CONSTRUCT's User Guide

This manual is under development . . . forever.

Institute of Physical Biology
Heinrich Heine University Düsseldorf

October 22, 2007

Open Publication License v1.0, 8 June 1999

I. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

> Copyright (c) <year> by <author's name or designee>. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, vX.Y or later (the latest version is presently available at http://www.opencontent.org/openpub/).

The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section VI).

Commercial redistribution of Open Publication-licensed material is permitted.

Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

II. COPYRIGHT

The copyright to each Open Publication is owned by its author(s) or designee.

III. SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

SEVERABILITY. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

NO WARRANTY. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

IV. REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

1. The modified version must be labeled as such.
2. The person making the modifications must be identified and the modifications dated.
3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
4. The location of the original unmodified document must be identified.
5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

V. GOOD-PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

1. If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.
3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

VI. LICENSE OPTIONS

The author(s) and/or publisher of an Open Publication-licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

A. To prohibit distribution of substantively modified versions without the explicit permission of the author(s). "Substantive modification" is defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.

To accomplish this, add the phrase 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.' to the license reference or copy.

B. To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

To accomplish this, add the phrase 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.' to the license reference or copy.

CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

CONSTRUCT (first publication Lück *et al.*, 1999) is a tool for the prediction of consensus structure of homologous RNA sequences. It combines standard sequence alignment, thermodynamic RNA structure prediction (Hofacker *et al.*, 1994; Hofacker, 2003), comparative sequence analysis [mutual information content, Chiu & Kolodziejczak (1991); *RNAalifold* score, Hofacker *et al.* (2002) including stacking Lindgreen *et al.* (2006)] and (in contrast to other programs) user intelligence.
CONSTRUCT allows for prediction of

- optimal secondary structures,
- suboptimal secondary structures,
- tertiary interactions like base triples and pseudoknots.

The prerequesite for this predictions is the existence of a structurally correct alignment. To circumvent this problem, one usually starts with a pure sequence alignment, predicts a consensus structure, and repeats these two steps until a satisfying solution is found. Since the RNA alignment problem is still an unsolved problem (Sankoff, 1985) and most databases contain hand curated alignments, the need for a tool which aids the user in creating such correct alignments is obvious.
CONSTRUCT provides a "elaborate GUI" (Zuker, 2000) that displays superimposed dotplots and the mutual information content in a consensus dotplot and allows the user to correct the initial sequence alignment with it's alignment editor.

1.1. Flowchart

For an overview of the general procedures have a look at Fig. 1.1.

Yellow part:

Top: The procedure starts with an initial sequence alignment (*e. g.* using CLUSTAL).

Bottom: Independently from this alignment, for each sequence is computed either a structure distribution (represented as dotplot triangle) using RNAFOLD or a dot plot containing all possible helices (Tinoco *et al.*, 1971).

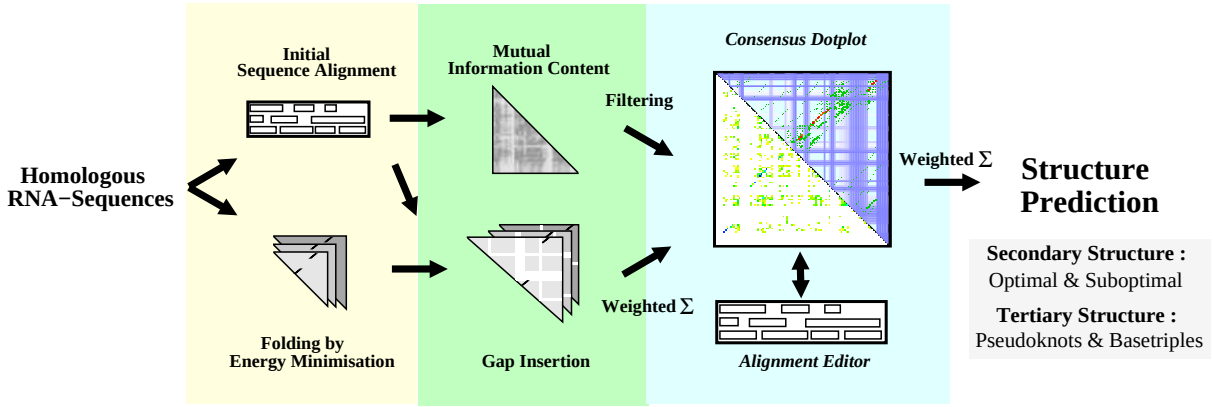

Figure 1.1: CONSTRUCT Flowchart See text for a description.

These two time consuming steps are executed only once.

Green part:

Top: The mutual information content (MIC) for the alignment is computed, which serves as a measure for compensatory base-pair changes. MIC detects correlations between characters in columns and thus is not restricted to Watson-Crick (WC) pairs; this is fortunate for detection of regions with non-WC pairs and/or base triples, but might lead to noise. As an alternative the *RNAalifold* measure can be used; this takes into account only WC and G:U pairs.

Bottom: The gaps from the sequence alignment are inserted into the base-pair matrices. Thus all matrices are of the same size and can be overlaid.

The MIC usually contains noise, which can be filtered on user's request. Similarly the individual base-pair matrices may be weighted to avoid over-representation of some (highly similar) sequences or sequence groups.

Blue part: The graphical user interface (GUI) of CONSTRUCT. Base-pair matrices and covariation plot are displayed as well as the editable alignment. Each time the user changes the alignment the dotplot is updated by reiterating through the green part.

Structure Prediction: The internal computation base of the structure prediction is a (user chosen) weighted combination of covariation scores and base-pair probabilities. Secondary structure as well as tertiary interactions may be predicted. The structures may be displayed in several ways.

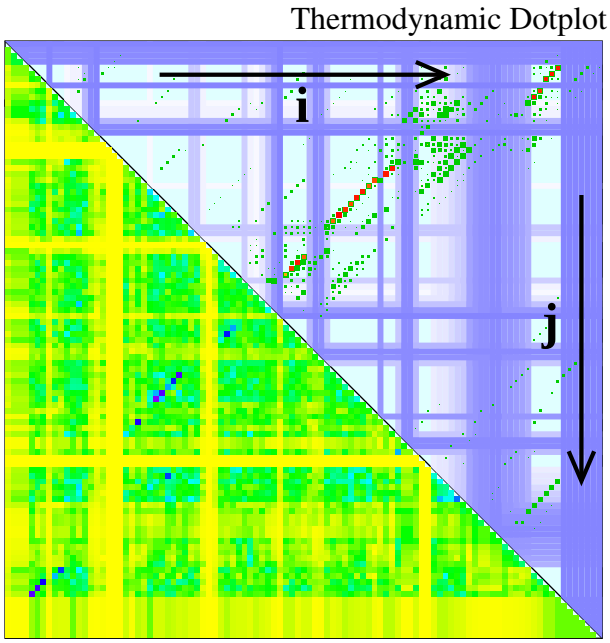

Figure 1.2: CONSTRUCT Dotplot. The top-right triangle contains the overlay of base-pair matrices calculated by RNAFOLD; the lower-left triangle contains the covariation matrix. The sequence is from left to right (i; 5' to 3') and from top to bottom (j).

Top-right: Dot plot based on thermodynamics

Green dots: Size of dots proportional to probability of a base pair in an individual sequence

Yellow to red dots: Consensus pairing probability of all sequences in the alignment

White to purple bars: Gaps in the alignment; with increasing darkness of color the number of gaps in this alignment column increases.

Bottom-left: Dot plot based on covariation Rainbow colors (from yellow to black) denote increasing statistical significance for a correlated pair at this position. The noise (yellow up to light blue) can be suppressed on command.

1.2. Dotplot

In the dotplot window of CONSTRUCT the overlay of the base-pair matrices (calculated by RNAFOLD or *tinoco*) together with the consensus pairing matrix are viewed in the upper-right triangle; the covariation matrix (calculated on demand by either mutual information content or *RNAalifold* score) is viewed in the lower-left triangle. For an example see Fig. 1.2.

1.2.1 Thermodynamic Dotplot

The upper-right triangular plot contains information about possible base pairs based either on thermodynamic prediction by RNAFOLD or on possible base pairs (simple dotplot).

Basepairs: green squares

In case of thermodynamics the area of a dot is proportional to the probability $p_k(i, j)$ of forming a base pair at this position i, j in an individual sequence k.

In case of a simple dotplot the area of a dot is proportional to the thermodynamic stability of the helix to which it belongs.

Clicking with the (left or right) mouse button highlights the corresponding nucleotides (5' and 3' nucleotide) in the editor window.

Basepairs: blue squares

Blue "base pairs" belong to a sequence selected in the editor window.

Consensus Basepairs: yellow to red squares

The area and the color of a dot are proportional to "probability" $P_c(i,j)$ of a consensus base pair at position i, j in an alignment with N sequences:

$$P_c(i,j) = \left(\frac{\sum_{k=1}^{N} w_k \cdot p_k(i,j)^{1/3}}{\sum_{k=1}^{N} w_k} \right)^3$$

Each individual sequences may have a weight w_k to avoid over-representation of highly similar sequences. The exponents ($1/3$ and 3) help to suppress low pairing probabilities in single sequences.

Summed Gaps: white to purple lines

The darkness of background lines (bars) is proportional to the number of gaps in this column.

1.2.2 Mutual Information Content

The lower-left triangular plot contains statistical information about possible base pairs based either on mutual information content (MIC; Chiu & Kolodziejczak, 1991) or *RNAalifold*'s covariation measure (Hofacker *et al.*, 2002). Both measures are in the range from 0. to 1.; accordingly positions are colored from yellow over green/blue/red to black.
The mutual information content I is calculated by

$$I_{x_i,x_j} = \sum_{x_i,x_j} f_{x_i,x_j} \log_b \left(\frac{f_{x_i,x_j}}{f_{x_j} f_{x_j}} \right)$$

with the fraction f of nucleotides of type $x \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ in columns i and j; as basis b of the logarithm might be chosen either 2 (bits) or e (nits).
FIXME: *RNAalifold* measure

2. INSTALLATION

If you ever run in trouble with CONSTRUCT don't hesitate to contact us (see chapter 5).

2.1. Installing precompiled packages

Get the precompiled package fitting your OS-Distribution. Debian-users use *dpkg -i* and RedHat-users *rpm -i*.

2.2. Installation from source

2.2.1 Requirements

CONSTRUCT uses TCL/TK and C. Since CONSTRUCT uses a custom interpreter you will have to install the TCL and TK devel packages $>= 8.4$. Next, CONSTRUCT reads base-pair matrices (dotplots) produced by RNAFOLD; that is, you have to install the Vienna package. Installation of libZ is recommended, so that compression of the base-pair matrices is supported.

2.2.2 Instructions

First unpack the distribution, *i. e.* using:

```
$ tar xzf ConStruct-<Version>.tar.gz
```

and change to the newly created CONSTRUCT-<Version> directory.
As CONSTRUCT was developed using the GNU autotools, this directory contains a file named INSTALLATION, which contains generic installation instructions.
In short, installation is done by the typical GNU "triple jump".

```
$ ./reconf
$ ./configure
$ make
$ make install
```

Depending on your system you will have to pass some options to configure. It needs to find your TCL/TK config-files (tclConfig.sh and tkConfig.sh). In case these files are not found automatically by configure, specify their location using the –with-tcl and the –with-tk flag.

2.3. Example

Installation to your home directory under Debian-GNU/Linux.

```
$ ./configure  \
>  --prefix=$HOME/local \
>  --with-tcl=/usr/lib/tcl8.4 \
>  --with-tk=/usr/lib/tcl8.4
$ make
$ make install
```

That is, the programs (*cs_fold*, *cs_dp*, *etc.*) are installed in `$HOME/local/bin/` and CONSTRUCT's libraries are installed in `$HOME/local/lib/construct/`; the compiler looks for the TCL and TK libraries in `/usr/lib/tcl8.4` and `/usr/lib/tcl8.4`, respectively.

2.4. Bugs

In case *cs_dp* starts with an error message like

```
Error in startup script: can't find package drawstructcore
    while executing
"package require drawstructcore"
...
```

`mkIndex.tcl` has missed to add the `libdrawstructcore` to the `pkgIndex.tcl` in `/usr/local/lib/construct/drawstruct/`. Edit the file
`/usr/local/lib/construct/drawstruct/pkgIndex.tcl`
and add the following line at the end

```
package ifneeded drawstructcore 0.9 \
    [list load [file join $dir libdrawstructcore]]
```

2.5. MacOS X

Installation under MacOS X requires FINK . **FIXME**

2.6. Windows

There is a ongoing effort to port ConStruct to the CYGWIN environment. Until now we had no success.

3. QUICKSTART

For the really impatient ...

- Sequence Alignment
  Align your set of homologous RNA-sequences using your favorite sequence alignment tool (*e. g.* using CLUSTAL, T-COFFEE, PRRN, ...).
- Sequence Folding
  Invoke *cs_fold*
  Load the alignment ("Browse", "Load Seq")
  (or do both in one step: `cs_fold -f seq.vie`)
  Write the project file ("Write Project file")
  Execute the folding method ("Execute method")
  Exit *cs_fold*
- Main GUI
  Invoke *cs_dp*
  Open the created project File ("File"/"Open Project")
  (or do both in one step: `cs_dp -f seq.proj`)
  At this point you can play around
  Modify the alignment to be structural correct, *i. e.* move the nucleotides in the alignment window, so that the base pairs (green squares) in the dotplot window are superimposed (red consensus base pairs). Now you may invoke optimal or suboptimal structure-prediction, create a structural alignment output, or predict base triples and pseudoknots.

4. IN-DEPTH GUIDE

4.1. Initial Sequence Alignment

First of all you have to create an initial sequence alignment. We call it initial since the user will have to modify it to be structurally correct. Take your unaligned sequences and align them using your favorite sequence alignment tools like CLUSTAL, T-COFFEE or PRRN. In case you already have an alignment, skip this step.

4.2. *cs_fold*

cs_fold creates either thermodynamically predicted base-pair matrices or thermodynamically weighted base-pair dotplots for each sequence in an alignment. It takes an RNA alignment as input and folds each sequence either with RNAFOLD (Hofacker, 2003) or with *tinoco*, which belongs to the CONSTRUCT package. Both programs are invoked via *cs_fold*.

4.2.1 Alignment Loading

In the sequence-frame (see Fig. 4.1) click the "Browse" button and select the alignment you wish to load. In the following an alignment-file name `<alignment>.ext` is assumed. Then click on "Load Seq". The IDs of your sequences will be displayed in the "Fold Options"-frame. (Skip this point if you provide the alignment via the command-line option `-f`; see section 4).

4.2.2 Fold Options and Method

Now you have to choose a folding method (RNAFOLD or *tinoco*) from the fold-method frame (see Fig. 4.1). *cs_rnafold* (default) is strongly recommended. Either method will produce a base-pair matrix for each sequence. With RNAFOLD the files are named `<sequence_id>_dp.ps` (or `<sequence_id>_dp.ps.gz` if libZ-compression is supported); with *tinoco* the files are named `<sequence_id>_ti.ps` (or `<sequence_id>_ti.ps.gz` if libZ-compression is supported).
All these matrix files are standard PostScript files, which are viewable f. e. via `ghostview` or

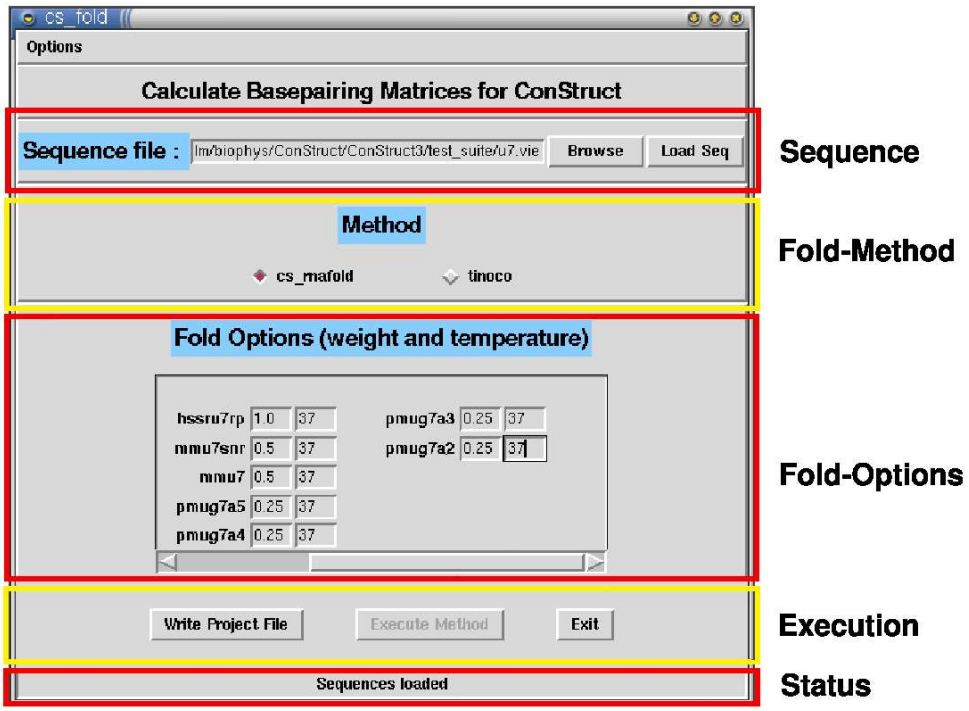

Figure 4.1: ***cs_fold*** The GUI is straight forward to handle: work your way from top to bottom as described. The single frames are highlighted and labeled.

printable to a PostScript printer. In addition RNAFOLD creates for each sequence a PostScript file containing the optimal structure, named <*sequence_id*>*_ss.ps(.gz)*). You can delete these files if you want, which will save you some disk space. All following procedures only depend on the created base-pair matrices.

You may specify some options specific to each folding method (fold-options-frame): For each sequence a weight may be assigned. This weight, ranging from 0. to 1., is a factor used to avoid overrepresentation of a set of sequence families. For example having a set of ten homologous RNAs, where four of them originate from the same organism whereas the remaining are from different organisms, each of the four sequences should be assigned a weight of 0.25 to give these four together a weight identical to each of the remaining sequences. These weights are used while computing the consensus-basepair probabilities and the consensus sequence.

4.2.3 Finishing the Project

Proceed by clicking the "Write Project File" button in the execution-frame (see Fig. 4.1). A plain-text file containing all relevant information (*e. g.* file names, weights *etc.*) named <*alignment_root*>*.proj* will be written to directory where the alignment file resides. This file is essen-

Table 4.1: Available command-line options for *cs_fold*.

flag	value	description
-h		show help
-f	FILE	load this aligned sequence file
-t	INT	set default temperature for RNAFOLD ($4 < \# <= 90$)
-l	INT	set minimal helix length for *tinoco* ($1 < \# <= 20$)

tial.

Now you can execute the folding method by invoking "Execute Method". This will call either RNAFOLD or *tinoco* with the appropriate options for each sequence.

The progress status can be observed in the status bar (see Fig. 4.1).

4.2.4 Options

There are only a few command-line options for *cs_fold*; these are listed in Table 4.1.

4.2.5 Tips & Tricks

cs_fold also creates two other files named <*alignment_root*>.*log* and <*alignment_root*>.*os*. The first one is simply a log file of the RNAFOLD execution. The second one contains the computed optimal structure for each sequence.

You can add data from structure probing experiments to the project file. E. g. if you know that nucleotides 4 to 10 are unpaired but 12 to 15 are paired and 20 pairs with 25, add the following line:

```
mapinfo: 4-10:u  12-15:p 20:25
```

To actually use this info enable the appropriate options:

option	description
Show Mapping Info in Struct.Aln	Colorizes violations in the structure alignment window
Use Mapping Info in Dotplot	Hides false positives (no matrix changes!)

```
/// ConStruct Project-File Version 3.0
Project-Name: secis_methanococcus_construct
Alignment: secis_methanococcus_construct.vie

#comment lines start with a dash and are ignored
#comments inside sequence entries have
#    special tags (see below) and are stored
#example entry:
#begin entry
#   id:      <string> e.g. h_SelD
#   weight:  <int/double> e.g. 0.125
#   seqlen:  <int> e.g. 67
#   bpmat:   <file> e.g. h_SelD_dp.ps[.Z|.gz]
#   foldcmd: <string> e.g. rnafold -T 37 -p -d 3
#   comment: <string> e.g. this is a comment
#   mapinfo: <string> e.g. 3-5:p 8-11:u 12:24
#end entry

begin entry
    id:      M_jannaschii_sps
    weight:  1.0
    seqlen:  35
    bpmat:   M_jannaschii_sps_dp.ps.gz
    foldcmd: RNAfold -T 37 -p -d 3
    comment:
    mapinfo:
end entry
...
```

Figure 4.2: Example of a project file.
The second line gives the project's name, which is used for labelling graphics produced by *cs_dp*.
The third line gives the name of the alignment file.
For each sequence entry in the alignment file an entry follows that gives the sequence ID, its weight, the sequence length, the name of the dotplot matrix, and the command used to produce the dotplot matrix.
Comment lines start with "#" and may appear everywhere.

4.3. The Project File

The project file with extension ".proj" is written by *cs_fold* and read by *cs_dp*. As shown in Fig. 4.2 it contains:

- a headline for identification as a CONSTRUCT file;
- a name, which is depicted in most windows of *cs_dp* and also in graphics produced by *cs_dp*;
- the file name of the alignment, this name is used as basis for most further file names;
- an entry for each sequence given in the alignment.

Each entry contains at least

id: the sequence ID, which has to be unique in the alignment

weight: the sequence weight

seqlen: the length of the sequence (without gaps)

bpmat: the file name of the basepair matrix

foldcmd: the program and options used to produce the basepair matrix

Note that full file names are allowed; that is, neither the alignment file nor the basepair matrices have to reside in the same directory as the project file.
Gaps present in the alignment file are inserted into the matrix files by *cs_dp*. If pairing constraints are known for individual sequences, for these sequences individual pairing matrices can be used; for example:

- Produce a file (f. e. with name `dummy.vie`) containing only the sequences for which constraints are known; in case of RNAFOLD this may look as follows:

  ```
  > M_jannaschii_sps
  acgaugugccgaacccuuuaagggaggcacaucga
  .<(.......xxx|...................)..
  ```

- Calculate the matrix file:
  `RNAfold -C -T 37 -p -d 3 < dummy.vie > dummy.log`
  Note the option `-C`; all other options are as for the other sequences. This produces new matrix files with identical names as without constraints. If necessary, compress the matrix files:
  `gzip -9 *_dp.ps`

- Run *cs_dp* with the "old" project file

4.4. *cs_dp*

cs_dp is the main application of CONSTRUCT. After loading the project file created with *cs_fold*, it reads the alignment and the basepair probability matrices. Afterwards the gaps from the alignment are introduced into the matrices. This way they all are equally sized and can be superimposed in the dotplot window.
If the alignment is correct in terms of structure, structural elements like helices (diagonals in the dotplot) are superimposed too.

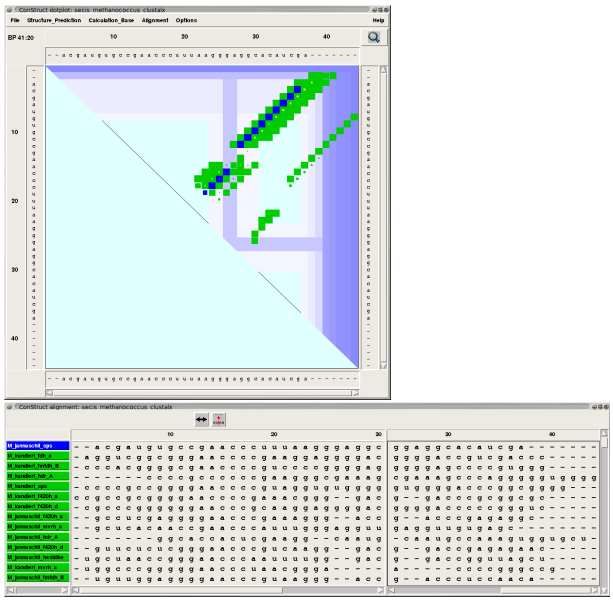

Figure 4.3: *cs_dp*'s graphical user interface. View after loading of `secis_clustalx.proj`, a CLUSTAL alignment of 14 SECIS elements from methanococcal bacteria.
Top: Dotplot with thermodynamic base pairing matrices. The sequence of `M_janaschii_sps` is selected (blue dots); base pairs of all other sequences are depicted in green. Light-blue to purple bars denote gaps in the alignment.
Bottom: Alignment window. The left column contains the sequence IDs on green background; on blue background is given the ID of the selected sequence `M_janaschii_sps`. The middle and right column contain two views of the alignment. Above the middle column are the two buttons ("double arrow" and "insert", which allow for movement of a selected nucleotide, a stretch of nucleotides or a block of nucleotides in the alignment.

In most cases the initial alignment will not be correct, which can be easily recognized by the clustering of homologous helices. The alignment must be modified by selecting the misaligned nucleotide range and moving it to the structurally correct position. Consequently the dotplot will be updated.

4.4.1 The Alignment Window

The alignment window consists of three major columns, which contain the sequence IDs and two identical copies of the sequence alignment.

ID column: The background colors (blue and green) correspond to the colors of dots in the base-pair dotplot; that is, the "blue sequence" is selected for editing. Selection of a sequence is done either by left-clicking onto the sequence ID or by left-clicking to any nucleotide of a sequence.

Sequence alignment columns: It's convenient to use the left and right columns for viewing the 5'- and 3'-part of a helical region. The following actions are available:

- Mouse over nucleotide in selected sequence:

In the dotplot window the nucleotide is highlighted in yellow. If the nucleotide belongs to a basepair, the corresponding dot is highlighted in lightblue.

- First ($2n+1; n > 0$) mouse click to a nucleotide:
  Select the sequence to which the nucleotide belongs; base pairs of this sequence are highlighted in blue in the dotplot. If the nucleotide neighbors a gap on its left and/or right position, it can be moved to this gap position by clicking to the double arrow; left-button clicking moves left, right-button clicking moves right.
  A left click to the insert arrow selects the nucleotide stretch up to the next 5'-gap and moves it towards the gap. A right click to the insert arrow selects the nucleotide stretch up to the next 3'-gap and moves it towards the gap.
- Second ($2n+2; n > 0$) mouse click to a nucleotide:
  If the second nucleotide belongs to the same sequence as the first, a stretch of nucleotides is selected. This stretch is moved towards a neighboring gap by a left or right click to the double arrow.
- Strg + Second ($2n+2; n > 0$) mouse click to a nucleotide:
  If the second nucleotide belongs to a different sequence then the first, a block of nucleotides is selected. The block is moved by a click to the double arrow, if it is neighboring a gap (without exception) and none of the bordering nucleotides is a gap.

4.4.2 The Dotplot Window

The dotplot window contains the basepair matrix files (calculated by RNAFOLD `-p` or *tinoco*) in the upper-right triangle. Base pairs of individual sequences are shown as green dots with size proportional to the probability of that base pair. Base pairs of a selected helix are shown in blue. Consensus base pairs are depicted as dots with size and color (from yellow to red) proportional to their probability. Gaps are shown as lines with a color from white to purple proportional to number of gaps in the corresponding alignment column.
The lower-left triangle of the dotplot window contains statistical information about possible base pairs, if calculated via mutual information content or *RNAalifold* covariation score (see below).
The dotplot is surrounded by the selected sequence.
Top-left is shown the position of the mouse in sequence coordinates.
The following actions are available:

- If the mouse is positioned over a green or blue base pair, the corresponding nucleotides are highlighted in the alignment window.

  Clicking with the left or right mouse button to a green or blue base pair, centers and highlights the corresponding nucleotides in the alignment window.

- If the mouse is positioned over a consensus base pair (yellow to red), the corresponding columns in the alignment window are highlighted with colors from black over yellow to red proportional to the probability of the base pair in the individual sequences.

  Clicking with the left or right mouse button to a consensus base pair centers the corresponding columns in the alignment window.

  The latter option can be disabled (`Options→Highlight Consensus-Nts`) if it's too slow with many sequences.

4.4.3 The "File" menu

Open Project: Load an existing project file.
Equivalent to a restart with *cs_dp* `-f <name>.proj`

Save Alignment: A warning is given if an existing alignment file will be overwritten.
The default name is built from the old alignment filename by adding "_construct" to the root-name; that is, if the old name is `<name>.vie`, as new name is proposed `<name>_construct.vie`. This new name is not inserted into the project file!

Print Dotplot: The actual content of *cs_dp*'s dotplot window is written to

- printer
- color printer
- file (with default filename `<project name>_dp.ps`)
- screen

These output devices—printer, color printer, and screen—are handled by the variables `opts(print_cmd,printer)`, `opts(print_cmd,colorprinter)`, and `opts(print_cmd,screen)` set in the files *cs_dp*, lines 112–114, or the resource file `$HOME/.cs_wish_v3.rc`, lines 37–39, to `lpr`, `lpr -Php2250`, `gv --media=a4 -`, respectively. Modify them for all users in *cs_dp* or for your own needs in the resource file.

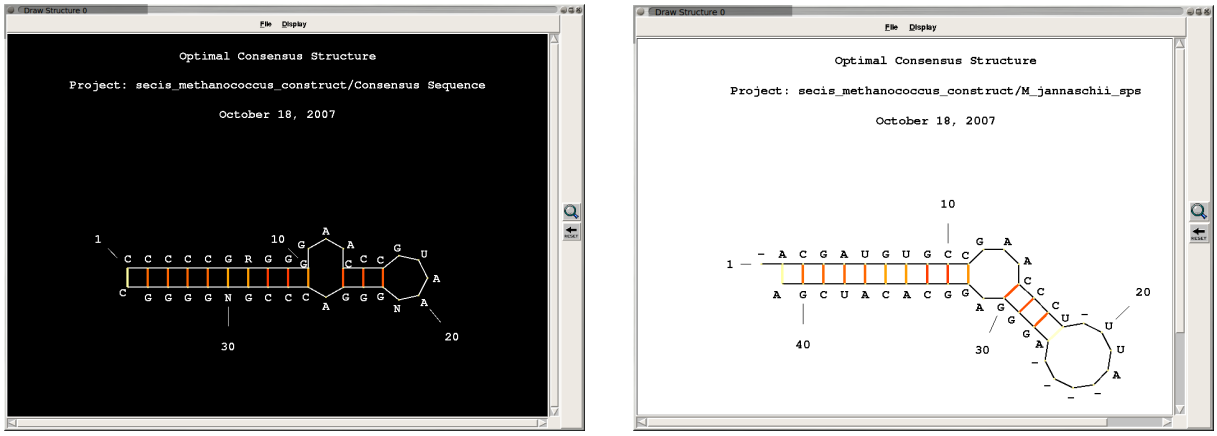

Figure 4.4: ***DrawStruct*.**

Print Alignment: See the prior entry.

Exit: On exit a warning is given if the alignment was modified but not saved.

4.4.4 The "Structure Prediction" menu

Optimal Structure prediction performed by dynamic programming (Nussinov *et al.*, 1978) maximizing the weighted combination of the thermodynamic and covariation pairing probability.

Suboptimal Structure prediction performed by dynamic programming (Steger *et al.*, 1984; Zuker, 1989).

Tertiary Structure prediction of tertiary interactions performed by maximum weighted matching procedures (Tabaska *et al.*, 1998) with two sub-options:

Pseudoknots:

Basetripels:

Each of the top-level entries has four alternatives of output; examples are shown in Figs 4.4 to 4.7:

Draw Structure:

Circles:

Structural Alignment:

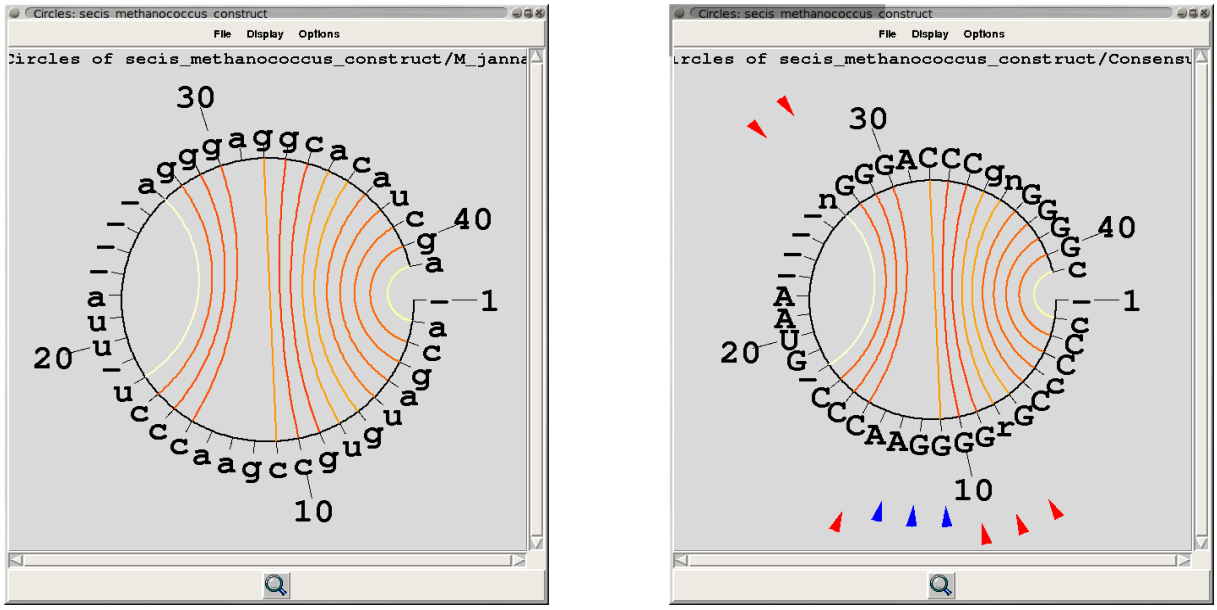

Figure 4.5: *Circles*.

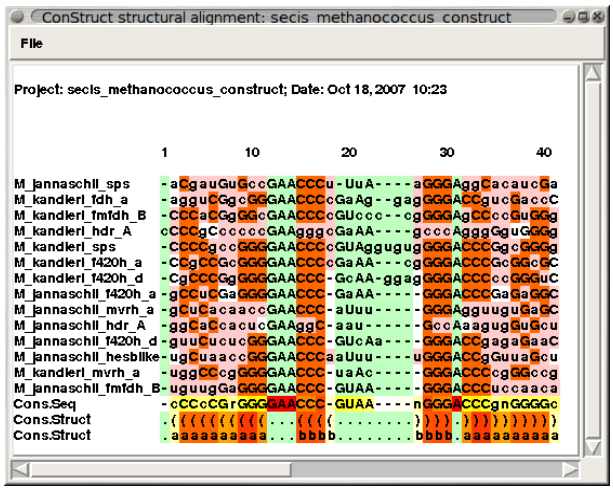

Figure 4.6: Structural alignment.

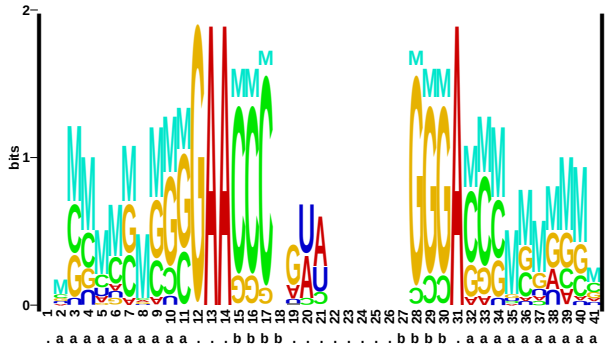

Figure 4.7: Structure Logo.

Request Structure Logo from

http://www.cbs.dtu.dk/~gorodkin/appl/slogo.html

4.4.5 The "Calculation Base" menu

The consensus structure prediction is done by dynamic programming in case of optimal and sub-optimal structures and by maximum weighted matching in case of tertiary interactions. In all three cases a consensus dotplot is the basis. This dotplot may consist either on the weighted and summed-up thermodynamically predicted dotplot (or helix dotplots) or a covariation dotplot or a combination of both. The thermodynamics dotplot is shown in the top-right triangle of *cs_dp*'s dotplot window; the covariation dotplot in the lower-left triangle. The covariation dotplot is created by either of two methods: the mutual information content (MI; Chiu & Kolodziejczak, 1991) or the *RNAalifold* covariation measure (CV; Hofacker *et al.*, 2002). Which of both methods including several options for them and the factors for combining thermodynamics and covariation dotplots are defined in the calculation-base menu.

The probability p_c of a consensus base pair at positions i and j is given by

$$p_c(i,j) = \left(\frac{\sum_{s=1}^{N} w_s \cdot p_s(i,j)^{1/3}}{\sum_{s=1}^{N} w_s} \right)^3$$

where p_s(i,j) is calculated either by RNAFOLD or *tinoco* (in *cs_fold*), N is the total number of sequences, and $0 \leq w_s \leq 1$ is the user-defined weight of sequence s. This weighting can be used to avoid over-representation of a closely related sequence family in comparison to other sequences; weights can be given in *cs_fold* or edited in the project file. The exponentiation helps to reduce low pairing probabilities from individual sequences.

The MI at two aligned nucleotide positions i and j is defined as:

$$\mathrm{MI}_{ij} = \sum_{\mathrm{X,Y}} f_{ij}(\mathrm{XY}) \log_b \frac{f_{ij}(\mathrm{XY})}{f_i(\mathrm{X}) \cdot f_j(\mathrm{Y})}$$

where f_i(X) and f_j(Y) are the frequencies of the nucleotide types X $\in$ {A,U,G,C} and Y $\in$ {A,U,G,C} at aligned positions i and j, and f_{ij}(XY) is the joint frequency of finding X at i and Y at j. In addition, the user may apply a normalization method (Martin *et al.*, 2005), which enhances separation of truly correlated positions from background correlations. That is done by dividing the MI by the joint entropy

$$h_{ij} = \sum_{\mathrm{X,Y}} f_{ij}(\mathrm{XY}) \log_b f_{ij}(\mathrm{XY}) \tag{4.1}$$

the upper bound of the MI. For statistical analysis of the MI, maximum likelihood or unbiased probability estimation (Gutell *et al.*, 1992) in nits ($b = \mathrm{e}$) (Chiu & Kolodziejczak, 1991) or bits ($b = 2$) (Schneider *et al.*, 1986) are available.
In comparison to the MI, the *RNAalifold* covariation score measures compensations in Watson-Crick and wobble base-pairs (Hofacker *et al.*, 2002) only, which is advantageous during search for helices. The meaningfulness of this score can be further improved by taking stacking into account (as shown in Lindgreen *et al.*, 2006).
The linear combination of the thermodynamic and the covariation pairing probabilities

$$P_c(i,j) = \quad w_{\mathrm{TD}} \cdot \begin{cases} p_c(i,j) & \text{if } p_c(i,j) > t_{\mathrm{TD}} \\ 0 & \text{otherwise} \end{cases} \\ + \; w_{\mathrm{CV}} \cdot \begin{cases} \mathrm{CV}_{i,j} & \text{if } \mathrm{CV}_{i,j} > t_{\mathrm{CV}} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

allows for thresholds t and a relative weighting ($w_{\mathrm{TD}} + w_{\mathrm{CV}} = 1$) of thermodynamics and covariation. The thresholds serve to further reduce the statistical noise and to suppress false positive base pairs and can be adjusted by the user.

View of covariation plot:

- **Compute/Show Covariation:** According to the other selected options (see below) either the MIC or the CV are calculated and depicted in the lower-left triangle of the dotplot window.
- **Hide Covariation:** Remove the content of the lower-left triangle of the dotplot window.
- **Threshold (color mapping):** The statistical probability $p_{i:j}$ for a basepair at position $i : j$ is normalized to values between 0. and 1.; this range is mapped to rainbow colors. Values below 0.3 in case of MIC and below 0.15 in case of CV are likely to be noise; this depends on the number of sequences in the alignment and on their average pairwise sequence identity. A slider may be used to fix the depicted range of values to get an optically pleasing range of colors.

  See also entries of “Combined dotplot” on page 21.

Type of covariation measure:

- **Use *RNAalifold* score:** For an explanation see Hofacker *et al.* (2002) and Lindgreen *et al.* (2006).
- **Use Mutual Information:** For an explanation see Chiu & Kolodziejczak (1991) and Martin *et al.* (2005).

***RNAalifold* options:**

- with or without stacking (Lindgreen *et al.*, 2006)

MI options:

- **Use $\log_e$ for probability estimation** according to Chiu & Kolodziejczak (1991)
- **Use $\log_2$ for probability estimation** according to Schneider *et al.* (1986)
- **Use unbiased probability estimation** according to Chiu & Kolodziejczak (1991)
- **Use maximum likelihood estimation** according to Gutell *et al.* (1992)
- **Pair-entropy normalization** according to Martin *et al.* (2005); see (4.1)

Combined dotplot: See equation (4.2) for the thresholds t and the relative weights w.

- **TD threshold:** t_{TD}
  To reduce the thermodynamics noise and to exclude low-probability base pairs raise this threshold. A good value is $t_{\text{TD}} = 0.03$
- **MI threshold:** t_{CV}
  To reduce the MI or CV noise raise this threshold; result is directly visible in the lower-left triangle of *cs_dp*'s dotplot window. Good values are $t_{\text{MI}} = 0.3$ for MI calculation and $t_{\text{CV}} = 0.15$ for *RNAalifold* CV calculation, respectively.
- **Function:** $P_c(i,j) = w_{\text{TD}} \cdot p_c(i,j) + w_{\text{CV}} \cdot \text{CV}_{i,j}$
  See (4.2)

4.4.6 The "Alignment" menu

Sequence Search: allows to search for subsequences in all or selected sequences of the alignment by regular expressions; hits get a selected background color in the alignment window. The search is performed in the unaligned sequences; that is, gaps do not play a role.

Map Nucleotides to Helix: replaces helices by alphabetical characters and loops by dots. For characters see the last "Cons.Struct." line in the "Structural alignment" output in section 4 and Fig. 4.6.

Clear the above mapping

4.4.7 The "Options" menu

Use Consensus Sequence for Structure Prediction: instead of using the selected sequence the consensus sequence is displayed in *DrawStruct* and *Circles* output (Figs 4.4 and 4.5).

Remove Gaps for DrawStructure: as default *DrawStruct* (Fig. 4.4) uses the aligned sequence including gaps for output; this option remove the gaps.

Allow single basepairs: most routines avoid to include lonely (non-stacking) base pairs in their predictions; sometimes, however, such lonely pairs are helpful during optimization

Show Consensus Basepairs: show/hide the yellow to red dots in the dotplot

Show Gaps: show/hide the white to purple bars in the dotplot

Show Basepairs: show/hide the green and blue base pairs in the dotplot

Show mirrored rectangle: show/hide the mirrored cursor
As default only a single mouse pointer is shown in the dotplot; on demand a second mouse pointer (a small square) is shown in the other triangle part of the dotplot

Highlight Consensus Nucleotides: FIXME

Number of suboptimal structures

Show Mapping Info in Structural Alignment: If mapping information—knowledge on paired/unpaired nucleotides from other sources—is available this can be given in the

"mapinfo" entries of the project file (see section 5). This information is shown/hidden in Circles plots.

Show Mapping Info in Dotplot: FIXME

The last three options add output to the text window written during creation of the "Structural alignment" (Fig. 4.6).

Show Sequence Statistics in Structural Alignment: This option adds output like average pairwise sequence identity (APSI), sum-of-pairs score (SOP), etc.

Show Structure Statistics in Structural Alignment: This option adds a table on significance of predicted helices; f. e.:

```
Helix b:
            BP    NoBP   BP   CsBP   Prob(CS)   Prob(TD)    I(x,y)          χ²(df,p)      R1      R2   Pairs
   15:  30=C:G       0   12      2      0.844      0.844   0.425>=0.382 = x2( 9,.70)   0.498   0.498   C:G G:C
   16:  29=C:G       0   12      2      0.846      0.846   0.425>=0.382 = x2( 9,.70)   0.498   0.498   C:G
   17:  28=C:G       0   13      1      0.829      0.829   0.435>=0.382 = x2( 9,.70)   0.572   0.572   C:G
   18:  27=-:n       7    0      7      0.100      0.100   0.333>=0.332 = x2(16,.10)   0.246   0.244   N:N
  Helix len= 4       7   37     12      0.493      0.183   0.403>=  (geometric means)
```

The first column gives position $i : j$ and consensus basepair,
the second the number of non-basepairs,
the third the number of Watson-Crick and wobble base pairs,
the fourth the number of covarying base pairs,
the fifth the sum of the thermodynamic base-pairing probabilities of the pairs, **FIXME**
the sixth the mutual information content $I(x, y)$ of the two positions,
the seventh the $\chi^2(df, p)$ statistics with degrees of freedom df and significance level p,
forget the eigth and nineth column (or see Gutell *et al.*, 1992), and
the tenth the pairs that contribute most to the MI.
The last line gives

The χ statistics is performed only if $\log_e$ is used for probability estimation.

Show Pattern Statistics in Structural Alignment: results in a table which might help in designing a pattern for a pattern search algorithm.

4.4.8 Command line options

All options are listed in Table 4.2.

Table 4.2: Available command-line options for *cs_dp*.

flag	value	description
-h		show help and exit
-v		be verbose
-V		print version and exit
-d		print debug messages
-t		do some time measurements
-f	FILE	open project file on startup

4.4.9 Tips & Tricks

4.5. Other executables

- *cs_remgaponly_cols* Removes columns that only consist of gaps from an alignment file.
  Usage:
  ```
  cs_remgaponly_cols -f <FILE> [-o <FILE>]
  ```
  The input file has to be a multiple sequence file (in any format accepted by the *seqio* package; Eddy, 2005) and writes the output file in Vienna format. If input and output filename might be identical. If the `-o` options is omiited output is written to `stdout`.
- *cs_proj_conv* Converts a project file from the format used by CONSTRUCT version 2 into a format used by CONSTRUCT version 3. The old matrix files, however, are not readable by *cs_dp* v3; thus it might be easier to recreate the project file by *cs_fold*. Then one has to enter again any weights.
- *cs_shift* Calculates the "necessary moves" (shifts) of nucleotides to rearrange a predicted target alignment into a trusted template or reference alignment. Normalization? **FIXME**
  Usage:
  ```
  cs_shift -t <FILE> -p <FILE>
      -t ''trusted template alignment''
      -p ''predicted target alignment''
  ```
- *cs_wish* **FIXME**
- *cs_struct_displ* Displays a structure file (in either CONSTRUCT or *connect* format) via *Circles* or *DrawStruct*.
  Usage:

```
cs_struct_display [options] [-f] <FILE>
    -f ``structure file''
    -d ``display method''
```

Valid formats of the "structure file" are CONSTRUCT's consensus format (extension `.cs`), "connect" format (extension `.ct`), or *RNAalifold*'s PostScript output (extension `.ps` or `.eps`). Available display formats are either *Circles* (with option `-d circles`; see Fig. 4.5) or *DrawStruct* (with option `-d drawstruct`; see Fig. 4.4).

- *csfoldbatch* **FIXME**
- *csdpbatch* **FIXME**
- *tinoco* **FIXME**
- *cs_add_ti_dp* **FIXME**

5. CONTACT

Visit the ConStruct's Homepage to get hopefully more recent information about the status of CONSTRUCT. If you ever run into trouble while installing or working with CONSTRUCT don't hesitate to contact the authors.

6. Downsides and Bugs

7. FIXME

8. REFERENCES

Chiu, D.K. & Kolodziejczak, T. (1991). Inferring consensus structure from nucleic acid sequences. *Comp. Appl. Biosci.*, **7**, 347–352. 1, 2, 5, 5, 5

Eddy, Sean R. (2005). SQUID - C function library for sequence analysis. http://selab.wustl.edu/cgi-bin/selab.pl?mode=software#squid. 5

Gutell, R.R., Power, A., Hertz, G.Z., Putz, E.J. & Stormo, G.D. (1992). Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Res.*, **20**, 5785–5795. 5, 5, 7

Hofacker, I.L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429–3431. 1, 2

Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, S., Tacker, M. & Schuster, P. (1994). Fast folding and comparsion of RNA structures. *Monatsh. Chem.*, **125**, 167–188, http://www.tbi.univie.ac.at/~ivo/RNA/. 1

Hofacker, Ivo L., Fekete, Martin & Stadler, Peter F. (2002). Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, **319**(5), 1059–1066. 1, 2, 5, 5, 5

Lindgreen, S., Gardner, P.P. & Krogh, A. (2006). Measuring covariation in RNA alignments: physical realism improves information measures. *Bioinformatics*, **22**, 2988–2995. 1, 5, 5

Lück, R., Gräf, S. & Steger, G. (1999). *ConStruct*: A tool for thermodynamic controlled prediction of conserved secondary structure. *Nucleic Acids Res.*, **27**, 4208–4217, http://www.biophys.uni-duesseldorf.de/local/ConStruct/ConStruct.html. 1

Martin, L. C., Gloor, G. B., Dunn, S. D. & Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics*, 21(22), 4116–4124. 5, 5

Nussinov, R., Pieczenik, G., Griggs, J.R. & Kleitman, D.J. (1978). Algorithms for loop matchings. *SIAM J. Appl. Math.*, **35**, 68–82. 4

Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825. 1

Schneider, T.D., Stormo, G.D., Gold, L. & Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. *J. Mol. Biol.*, **188**, 415–431. 5, 5

Steger, G., Hofmann, H., Förtsch, J., Gross, H.J., Randles, J.W., Sänger, H.L. & Riesner, D. (1984). Conformational transitions in viroids and virusoids: Comparison of results from energy minimization algorithm and from experimental data. *J. Biomol. Struct. Dyn.*, **2**, 543–571. 4

Tabaska, J.E., Cary, R.B., Gabow, H.N. & Stormo, G.D. (1998). An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics*, **14**, 691–699. 4

Tinoco, Jr, I., Uhlenbeck, O.C. & Levine, M.D. (1971). Estimation of secondary structure in ribonucleic acids. *Nature*, **230**, 362–367. 1

Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science*, **244**, 48–52. 4

Zuker, M. (2000). Calculating nucleic acid secondary structure. *Curr. Opin. Struct. Biol.*, **10**, 303–310. 1