Laboratories, University of Leiden, the Netherlands. Department of Biochemistry, Gorlaeus Laboratories, University of Leiden, the Netherlands.

³ Abrahams,J.P. & Berg,M.v.d. & Batenburg,F.H.D. & Pleij,C.(1990): Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. Nucleic Acids Res. 18(10)3035-3044.

⁴ Gulyaev,A.P. (1991): The computer simulation of RNA folding involving pseudoknot formation. Nucleic Acids Research Vol.19(9)2489-2494.

⁵ Batenburg,F.H.D.van & Gulyaev,A.P. & Pleij,C.W.A.(1995): An APL-programmed Genetic Algorithm for the prediction of RNA secondary structure. J.Theor.Biol. (174)269-280.

TABLE OF CONTENTS

| | | |
|------------|--|-----------|
| 1. | HOW TO USE THIS TUTORIAL | 6 |
| 1.1 | WHY THIS TUTORIAL? | 6 |
| 1.2 | UPDATE OR NEW VERSION? | 6 |
| 1.3 | HOW TO USE THIS TUTORIAL? | 6 |
| 1.4 | NOTATION..... | 7 |
| 1.5 | MAKING THE EXERCISES MORE INTERESTING. | 7 |
| 2. | SIMPLE COMPUTATION OF SECONDARY STRUCTURE USING GREEDY FOLDING..... | 8 |
| 2.1 | GET A SEQUENCE INTO <i>STAR</i> | 8 |
| 2.2 | GET A PREDICTION. | 8 |
| 2.3 | VIEW PREDICTION | 9 |
| 2.4 | SAVE YOUR WORK | 9 |
| 3. | QUICK TOUR AROUND VARIOUS OUTPUT VIEWS. | 10 |
| 3.1 | TABLE OUTPUT | 10 |
| 3.2 | SPAN-VIEW | 10 |
| 3.3 | LEGO- AND MOUNTAIN-VIEW | 11 |
| 3.4 | BRACKET-VIEW..... | 12 |
| 3.5 | FINALLY | 12 |
| 4. | STAR OUTPUT TO GRAPHIC PROGRAMS AND TEXT-PROCESSORS..... | 14 |
| 4.1 | <i>STAR</i> OUTPUT ON PAPER | 14 |
| 4.2 | <i>STAR</i> OUTPUT INTO A TEXT-PROCESSOR | 14 |
| 4.3 | <i>STAR</i> OUTPUT TO OTHER PROGRAMS | 14 |
| 5. | ENTERING SEQUENCES BY HAND OR FROM DATABASES..... | 16 |
| 5.1 | <i>STAR</i> EDITOR | 16 |
| 5.2 | LOAD ASCII-SEQUENCE | 16 |
| 5.3 | SEQUENCE DATABASE..... | 17 |
| 5.4 | OTHER DATABASES..... | 17 |
| 5.5 | FINALLY | 17 |
| 6. | FORCING AND PROHIBITING BASE-PAIRING..... | 18 |
| 6.1 | EXCLUDING NUCLEOTIDES | 18 |
| 6.2 | FORCING STEMS | 18 |
| 7. | COMPUTATION OF SECONDARY STRUCTURE USING STOCHASTIC FOLDING..... | 20 |
| 7.1 | STRAIGHT STOCHASTIC FOLDING..... | 20 |
| 7.2 | STOCHASTIC FOLDING OF GROWING RNA | 20 |
| 8. | GENETIC ALGORITHM..... | 22 |
| 8.1 | STRAIGHT G.A. | 22 |
| 8.2 | WHAT <i>STAR</i> REPORTS | 22 |
| 8.3 | FOLDING PROCESS..... | 23 |
| 9. | COMPARING DIFFERENT STRUCTURES (ALGORITHMS). | 24 |
| 9.1 | THREE ALGORITHMS | 24 |
| 9.2 | PARAMETER SETTINGS | 24 |
| 9.2.1 | <i>greedy algorithm</i> | 24 |
| 9.2.2 | <i>stochastic algorithm</i> | 24 |
| 9.2.3 | <i>Genetic algorithm</i> | 25 |
| 9.3 | GOOD, BETTER, BEST? | 25 |
| 10. | TIPS AND TRICKS..... | 26 |
| 10.1 | QUICK INVESTIGATION OF MAIN STEMS | 26 |

| | | |
|------|-----------------------------------|----|
| 10.2 | QUICK PSEUDOKNOT RECOGNITION..... | 26 |
| 10.3 | QUICK ENERGY CALCULATION | 26 |
| 10.4 | PRINTER/SCREEN PROBLEMS..... | 26 |
| 10.5 | CIRCULAR VERSUS LINEAR RNA..... | 27 |
| 10.6 | TEMPERATURE..... | 27 |
| 10.7 | SEVERAL SIMULATIONS..... | 27 |

1. HOW TO USE THIS TUTORIAL

This chapter tells you the purpose of this tutorial. It presents three alternatives to use this booklet and explains the used notation.

1.1 Why this tutorial?

This tutorial introduces you to *STAR*.

First of all you have to install the program. This is described in a separate leaflet.

After installation you can jump straight into using the program without reading anything. This is what we ourselves often do and we did our best to design the software as intuitive as possible to make this tutorial completely useless.

If, however, you like to be introduced to a program step by step, this tutorial will be your helpful guide. Here we gradually introduce everyday problems that you may want to solve with *STAR*. This booklet treats many of the options of *STAR*, although not all of them and certainly not in a systematic order, but in an order that depends on the need during particular research activities.

Another aim of this tutorial is to show you some of the less obvious options, like forcing stems. So if you have used *STAR* already with the "jump-straight-in-the-water" approach, you might consider later to spend some of your leisure time to skip this tutorial over and to look for some particular chapters that might be interesting to learn something new.

1.2 Update or new version?

Experienced *STAR* users will probably find a substantial part of this manual rather boring. The main change that all user will find in this tutorial is that the menu structure changed completely and that energy values are not editable anymore, but are recomputed automatically when you change the temperature.

For old friends that are familiar with version 3, the new exciting thing is the Genetic Algorithm. This makes chapters 9 and 10 worthwhile. In that case you can skip all other chapters.

For "older" users chapter 8 covering the stochastic algorithm is worthwhile too. Some new topics are discussed in paragraphs 3.4, 4.3 and 6.3.

1.3 How to use this tutorial?



How to proceed?

Because each chapter is (relatively!) independent, it is possible to jump to some especially interesting topic and discard previous chapters. However, this works only to a certain extent. Just try it if you like to learn in this way. If this doesn't work, then you apparently need a more gradual approach.

A more comfortable way is the gradual approach of following the order of the tutorial. First read chapter 2, then 3 and so on. This ensures that new things are introduced more gradually.

If you don't want to spend the time to follow each chapter faithfully (but on the other hand feel a bit unsure to plunge right in the middle) there is a third alternative.

Just skim each chapter (one by one)

superficially. If it seems all very obvious, it suffices if you just try to find the mentioned menu-options (but do not execute them) and check if each chapter summary agrees with your understanding. If so, you can jump to the next chapter quickly.

1.4 Notation

Although we tried to avoid computerese, we used a shorthand notation throughout this tutorial that may need an explanation.

To indicate menu-titles we enclose the text in square brackets like [File].

To indicate an option of that menu-title, we use the "/" as a separator like in [File/Open]. If we omit the menu-title, we still keep the slash to indicate that we mean an option and not the title, like in [/Open].

We also use square brackets to indicate buttons. For example as in: choose alternative [iteration] and press [Cancel]. For "ease of speak" we use shorthand "press [OK]" instead of "position the mouse pointer above the box-frame titled [OK] and press your left button on the mouse". We also use this shorthand for clicking on options in menus.

Finally we use a shorthand for a sequence of clicks. For example, instead of saying:

- press menu-title [Calculate]
- choose option [/Parameters]
- click in checkbox [Growth]
- press [OK]

we often say

- press
[Calculate/Parameters] [Growth] [OK]

1.5 Making the exercises more interesting.

This manual intends to drag you quickly over the main features of *STAR*. If you are very eager to analyse your "own" RNA, the exercises in this manual are far more interesting if you use that RNA. Remember however, that the

exercises here use small (sometimes artificial) sequences because these are quickly analysed. As computer time roughly increases with a square of sequence length, a long RNA may require several hours for computation. This can make the exercises quite boring. So for exercise purposes we advise to use (parts of) RNA's that don't exceed a 100 nucleotides.

Summary of this chapter:

Documentation

- separate leaflet for installation
- tutorial to learn *STAR*

Old users

- most recent friends will notice the new menu-structure
- recent old users: stochastic and genetic algorithm in chapters 7-9.
- very old friends: GA in chapter 8.

Study advises; either....

- jump to interesting topics.
- skim sequentially, study summaries.
- read sequential faithfully.

Exercise:

- more interesting with your own sequences.
- keep length small (<100) for exercise purposes.

2. SIMPLE COMPUTATION OF SECONDARY STRUCTURE USING GREEDY FOLDING.

This chapter will take you on a quick tour through the most essential steps of the program. You will learn how to read a small sequence of nucleotides from disk into the program and how to get a prediction of secondary structure.

In this chapter we shall predict the structure of a tRNA. This prediction proceeds along the following steps:

1. get the sequence into program *STAR*.
2. compute a secondary structure.
3. look at the results.

2.1 Get a sequence into *STAR*

For its prediction, *STAR* needs a sequence to work with. If necessary, you can type the characters that describe the nucleotides of a sequence yourself, but we have included a couple of sequence files in this package. So we will read one of those files.

- start *STAR*.
- press [File/Import/Primary].
- choose file TMVRNA.1.

Now *STAR* detects some lower case characters and asks you if you want them converted to upper case. This is no capriciousness of *STAR*, nor a bug, but a feature. Lower-case characters will prohibit pairing (this is useful if you have experimental evidence for single strands). So, if you want to exclude a part of the sequence from pairing, you formulate that part in lower-case characters. As you can always change your mind later, our choice is not very important, so...

- press [Return] (which equals [OK]).

On screen you see a sequence like:

```
Source: "C:\TMVRNA.1"; Date: 2001 2 8
      5      10      15      20      25      30...
AUAAU AAAUA ACGGA UUGUG UCCGU AACAG...
GCAUA GUGUU UUUCC CUCCA CUUAA AUCGG...
CGGGU CAAAU GUAUA UGGUU CAUAA ACAUC...
AGGGG UUCGA AUCCC CCCGU UACCC CCGGU...
Fig.1: Sequence of 3' end of TMU RNA.
```

Notice also that the screen-title now says at the right: "...TMVRNA.1/_". This means that *STAR* read and accepted this file for use. It also helps you to remember what RNA is active.

Unfortunately we made a mistake (on purpose) because we didn't want to work with this RNA. Correcting this mistake is no problem at all, just read in the other file:

- Press [File/Import/Primary] again.

STAR warns you that the sequence that you just read will be replaced. This is what you want to, so...

- press [OK] and choose file TRNATHRT.1.

STAR reads this sequence. You see this new sequence displayed on screen. Moreover *STAR* replaced "TMVRNA.1/_ " by "TRNATHRT.1/_ " in the window-title.

2.2 Get a prediction.

For computation of the secondary structure we use (evidently) menu [Calculate]. All structure procedures are handled by this menu-item. What are those procedures?

- press [Help/Calculate]

You see a short description of the options. In particular notice that option [/Parameters...] will set parameters, and that [/Compute...] and [/BatchCompute...] will compute a secondary structure.

For now, ignore option [/Parameters...] and ask for a straight prediction of the secondary structure:

- close the Help box.
- press [Calculate/Compute/Greedy]

STAR starts the computation of prediction using the greedy algorithm ⁶.

You see the various steps that *STAR* takes in a report that looks somewhat like: ⁷

```

:::::
----- Iteration:  4
Best stems:   10 13 22 25 -6.4
               10 12 18 20 -2.2
               15 17 23 25 -1.6
               24 25 46 47 -0.9
Chosen:10 13 22 25 paired, ΔG=-6.4kcal/mol.
----- Iteration:  5
:::
Fig.2: Report of prediction progress.

```

Although you don't need to understand these intermediate messages, it informs you that *STAR* is computing and not asleep. If you are interested, observe how *STAR* selects stems, step by step.

At the end of the computation your mouse-arrow appears again and *STAR* displays the final structure. You also see in the window-title "C:\TRNATHRT.1/2" where "/2" means that *STAR* has computed a secondary structure.

⁶ Abrahams, J.P. & Berg, M.v.d. & Batenburg, F.H.D.v. & Pleij, C. (1990): Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. Nucleic Acids Res. 18(10)3035-3044.

⁷ As we are continuously improving *STAR*, changes after writing this manual may change this result. Don't think that you did something wrong if your prediction differs slightly from this tutorial.

2.3 View prediction

Next, let us have a quick look at that structure (in the next chapter we will look at it more elaborately).

First look at the original sequence again:

- press [View/Primary].

You see the sequence. Now the structure:

- press [View/Secondary/...].
- You can choose from several output alternatives, but let us take the simplest: [/Table].

You see the result in Fig.3.

Can you identify these stems in the original sequence?

```

27 32 38 43 -11.4 4.0 -7.4 * CACCCU
                                * GUGGGA
49 53 61 65  -9.8 4.5 -5.3 * GGCAG
                                * CCGUC
10 13 22 25  -7.4 3.6 -3.8 * GCUC
                                * CGAG
 1  7 66 72 -11.2 2.7 -8.5 * GCUGAUA
                                * CGACUAU

```

Fig.3: Final report of TRNATHRT structure

2.4 Save your work

Finally, let us save our work to resume tomorrow:

- press [File/Save]. *STAR* will save everything(parameters, sequence, structure) in file as "TRNATHRT.star".
- press [File/Quit].

Summary of this chapter.

Prediction using greedy folding by...

1. read sequence [File/Import/Primary]
2. calculate secondary structure by [Calculate/Compute/Greedy]
3. look at results by:
 - sequence: [View/Primary]
 - structure: [View/Secondary/Table]
4. store your work:
 - [File/Save]

3.3 Lego- and mountain-view

Mountain-view is a type of output published by Hogeweg and Hesper⁸. Like span-view, it can help you to discover patterns quickly.

Unfortunately the output is very bulky (mountainous indeed), therefore we designed a more compact form called lego-view.

- press [View/Secondary/Lego].

You see patterns like in Fig.4. Notice that in lego-view non-paired nucleotides run horizontally and paired nucleotides run vertically. The nucleotide pairs are placed opposite to each other.

```

CCGAGCCUUGGUAAA
G-----|-----C
A-----|-----U
G-----|-----C
A-----|-----U
U-----|-----AGAGGUCCCCGGCAGUCCAAUC
                    a:Hairpin

CCUUGGUAAA
G----|---C
A----|---U
G----|---C
C----|---G
C----|---GGAGGU
G-----|-----C
A-----|-----U
G-----|-----C
A-----|-----U
U-----|-----AGCAGUCCAAUCUGCCUAAUUU
                    b:Hairpin with bulge loop
      GUAAA
      G-|-C
      U-|-A
      U-|-A
      C-|-G
CCGAGC-|-GGAGGU
G-----|-----C
A-----|-----U
G-----|-----C
A-----|-----U
U-----|-----AGCAGUCCAAUCUGCCUAAUUU
                    c:Hairpin with internal loop

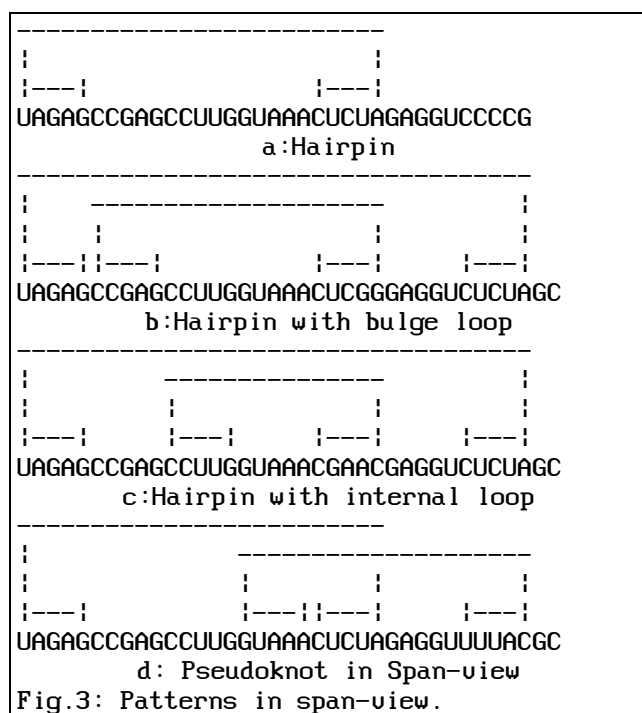
```

Fig.4: Patterns in Lego-view.

In lego-view you can easily recognise patterns. See Fig.4. Look how easily hairpin, bulge and internal loops can be recognised. **Beware that**

lego-view and mountain-view cannot display pseudoknots.

⁸ Hogeweg, P. & Hesper, B. (1984): Energy directed folding of RNA sequences. Nucleic Acids Res. (12)67-74.



3.4 Bracket-view

Bracket-view is a quick way to see if a structure contains any pseudoknots. In this view opening and closing parentheses indicates the base pairing. However, for a pseudoknot, one of the two stems is indicated by square brackets instead of parentheses. Look at Fig.5 to see the difference between a hairpin with a bulge loop and a pseudoknot.

```
$ 1 UAGAGCCGAGCCUUGGGCUAAGGGCGGGUCCCCGGC
% 1 -----((((((-----)))-----)))-----
```

```
$ 50 UAGAGCCGAGCCUUGGCGGAAGGGGCUUGGUCCCCGGC
% 50 -----(((([[[-----]])-----]])-----
```

Fig.5: Bulge loop (top) versus pseudoknot (bottom), both in bracket-view.

- press [View/Secondary/Bracket].

Look at the result. You see that TRNATHRT does not contain pseudoknots.

3.5 Finally

Ask *STAR* to display the structure of TRNATHRT in lego-view and try to recognise the patterns of a hairpin, bulge and internal loop.

4. STAR OUTPUT TO GRAPHIC PROGRAMS AND TEXT-PROCESSORS.

After prediction of a structure you might want to enter the predicted structure in a text processor. Or you want to massage the output in another software package.

Here we will exercise with both alternatives. We will show you how to get the output of *STAR* on paper and how to get it into a text processor. After that we will show how to get the output of *STAR* in another package like for example LoopDLoop or RNADraw.

First, if you haven't done so before, 1) start *STAR*, 2) get sequence and structure by [File/Open] for file "TRNATHRT.star".

4.1 *STAR* output on paper

To get your *sequence* on paper is easy:

- choose [File/Print/Primary]

STAR will print your sequence ⁹⁾.

To print a *structure* is very similar.

- choose [File/Print/Secondary/...]
- choose what view you want to see, for example [/Span].

STAR will print the structure.

4.2 *STAR* output into a text-processor

You can easily insert the output of *STAR* in a publication. Because the output can be rather unwieldy sometimes, we chose to store to file, rather than to the clipboard.

Assume that you want to get the lego-view.

- choose [File/Export/Secondary/Lego].
- A file-dialogue-box appears that asks for a file name. Type "TRNATHRT.2lg" and press [OK].

Now *STAR* writes the lego-view of your secondary structure to that file. You can view it in a text processor like Word, but this will look awful. The reason is that the picture only

makes sense when displayed in a so-called non-proportional font. Let us do this now; start your text processor and...

- import file TRNATHRT.2lg (if necessary tell the text processor this is a text-file).
- select the whole text by mouse.
- choose a non-proportional font like Courier.

Now you have the *STAR* picture in your text processor.

4.3 *STAR* output to other programs

STAR is developed primarily for prediction of secondary structures. Therefore, the output-views are limited to typewriter-like pictures. For more beautiful line drawings other software is more suitable.

A well known drawing package for Mac is LoopDloop ¹⁰. This package can draw beautiful line drawings of RNA structures. For Windows you could make a drawing using RNADraw ¹¹.

Be aware that currently most drawing packages (including RNADraw and LoopDloop) cannot handle pseudoknots. If your structure has pseudoknots you must first remove them.

STAR can export the RNA information (both primary and secondary) for RNADraw and

⁹ Occasionally *STAR* finds erroneously that a printer in a network is busy and reports so. If your network has this problem, you should store the output to file [File/Export/...] and print that file using your favourite text processor. See paragraph 4.2.

¹⁰ Author Don Gilbert, Biocomputing office biology dept. of Indiana University, Bloomington, IN47405, USA. Available on FTP: ftp.bio.indiana.edu as /molbio/loopdloop/loopdloop.hqx.

¹¹ Authors: Ole Matzura & Anders Wennborg (<http://rnadraw.base8.se/>)

LoopDloop. These files bracket-type and ConnecT (also used by MFOLD of Zuker ¹²).

The procedure to make such a file is:

- check first if there are pseudoknots (for a quick check, use [View/Secondary/Bracket] and watch out for [[[: :]]]-brackets). If so, remove one of the stem-pairs of each pseudoknot using [Edit/Secondary].
- export the structure using [File/Export/Secondary/Connect].
- in the file-selector box that appears give a file name; for example TRNATHRT.2cr.

Now import the file into LoopDloop:

- start LoopDloop.
- choose [File/Open] and choose file TRNATHRT.2br.

The result is displayed in Fig.1. If you are not satisfied with this drawing, LoopDloop has several means to improve that picture, but we have not done so for Fig.1.

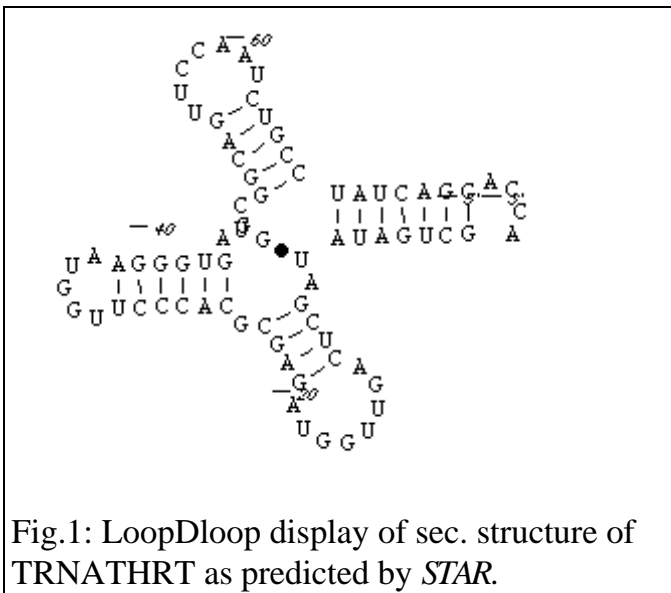


Fig.1: LoopDloop display of sec. structure of TRNATHRT as predicted by *STAR*.

For Windows you could make a drawing using RNADraw. The recipe is similar:

- start the program RNADraw.

- choose [File/Import/Mfold connect file (.ct,.con)].
- select the ConnecT-file TRNATHRT.2cr.
- select [OK] in the sequence title window.
- use the right mouse button on the text *Secondary Structure 0 kcal 37C*.
- choose [New/2DRadial Drawing] in the displayed menu.
- use the right mouse button on the text *2dradial Drawing*.
- choose [View] in the displayed menu.

You will see the following drawing:

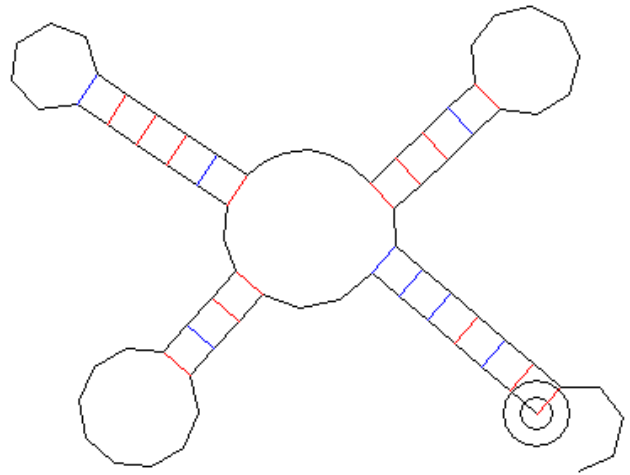


Fig.2: 2D radial drawing by RNADraw.

Summary of this chapter. You can export *STAR* data in several ways using [File/Print...]:

- sequence on paper by [File/Print/Primary].
- structure on paper: [Print/Secondary/...] followed by choice of view-type.
- export for publication: [File/Export/Secondary/...], then choose view-type; read file in text processor and change font to a non-proportional font such as Courier.
- export for drawing: first remove one of each pseudoknot stem-pair, next activate [File/Export/Secondary/Connect] and save the file. Read this file in a graphing program such as LoopDloop or RNADraw.

¹² Zuker,M (1989): On finding all suboptimal foldings of an RNA molecule. Science (244)48-52: see also <http://www.ibc.wustl.edu/~zucker/index.html>

5. ENTERING SEQUENCES BY HAND OR FROM DATABASES

This chapter will show how to enter a primary string of nucleotides in various ways.

STAR needs a sequence of nucleotides to work on. There are several ways to enter that sequence.

1. You can type in all the characters of the nucleotides using the editor of *STAR*.
2. You can type in the sequence using your favourite editor and then ask *STAR* to read it.
3. You can get a sequence from EMBL database.
4. You can get a sequence from any other source and remove those parts that are offensive to *STAR*.

We will exercise each of these four alternatives.

But if you haven't done so before, first start *STAR*.

5.1 *STAR* editor

To use the *STAR* editor, do as follows:

- press [File/New]

An empty window appears. If you continued from previous chapters you are warned that a sequence exists; click [OK] to continue. Type the sentence:

THIS IS NOT A RNA SEQUENCE

- press [File/Exit&Save].

You may feel happy to observe that *STAR* does not accept this rubbish. A dialogue-box appears which reports illegal characters and you are asked if you want to correct this automatically.

- press [Manual].

Now you have to correct it yourself. You see the sentence again, but all offensive characters that are not proper nucleotides are marked underneath. You see:

THIS IS NOT A RNA SEQUENCE
*^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^

You could now remove all offensive characters (the lowest line with carets is harmless because it is preceded by a "*") so that you get: **AAUC**. It is quicker to let *STAR* do this:

- press [File/Exit&Save].
- you get the warning again, press [Automatic].

You see that *STAR* converted T to U and finally accepts: **UUAUUC**.

5.2 Load ASCII-sequence

For longer sequences you probably have the nucleotides already somewhere in a text-file¹³. Then you don't need to type those nucleotides again. Here is how you feed *STAR* such a sequence.

- press [File/Import/Primary].

STAR warns you that this will overwrite your current sequence, but this is OK:

- press [OK] and choose file STAR.1 that is supplied by *STAR*.

On screen you see the sequence that *STAR* extracted from the original file. The original file was a bit different, however (see Fig.1).

To see that original:

- press [Edit/Primary].

Notice the asterisks. *STAR* ignores all text in a line after an asterisk. You can use this to comment (parts of) your sequence if you want.

To leave the editor:

- press [File/Exit&Save] or [File/Quit].

¹³ Text-files are files that contain only "normal" characters. That is, no layout characters. Wordprocessors can make such files, but you have to instruct them explicitly that you want to save without layout as "txt" type files.

Notice also that *STAR* accepted your file: in the window-title you see "STAR.1" that acknowledges the acceptance of this file. The suffix "/2" is removed, because no computed structure is left.

mentioned before, this is not a bug but a feature that allows you to keep some parts of a sequence single-stranded. You can change this whenever you want in option [Calculate/Parameters].

```
AUAAUAAAUACGGAUUGUGUCC * This file is provided with STAR
GUAAUCACACGUGGUGCGUACGA * as a sample RNA-sequence
UAACGCAUAGUGUUUUUCCUCCACUUAUAUCGAAGGGUGUGUCUUGGAUCGCGCGGGUC
AAUGUAUAUGGUUCAUAUACAUCGACAGGCACGUAAUAAAGCGAGGGGUUCGAAUCCCC
CCGUUACCCCCGGUAGGGGCCCA
```

Fig.1: Example of STAR file: "STAR.1".

For now, choose [OK].

- Choose [Edit/Primary] to see how *STAR* converted this file. You see much more text

5.3 Sequence database

STAR can also read files from some sequence databases like EMBL. *STAR* can read these files and will remove all superfluous text. Let us read such a file from EMBL database.

- press [File/Import/Primary].
- change *.1* in the file name field to *.* and choose file TY.EMB that is supplied with *STAR*.
- *STAR* removes offensive characters and discovers several T-characters. It changes them to U and informs you of this conversion. You can only press [OK] to acknowledge this message; do so.
- *STAR* also discovers lower-case characters. The program will inform you of this discovery and asks if you agree to an automatic transformation to upper case. Upper-case characters are used in the prediction of stems, but nucleotides in lower-case are not. As

then only nucleotides. Remember that "*" is a comment; all non-sequence lines in the EMBL text are converted to comments.

- Leave the editor by clicking [Cancel].

5.4 Other databases

If you have sequences from another data bank, one that *STAR* doesn't know, you have two options. You must remove or *-comment the offensive characters such that only nucleotide characters remain. You can do that:

1. either in your own favourite editor and read the purified file later into *STAR*.
2. or in the editor of *STAR* which jumps in if you read the file with [File/Import/Primary] and *STAR* subsequently protests against the unknown characters in that file.

5.5 Finally

Try a (small) dirty text-file to see how *STAR* handles such a problem.

Summary of this chapter.

You can enter a primary sequence by...

- using *STAR* editor [Edit/Primary].
- loading a text file (only containing ACGUT) by [File/Import/Primary]. Lower-case characters can be converted to upper case.
- loading an EMBL file by [File/Import/Primary]. *STAR* will changes all superfluous text. Lower-case characters can be converted to upper case.

In each of these alternatives, T is converted to U.

6. FORCING AND PROHIBITING BASE-PAIRING.

In this chapter you learn how to enforce available data and ideas about a certain RNA structure on *STAR*. First you will learn how to enforce single-stranded regions by prohibiting basepairing of certain parts of the sequence. Secondly you will learn how to force a particular stem into the calculation.

First, if you haven't done so before, 1) start *STAR*, and 2) get the sequence from file TRNATHRT.1 using [File/Import/Primary].

6.1 Excluding nucleotides

Imagine (just as an example) that you are absolutely sure that the first ten nucleotides of TRNATHRT do not basepair. Then you don't agree with the prediction that nucleotides 1-7 pair with nucleotides 66-72.

To get a prediction that excludes those nucleotides you can explicitly force them to remain single-stranded:

- press [Calculate/Parameters]

You see a dialogue-box that reports at the right of the first line that currently the whole sequence is available for computation saying:

Nucleotids are marked single stranded

Below this text it says after "**Analysis**" that all nucleotides (1-76) are available for analysis.

- Type in the field after "**Single**" the numbers 1-10 (alternatively you could have specified in line "**Analyse**:[1] [76]" that you want [11]-[76] instead of [1]-[76] analysed).
- press [OK]

Let us first verify if these nucleotides are excluded from analysis indeed.

- press [View/Primary]

You will notice that the first 10 characters are in lower case. This means that they will not be used in the calculation. Just as we intended. Now let us make a prediction.

- press [Calculate/Compute/Greedy].

You see a new structure being computed without stem 1-7:66-72. Furthermore some stems are

similar to those in the previous prediction but others are different.

| | | | | | | | | | |
|---|----|----|----|-------|-----|------|---|---------|--|
| Pag.1 -Source "TRNATHRT.1/" - Date:1995 1 6 | | | | | | | | | |
| 27 | 32 | 38 | 43 | -11.4 | 4.0 | -7.4 | * | CACCCU | |
| | | | | | | | * | GUGGGA | |
| 49 | 53 | 61 | 65 | -9.8 | 4.5 | -5.3 | * | GGCAG | |
| | | | | | | | * | CCGUC | |
| 15 | 21 | 66 | 72 | -7.3 | 4.0 | -2.8 | * | GUUGGUA | |
| | | | | | | | * | CGACUAU | |
| 24 | 25 | 46 | 47 | -1.9 | 1.7 | -.7 | * | GC | |
| | | | | | | | * | UG | |

Look at the result in lego-view:

- press [View/Secondary/Lego].

Check bracket-view and span-view yourself.

6.2 Forcing stems

Imagine now that you suspect that nucleotides 1-6 do not pair with 67-72 but 2-4 with 13-15 and you know that 62-64 pair with 71-73. This should definitely change the prediction.

You can enforce this by:

- press [Edit/Secondary].

You see:

```

27 32 38 43
49 53 61 65
15 21 66 72
24 25 46 47

```

Apparently *STAR* proposes to enforce the previously predicted structure. Replace this into something like:

```

2 4 13 15 * My own intuition
62 64 71 73 * Discovery by J.Peters *

```

- click [File/Exit&Save].

Subsequently *STAR* computes the energies of these forced stems and you see:

```

2 4 13 15 -3.5 3.8 0.3
62 64 71 73 -5.3 3.7 -1.6

```

Apparently the first stem is not very probable, as its energy contribution is positive.

Nevertheless *STAR* dutifully accepted your orders.

- choose [Calculate/Compute/Greedy].

STAR first warns you that this will change the current structure. If you press [OK] *STAR* computes a new structure that contains the forced stems. You see the following structure predicted:

```

2  4 13 15  -3.5 3.8  0.3 * cug
                        * GAC
62 64 71 73  -5.3 3.7 -1.6 * UGC
                        * ACG
27 32 38 43 -11.6 4.1 -7.5 * CACCCU
                        * GUGGGA
16 19 56 59  -5.8 3.3 -3.6 * UUGG
                        * AACC
24 25 46 47  -2.1 0.7 -0.3 * GC

```

As I am convinced that 2-4:13-15 should be changed to 1-6:15-20 I make a new effort. The resulting structure is shown in Fig.1, 2 and 3.

```

                UGGUA
                U--|A
auagCUCA      C--|G
u---|---G     C--|G
a---|---U     C--|G
g---|---U     A--|U
u---|---G     GC--|GAG
c---|---G     C---|---G
g---|---UAGAG---|---UCGGCAGUCCAUCU--|AC
)::1:::2:::3:::4:::5:::6:::7:
1890123412347456745890123456789012678904
Fig.3: Lego-view of TRNATHRT with
forced stems.

```

```

-----
|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|
GCUGAUUAUAGCUCAGUUGGUAGAGCGCACCCUUGGUAAGGGUGAGGUCGGCAGUCCAUCUGCCUAUCAGCACCA
0:::1:::2:::3:::4:::5:::6:::7:::
12345678901234567890123456789012345678901234567890123456789012345
Fig.1: Span-view of TRNATHRT with forced stems.

```

```

$ 1 GCUGAUUAUAGCUCAGUUGGUAGAGCGCACCCUUGGUAAGGGUGAGGUCGG
% 1 ((((((-----))))))---((-----))---
$ 51 CAGUCCAUCUGCCUAUCAGCACCA
% 51 -----(((-----)))---
Fig.2: Bracket view of TRNATHRT with forced stems.

```

Summary of this chapter.

To exclude nucleotides from calculation...

1. choose [Calculate/Parameters] and
- 2/3. either specify which part to analyse in "Analyse[.]-[.]", or specify what to exclude in "NoFolding[.]-[.]".
4. predict using [Calculate/Compute].

To force certain stems into the prediction...

1. choose [Edit/Secondary].
2. type stems *STAR* should start with and press [File/Exit&Save].
3. predict using [Calculate/Compute/...]

7. COMPUTATION OF SECONDARY STRUCTURE USING STOCHASTIC FOLDING.

In this chapter we will predict the secondary structure using an algorithm for stochastic folding.

Apart from greedy folding, *STAR* can also apply another algorithm: stochastic folding ¹⁴. The idea behind this algorithm is that in RNA various stem combinations are possible and that the most favourable one is realised. Thus the algorithm randomly generates a set of alternatives using Monte-Carlo simulation. For addition of a stem, its frequency in the generated structures as well as its stability is taken into account. This algorithm takes more computer time than the greedy algorithm, but the results are often better.

7.1 Straight stochastic folding

Let us compute the structure using stochastic algorithm and compare that to the greedy folding. First greedy folding:

- choose [File/Import/Primary].
- choose file STAR.1 (this is just an artificial demonstration sequence).
- choose [Calculate/Compute/Greedy].

This yields the prediction:

| | | | | | | | | | |
|----|----|----|----|-------|-----|------|---|-----------|---|
| 26 | 34 | 70 | 78 | -12.0 | 4.5 | -6.6 | * | GAGGUCCCC | |
| | | | | | | | * | UUUUGGGGG | |
| 35 | 39 | 47 | 51 | -9.8 | 3.0 | -4.5 | * | GGCAG | |
| | | | | | | | * | CCGUC | |
| 10 | 14 | 21 | 25 | -6.8 | 4.4 | -2.8 | * | GCCUU | |
| | | | | | | | * | UGGGA | |
| 5 | 7 | 15 | 17 | -6.6 | 3.0 | -3.2 | * | GCC | |
| | | | | | | | * | UGG | |
| 57 | 59 | 67 | 69 | -1.8 | 3.3 | -.2 | * | UUC | |
| | | | | | | | * | GGG | 5 |
| 44 | 46 | 54 | 56 | -4.1 | 5.5 | -.1 | * | AAU | |
| | | | | | | | * | UUA | |

Fig.1: Result of greedy folding.

Next let us do stochastic folding:

- choose [Calculate/Compute/Stochastic].

At the end you see the prediction ¹⁵ of the stochastic folding. This is often similar to the prediction of the greedy algorithm, but occasionally it is not; we will elaborate on this later.

7.2 Stochastic folding of growing RNA

An interesting feature of *STAR* is that stochastic folding can simulate the RNA synthesis process. *STAR* will compute stems first for a short sequence, and continues to do so when gradually increasing the synthesised chain of nucleotides. We think that this growth leads the algorithm more or less along the folding pathway and so improves the quality of the result because it simulates the real process. It also increases the computer time substantially.

Let us do so for this sequence. This requires that we reflect about the synthesis process: what size of string do we start with, and how many nucleotides should this string increase in each computing (synthesising) step. The sequence is about 78 nucleotides long, so let us try an initial size of 25, and extend this size each step with 25 nucleotides.

- choose [Calculate/Parameters]
- look at the specs for "Stochastic" and set "Initial length" to [25]
- check if "increment" is [25].
- press [OK].
- press [Calculate/Compute/Stochastic].

The resulting structure is:

¹⁴ Gulyaev, A.P. (1991): The computer simulation of RNA folding involving pseudoknot formation. Nucleic Acids Research Vol.19(9)2489-2494.

¹⁵ We should stress again that we continuously update the energy rules to the newest values known. So these examples at the time of writing could give different result with your newest version of *STAR*.

```

35 39 47 51  -9.8 4.5 -5.3 * GGCAG
                        * CCGUC
  5  8 14 17  -6.3 3.2 -2.3 * GCCG
                        * UGGU
10 13 22 25  -6.8 3.2 -4.4 * GCCU
                        * UGGG
26 34 70 78 -12.0 4.5 -5.8 * GAGGUCCCC
                        * UUUUGGGGG
57 59 67 69  -1.8 3.3  -0.2 * UUC
                        * GGG      5

```

Fig.3: Stochastic folding with growth.

In this chapter we present 3 figures that show 3 different predictions (see Fig.4, 5 and 6). For a preliminary analysis we look at the total energies of these alternatives. They are respectively, -13.1 (accumulate column 7 of Fig.1), -19.1 and -11.1 (sum of column 7 in Fig.3) respectively, so we conclude that solution 2 (Figs.2 and 5) is the most stable and probably most reliable. Chapter 9 will deal with various alternatives more extensively.

Summary of this chapter.

Straight stochastic folding:

press [Calculate/Compute/Stochastic].

Stochastic folding with growth:

1. choose [Calculate/Parameters].
2. at the "Stochastic"-specifications choose appropriate "initial length[]" and "stepsize[]".
4. press [OK] and choose [Calculate/Compute/Stochastic].

```

UAGAGCCGAGCCUUGGUAAAAGGGUGAGGUCCCCGGCAGUCCAAUCUGC CUAUUUUUCAGCACCAGGGGGGGGUUUU
----[[[--(((([]]]--)))](((((([[[[---(((([]]]--))((-----)))))]]))

```

Fig.4: Prediction using greedy folding.

```

UAGAGCCGAGCCUUGGUAAAAGGGUGAGGUCCCCGGCAGUCCAAUCUGC CUAUUUUUCAGCACCAGGGGGGGGUUUU
-((((---((((---((-----)))-((((---((-----)))------)))))---)))))

```

Fig.5: Prediction using stochastic folding.

```

UAGAGCCGAGCCUUGGUAAAAGGGUGAGGUCCCCGGCAGUCCAAUCUGC CUAUUUUUCAGCACCAGGGGGGGGUUUU
----[[[--(((([]]]--)))](((((([[[[---(((([]]]--))((-----)))))]]))
12345678901234567890123456789012345678901234567890123456789012345678

```

Fig.6: Prediction using stochastic folding with growth.

8. GENETIC ALGORITHM.

This chapter will show you how to apply the Genetic Algorithm and briefly explains its output.

We feel that the genetic algorithm ¹⁶⁾ is the most superior algorithm because it closely follows the folding pathway to reach its final structure. It shows the intermediate stages where stems are formed and broken in the process. This ensures that the final prediction is more reliable. Moreover, showing the intermediates can be an invaluable tool for studies for example on metastable states.

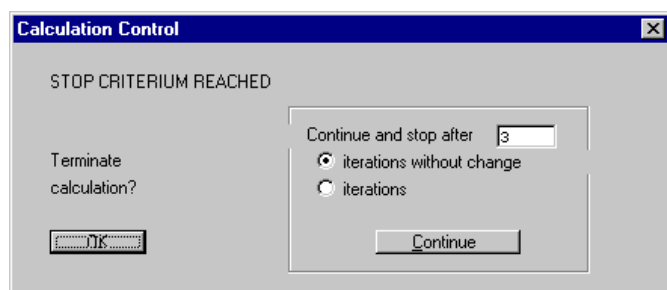
8.1 Straight G.A.

You may ask yourself is if you are interested in the intermediate structures or not.

If not, prediction with the GA is easy:

- Start *STAR*.
- Get file STAR.1 by [File/Import/Primary].
- Press [Calculate/Compute/Genetic].

After a number of iterations *STAR* shows you:



This dialog box tells you that *STAR* could not improve its prediction in the last 3 iterations. To be *very very* sure that this is not an intermediate structure you could ask to try more iterations (click on [Continue]). In general, 3 suffices. Occasionally, for example in a study with SV-11 RNA, a metastable state was reached quickly, but continuing to the final state required many more iterations. In such

cases you change 3 to –say– 10 and press [Continue] instead of [OK] to simulate longer.

Press [OK] and you will see:

```

5  8 14 17  -6.3 3.2 -2.3 * GCCG
                        * UGGU
10 13 22 25  -6.8 3.2 -4.4 * GCCU
                        * UGGG
26 34 70 78 -12.0 4.5 -6.6 * GAGGUCCCC
                        * UUUUGGGGG
35 39 47 51  -9.8 3.0 -4.5 * GGCAG
                        * CCGUC
57 59 67 69  -1.8 3.3 -0.2 * UUC
                        * GGG      5
44 46 54 56  -4.1 5.5 -0.1 * AAU
                        * UUA
  
```

8.2 What *STAR* reports

The GA gives various types of information.

- Approximate time needed 1.0 hours.

To help you estimate how long your computer will be occupied, *STAR* gives a very rough guess. It compares the time needed for previous iterations with the number of iterations to be done. As we have no way to exactly predict how long each simulation takes, please consider this as a crude indicator only.

The GA also reports for each iteration cycle:

- -----iteration 11: chain length=100

This tells you that *STAR* will start iteration 11 with a sequence of 100 nucleotides. You will see that this size increases at some steps, depending on the improvements in the previous iteration. This simulates the growth of the RNA during transcription.

- Mutating structure 1: -22.9 kcal/mol
: : : : : :
Mutating structure 5: -21.6 kcal/mol

This tells you that *STAR* uses a population of 5 alternative solutions and what the energy values are for each of them.

- Crossover: -22.9 kcal/mol

¹⁶ Batenburg, F.H.D. van & Gulyaev, A.P. & Pleij, C.W.A. (1995): An APL-programmed Genetic Algorithm for the prediction of RNA secondary structure. J.Theor.Biol. (174)269-280.

This tells you that *STAR* recombined the 5 solutions into a sixth one with energy value – 22.9 kcal/mol.

- | | |
|----------------|---------------|
| • Added | Removed |
| 35 39 47 51 | 97 99 109 111 |
| 99 105 114 120 | |

As long as improvement is possible, *STAR* reports how the best solution changed from the best one in the previous iteration. This is the most interesting output, because this reflects how the RNA folds step by step. In this example you see how stem 97-99:109-111 breaks in order to fold stem 99-105:114-120. Apparently the latter stem is much more stable than the former.

This information is the source of “transition-movies” like in Fig.1.

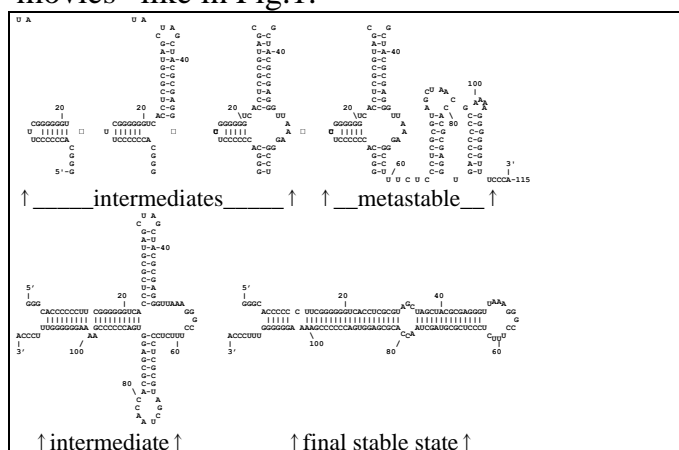


Fig.1: Transitions in SV-11 (graphic display not produced by *STAR*).

- Best free energy=–22.9 kcal/mol
Mean free energy=–17.4 (quality=0.87)

STAR compares the energy of the best solution with that of a random sequence. This ratio is used for “quality”. Values below 1 are low-energetic foldings that will unfold later to enable more stable stems, often in a quick “zippering” process. Such intermediates exist in nature and can have important functions ¹⁷.

¹⁷ Gultyaev, AP & Batenburg, FHDv & Pleij, CWA (1995): The computer simulation of RNA folding pathways using a genetic algorithm. J.Mol.Bio.(250)37-51.
Nagel, JHA & Gultyaev, AP & Gerdes, K. & Pleij, CWA

8.3 Folding process

To analyse metastable states you should look at structures that are relatively stable. During simulation you could look at the screen every once in a while. But don't be afraid to miss steps occasionally; fortunately *STAR* keeps track of its intermediate simulation results so you could also analyse the process after the simulation. To see the pathway:

- Press [View/Secondary/Pathway].

You see something like:

| Step | Added | Removed | Energy | Length |
|------|-------------------------|-----------|--------|--------|
| 1 | 5 8 14 17 | | –2.9 | 20 |
| 2 | 10 14 21 25 | 5 8 14 17 | –9.7 | 30 |
| | 5 7 28 30 | | | |
| 4 | 36 39 47 50 | | –12.3 | 50 |
| 6 | 35 39 47 51 36 39 47 50 | | –15.3 | 60 |
| 8 | 31 34 67 70 | | –20.8 | 70 |

This shows the various changes during simulation. Column 1 reports the steps were the simulation changed; you see that step 3, 5 and 7 did not produce an improvement. You see in column 2 and 3 how stems are removed that were added previously (for example the first stem 5-8:14-17 has been removed at step 2). In column 5 you see how the simulation “synthesised” more and more of the molecule.

Summary of this chapter.

Straight GA prediction:

- [Calculate/Compute/Genetic].

Succumb metastable states:

- at the end of computation “continue and stop after[3]”, set [3] to [10] and press [Continue]

Analyse the folding pathway:

- After computation press [View/Secondary/Pathway].

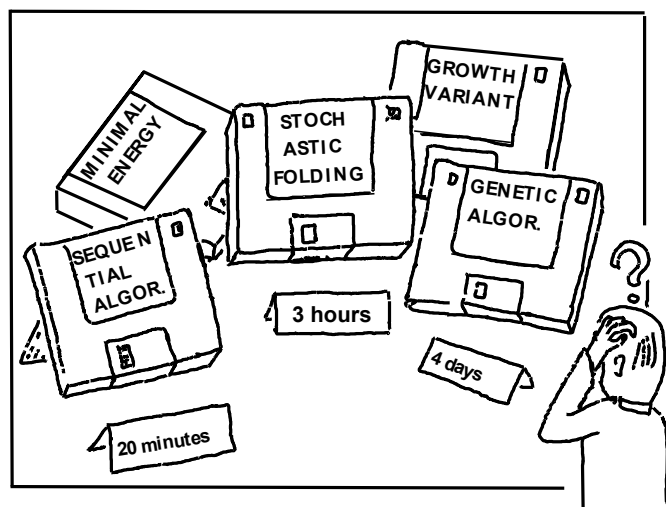
(1999): Metastable structures and refolding kinetics in hok mRNA of plasmid R1. RNA (5)1408-1419

9. COMPARING DIFFERENT STRUCTURES (ALGORITHMS).

This chapter talks about alternative structures due to different parameter settings and due to different algorithms. Finally it give suggestions how to deal with such alternatives.

9.1 Three algorithms

STAR contains 3 algorithms that may predict more or less conflicting structures. Which one is the best?



The **greedy algorithm** is a good algorithm that yields predictions that are realistic and has about the same quality as the minimal energy approach. As the greedy algorithm follows a “greedy” pathway, it doesn’t necessarily ends in a minimal energy prediction. This is often better, occasionally it is not.

Moreover, this algorithm is rather quick.

The **stochastic algorithm** doesn’t necessarily ends in a minimal energy prediction either, because it follows a pathway too. This “majority” pathway is often better than the “greedy” pathway because it takes combinations of stems into account.

However this algorithm need much more computer time.

Both algorithms form stems step by step, but do not break them. The **genetic algorithm** will make and break bonds during simulation. Therefore the GA is the superior algorithm because it follows the real folding pathway

most closely and as such is a more realistic simulation of the real process.

Besides of its closer realism, the genetic algorithm has another advantage. It shows the intermediate stages where stems are formed and broken in the process. This can be an invaluable tool for some studies for example studies on metastable states.

It's most important drawback is, however, that this simulation is *very* time consuming; about 10 times as slow as the greedy algorithm.

9.2 Parameter settings

All algorithms have some parameters that you can change and which affect the results. This tutorial is not the place to deal with it in depth. We choose the default settings of all parameters that worked best for our research, but specific investigations may demand other parameter settings. We will mention the most important one here.

9.2.1 greedy algorithm

The parameters for the greedy algorithm are very empirical and are not amenable for change.

9.2.2 stochastic algorithm

The stochastic algorithm adds stems using a Monte-Carlo simulation that can possibly favour a stem in one simulation, but another stem in the next.

An important parameter setting is the growth variant. Folding a full-grown RNA might result in a different structure than when the RNA already folds partly during its synthesis. By default, the stochastic algorithm folds the full-grown RNA. Let us see what happens if we fold during growth.

First, if you haven't done so before, 1) start *STAR* and 2) get sequence TRNATHRT.1 using [File/Import/Primary]. Next:

- choose [Calculate/Parameters] and look at "Stochastic".
- set "Initial length[]" to [25].
- set "increment[]" to [25] and press [OK].
- press [Calculate/Compute/Stochastic].

9.2.3 Genetic algorithm

The most useful parameters are as follows:

Parameter "Population size[5]" determines the speed and quality. A higher number (say 10) steers *STAR* quicker (well... less iterations but much slower per iteration) and "rougher" towards a solution. A small number will create a subtler walk in the solution-space and sometimes get lost; 5 is a good compromise.

Parameter "Growth" is set to "on" by default. When "on" the program will simulate the synthesis of the RNA: it will slowly increase the size, meanwhile folding and unfolding stems.

You can *not* set parameter "IterationStop" in this parameter panel, but you can change it at the end of a simulation run. The IterationStop parameter determines how persistent *STAR* keeps looking for improvements. Normally 3 to 5 is fine, and this is the default. However if you expect metastable structures and want *STAR* to pass beyond, you should set it to a higher number (10, 20 or even 30) in the dialog box at the end of a run.

9.3 Good, better, best?

How to make sense of all these alternatives?

As stated before, in general the GA is the best algorithm, the greedy folding is the simplest and the stochastic algorithm is somewhere in between. Nevertheless, sometimes the GA will err where the greedy algorithm is fine. So you always need to judge the result critically.

A second way to judge between alternatives is to look at the energy values. Stems with low negative values are more reliable than others.

A third criterium is to look at the frequency of stem occurrences. The similarity of alternative foldings can reflect their reliability: stems evolved in several structures are probably more reliable than those in a single simulation only.

Finally, there is a method to zoom in iteratively to better and better solutions. Just select the "best" stems from several alternative solutions (see previous criteria) and enter these as forced stems. Do new predictions with this as a starting point and extend your "best" stems.

Summary of this chapter.

- For quick predictions: Greedy Algorithm.
- For slower but more reliable predictions: Stochastic Algorithm.
You could improve by small initial length and small increment using [Calculate/Parameters]
- For slow but very reliable predictions: Genetic Algorithm.
Set population size in [Calculate/Params]: increase value for quick and rough walk, decrease value for a subtler walk.
- For study of intermediate foldings:
 - fold with [Calculate/Compute/Genetic] and pass beyond metastable states by increasing IterationStop to 15-30 in the dialog at the end of the simulation
 - to look at pathway use [View/Secondary/Pathway].

10. TIPS AND TRICKS.

This chapter lists a number of quick investigations and a number of tips that may help in your daily work.

In this chapter you will learn how to...

1. do a quick and rough investigation for a few main stems.
2. recognise pseudoknots quickly.
3. find energy of a structure you have in mind.
4. Do several simulations unattended.
5. Process circular RNAs.

First, if you haven't done so before, 1) start *STAR*, and 2) get sequence TRNATHRT.1 using [File/Import/Primary].

10.1 Quick investigation of main stems

If you have a very big sequence, all algorithms will require a long time to calculate a prediction. If you are not interested in all stems, you can ask *STAR* to limit the number of computed stems to the main ones.

Say you are only interested in the 2 main stems. For this we use the *greedy* folding algorithm:

- choose [Calculate/Parameters]
- set "Stop after stem [1000]" to [2] and
- press [OK]
- choose [Calculate/Compute/Greedy].

10.2 Quick pseudoknot recognition

If you have done a calculation and if you are interested in pseudoknots in particular, they are very easily spotted in bracket-view.

```
UAGAGCCGAGCCUUGGGCUAAGGGCGGGGUCCCCGGC
-----((((((-----)))-----)))-----
```

```
UAGAGCCGAGCCUUGGCGGAAGGGGCUUGGUCCCCGGC
-----(((([[[-----]])-----]])-----
```

Fig.1: Hairpin with bulge loop (top) versus pseudoknot (bottom) in bracket-view.

If you see parenthesis only, no pseudoknot is predicted (remember that *STAR* predicts only those pseudoknots that its energy rules allow).

If you see square brackets (as shown in Fig.1), a pseudoknot is predicted. For example:

- import PKB131.1 with [File/Import/Primary].
- predict with [Calculate/Compute/Greedy].
- now look at [View/Secondary/Bracket].

10.3 Quick energy calculation

Imagine you have a structure suggested by evolutionary data, or by biochemical analysis. If MFOLD or *STAR* do not predict that structure you might be interested to know how big the difference is. You can find this by asking *STAR* for the energy value of that particular structure.

Imagine that you do not agree with *STAR* its prediction for TRNATHRT stem 1-7:67-72 and that you are convinced that the RNA folds 1-6:15-20. To know the energy of this structure:

- choose [Edit/Secondary]
- enter your structure (not just that one stem, but the complete structure):

```
27 32 38 43 * proven structure
49 53 61 65 * proven structure
1 6 15 20 * my own hunch
```
- press [File/Exit&Save].

STAR shows you the structure and the associated energy; something like:
 Energy (re)calculation of forced stems:

```
27 32 38 43 -11.4 4.9 -6.5 * CACCCU
                                * GUGGGA
49 53 61 65 -9.8 4.9 -4.9 * GGCAG
                                * CCGUC
1 6 15 20 -5.2 5.5 0.3 * GCUGAU
                                * UGGUUG
```

Apparently your alternative stem costs 0.3 kcal/mol and the whole structure is (-6.5 + -4.9 + 0.3=) -11.1 kcal/mol.

10.4 Printer/screen problems

Some users reported printer problems, which we couldn't reproduce, unfortunately.

As long as we have not repaired this problem, you could write your output to file and view or print the result using your text processor.

For example:

- choose [Export/Primary] for sequence or [Export/Secondary/...] for structure or pathway.
- read file in your favourite word processor.
- select all text and choose font Courier.

10.5 Circular versus linear RNA

To force a circular RNA you should know one reliable hairpin (if you don't, you could try the most stable one using a preliminary prediction). Let us say for example 27-32:38-43 in tRNATHRT.

1. Rearrange the nucleotides in the sequence using [Edit/Primary] such that the new ends fall within the loop. In our example remove nucleotides 1-32 and add them to the right of the last nucleotide (number 76).
2. Force the hairpin in this new configuration using [Edit/Secondary]. In the example: 6-11:71-76; because 27-32 are now at positions 71-76 and 38-43 are now at 6-11.

10.6 Temperature

STAR uses the Gibbs energy values at 37°C by default. You can change the temperature by:

- Choose [Calculate/Parameters]
- Change "temperature[37]" to the appropriate value, for example [20]
- Start computation: [Calculate/Compute/...]

10.7 Several simulations

As the computations often need a lot of time, it could be useful to let STAR work at night unattended. This you can do as follows:

- first make sure you have one or more proper sequence files (lowercase for nucleotides to ignore, others in uppercase; no non-nucleotide characters).

- press [Calculate/Batch/...] (choose the algorithm you want).
- the file chooser dialog appears; choose the first sequence –say STAR.1 and press [OK].
- the file chooser dialog appears again, choose your next sequence file, say TMVRNA.1 and so on.
- once you say [Cancel] to the file-dialog, STAR starts computations for each sequence file.

STAR will save the computation results in "*.star" files. So the next day you can look at the results using [File/Open].

Summary of this chapter.

For quick investigation of *main stems*:

1. press [Calculate/Parameters].
2. set "stop after stem[]" to small number.
3. press [OK] and start computation.

For quick *pseudoknot spotting*:

1. choose [View/Secondary/Bracket].
3. look for brackets amidst parenthesis.

For *energy calculation* of hypothetical stems:

1. choose [Edit/Secondary].
2. enter structure up to hypothetical stem.
3. close edit box.

For *Print/View problems*:

1. output to file with [File/Export/...].
2. read file into your favourite editor with a non-proportional font.

For *several simulations*:

1. prepare sequence files.
2. choose [Calculate/Batch/...].
3. add each sequence file, end with [Cancel].
4. next day read results with [File/Open].

For *circular RNA*:

1. determine one reliable hairpin.
2. move nucleotides from 1-within hairpin.
3. enter this hairpin as forced structure.