
rna-tools documentation

Marcin Magnus

Mar 16, 2026

USER DOCUMENTATION

1	Install	3
1.1	Get the Source Code	3
2	Quickstart	5
2.1	fetch a structure from the PDB database	5
2.2	fetch a biological assembly	5
2.3	get sequences of a bunch of PDB files	6
2.4	get secondary structures of your PDB files	6
2.5	delete a part of of your structure	8
2.6	get numbering of your structure and rename chains	8
2.7	edit your structure (rename chain)	8
2.8	extract part of your structure	9
2.9	find missing atoms in my structure	9
2.10	mutate residues	10
2.11	add missing atoms	11
3	rna-pdb-tools	15
3.1	rna_pdb_tools	15
3.1.1	rna_pdb_tools.py	15
3.1.2	get RNAPuzzle ready	22
3.1.3	get sequence	25
3.1.4	fetch	26
3.1.5	fetch Biological Assembly	26
3.1.6	delete	27
3.1.7	edit	27
3.1.8	RNAStructure (rna_tools_lib)	27
3.2	Standardize your PDB files	37
3.2.1	rna_standardize.py	37
3.2.2	Atoms order	42
3.2.3	OP3	43
3.3	Torsion angle analysis	44
3.3.1	rna_torsions.py	44
3.4	PDB Edit Bfactor/Occupancy	45
3.4.1	rna_pdb_edit_occupancy_bfactor.py	45
3.5	Add chain to a file	47
3.5.1	rna_add_chain.py	47
3.6	Measure distance between atoms	48
3.6.1	pdb_measure_atom_dists.py	48
4	rna-tools	51

4.1	RNA Sequence	51
4.1.1	Installation	51
4.1.1.1	ContextFold	51
4.1.1.2	ViennaRNA	51
4.1.1.3	ipknot OSX	51
4.1.1.4	RNA Structure	52
4.1.1.5	MC-Sym	52
4.1.2	FAQ	52
4.1.3	TIPS	52
4.2	RNA Secondary Structure	56
4.2.1	rna_dot2ct.py	57
4.2.2	Secondary structure format conversion	58
4.3	Search	59
4.3.1	Blast PDB	59
4.3.2	Rfam Search	60
4.4	RNA Alignment	61
4.4.1	RNASEq	62
4.4.2	RNAalignment	67
4.4.2.1	rna_alignment_get_species.py	72
4.4.2.2	rna_alignment_calc_energy.py	74
4.4.2.3	rna_align_get_ss_from_fasta.py	76
4.4.2.4	rna_align_get_ss_from_stk.py	76
4.4.2.5	rna_align_distance_to_seq.py	77
4.4.2.6	rna_align_foldability.py	78
4.4.3	Random assignment of nucleotides	83
4.4.3.1	random_assignment_of_nucleotides.py	83
4.4.4	CMAlign	85
4.4.5	RChie	87
4.4.6	Renumber a pdb file according to alignment	88
4.4.6.1	renum_to_aln.py	88
4.5	Root Mean Square Deviation (RMSD)	90
4.5.1	rna_calc_rmsd	90
4.5.1.1	rna_calc_rmsd	90
4.5.1.2	rna_calc_rmsd_multi_targets.py	95
4.5.2	rna_calc_rmsd_trafl	96
4.5.2.1	rna_calc_evo_rmsd	96
4.5.2.2	rna_cal_rmsd_trafl_plot	97
4.5.3	rna_calc_rmsd_all_vs_all	98
4.5.3.1	rna_calc_rmsd_all_vs_all	98
4.6	Interaction Network Fidelity (INF)	99
4.6.1	rna_calc_inf.py	99
4.7	RNA filter (DCA)	101
4.7.1	Calculate distances based on given restraints on PDB files or SimRNA trajectories	101
4.7.1.1	rna_filter.py	101
4.7.1.2	rna_dca_mapping.py	102
4.7.2	Show distances in PyMOL	103
4.7.2.1	rna_ec2x.py	103
4.7.3	Convert pairs to SimRNA restraints	104
4.7.3.1	rna_pairs2SimRNArestrs.py	104
4.7.4	Get a list of base pairs for a given “fasta ss” file.	105
4.7.4.1	rna_ss_get_bps.py	105
4.7.5	Get a diff of pairs	106
4.7.5.1	rna_pairs_diff.py	106
4.8	Contacts classification & secondary structure detection	106

4.8.1	3DNA (contacts classification & secondary structure detection)	106
4.8.2	ClARNA (contacts classification)	108
4.9	RNA 3D model quality assessment	109
4.9.1	RASP	109
4.9.2	Dfire	110
4.9.3	RNAkb	111
4.9.3.1	rna_mq_rnakb.py	112
4.9.4	QRNA	112
4.9.5	eSCORE	112
4.9.6	3dRNAscore	113
4.9.7	RNA3DCNN	113
4.9.8	FARNA	115
4.9.9	ClashScore	116
4.9.10	AnalyzeGeometry	117
4.10	RNA 3D structure prediction	118
4.10.1	ROSETTA	118
4.10.1.1	Run (modeling)	118
4.10.1.2	Get a number of structures	123
4.10.1.3	Get a head of a Rosetta silent file	124
4.10.1.4	Cluster	126
4.10.1.5	Minimize	127
4.10.1.6	Extract lowscore decoy	129
4.10.1.7	Check progress	130
4.10.2	SimRNA	131
4.10.2.1	Select low energy frames	131
4.10.2.2	Extract	131
4.10.3	SimRNAweb	132
4.10.3.1	Download files of a SimRNAweb run	132
4.10.4	SimRNATrajectory	133
4.11	RNA Refinement (QRNAS)	135
4.11.1	rna_refinement.py	135
4.12	RNA Molecular Dynamics (MD)	138
4.13	diffpdb	138
4.14	RNA clustering with CLANS (clanstix)	141
4.14.1	Color groups	142
4.14.2	How to use ClanstixRNA?	142
4.15	Misc	144
4.15.1	Plotting	144
4.15.1.1	rna_plot_hist.py	144
4.15.1.2	rna_plot_density.py	145
4.15.2	rna_sali2dotbracket	146
4.15.2.1	rna_sali2dotbracket	146
4.15.3	Cluster load	147
4.15.4	RNAkb	148
5	rna-tools & PyMOL [PyMOL4RNA]	149
5.1	Inspect structure	149
5.2	Show base pair edges	152
5.3	PyMOL4Spliceosome: all spliceosome structures in one PyMOL session	154
5.4	PyMOL: Color by conservation	155
5.5	PyMOL Drawing	155
5.6	Install	157
5.7	Libs: Pucker analysis	158

6	rna-tools & Mac/Linux	161
6.1	PyMOL preview generator	161
7	rna-tools & Jupyter Notebooks / Colab	163
7.1	Share your notebooks	164
7.2	Use Colab	164
7.3	Learn Jupyter	164
8	rna-tools & Emacs	165
9	rna-tools & geekbook	167
9.1	Draw VARNA-based image of RNA secondary structure	167
10	RNA Puzzle Submission	169
11	Selection	171
12	Tips	173
12.1	Run in batch	173
12.2	Using sed	173
12.3	In PyMOL	174
12.4	In Python	176
12.5	Working with cluster	177
12.6	Numbering line used in my flat-file notes	178
12.7	TER format	178
12.8	Add missing atoms	179
12.9	Test for Cuda	179
13	Examples of workflows	181
13.1	Example #1	181
13.2	Example #2	182
13.3	Example #3	183
13.4	Example #4 Calculate RMSDs of unstandardized structures (RNA Puzzle #1)	183
14	Install & Configure	187
14.1	Pip as a developer	187
14.2	Test if installed	187
14.3	Configuration	188
14.4	All requirements	188
14.5	Test all	188
14.6	PyMOL4RNA: adv config	189
15	Git Quickref	191
15.1	Git sheet cheat	192
15.2	Learn Git	192
15.3	Git GUI	194
16	Configuration	195
17	Documentation	197
17.1	Generate docs	197
18	Add a new tool to the package	199
19	Indices and tables	201

Python Module Index

203

Index

205

rna-tools: a toolbox to analyze structures and simulations of RNA

The code of the project can be found at GitHub <https://github.com/mmagnus/rna-tools>.

This documentation can be found at <http://rna-tools.rfd.io>.

The PDF version of documentation can be found at [here](#).

If something does not work for you, please make an issue, using <https://github.com/mmagnus/rna-tools/issues>.

INSTALL

This guide assumes you already have python and pip installed.

To install pytube, run the following command in your terminal:

```
$ pip install rna-tools
```

1.1 Get the Source Code

rna-toos is actively developed on GitHub, where the source is available:

You can either clone the public repository:

```
$ git clone http://github.com/mmagnus/rna-tools.git  
$ cd rna-tools && pip install -e .
```

or (to install in current ./src/):

```
pip install -e git+http://github.com/mmagnus/rna-tools.git#egg=rna-tools
```


QUICKSTART

This guide will walk you through the basic usage of rna-tools.

Let's get started with some examples.

2.1 fetch a structure from the PDB database

Example:

```
$ rna_pdb_tools.py --fetch 1xjr  
downloading...1xjr ok
```

2.2 fetch a biologicaly assembly

Example:

```
$ rna_pdb_tools.py --fetch-ba 1xjr  
downloading...1xjr_ba.pdb ok
```

or over a list of pdb ids in a text file:

```
$ cat data/pdb_ids.txt  
1y26  
1fir  
  
$ while read p; do rna_pdb_tools.py --fetch-ba $p; done < data/pdb_ids.txt  
downloading...1y26_ba.pdb ok  
downloading...1fir_ba.pdb ok  
  
$ ls *.pdb  
1fir_ba.pdb 1y26_ba.pdb
```

2.3 get sequences of a bunch of PDB files

Example:

```
$ rna_pdb_tools.py --get-seq *.pdb
# 1xjr
> A:1-47
GGAGUUCACCGAGGCCACGCGGAGUACGAUCGAGGGUACAGUGAAUU
# 6TNA
> A:1-76
GCCGAUUUAgCUCAGuuGGGAGAGcGcCAGAcUgAAgAucUGGAGgUCcUGUGuuCGaUCCACAGAAUUCGCACCA
# rp2_bujnicki_1_rpr
> A:1-15
CCGGAGGAACUACUG
> B:1-10
CCGGCAGCCU
> C:1-15
CCGGAGGAACUACUG
> D:1-10
CCGGCAGCCU
> E:1-15
CCGGAGGAACUACUG
> F:1-10
CCGGCAGCCU
> G:1-15
CCGGAGGAACUACUG
> H:1-10
CCGGCAGCCU
```

in some more fancy way ;-)

```
$ rna_pdb_tools.py --get-seq \
    --oneline 3_bujnicki_1_rpr* \
    --color-seq --compact
```

```
(py37) [mx] rp03$ git:(master) * rna_pdb_toolsx.py --get-seq --oneline 3_bujnicki_1_rpr* --color-seq --compact
CUCUGGGGGGGCCGGUUUUCGGGUCGGCCGGGGGGGGCGGCUUGGGGGCUUGGGGGGUGGCUUCGGGGCGGGGGCGGGG # A:1-84 # 3_bujnicki_1_rpr
CUCUGGGGGGGCCGGUUUUCGGGUCGGCCGGGGGGGGCGGCUUGGGGGCGGGGGUGGCUUCGGGGCGGGGGCGGGG # A:1-84 # 3_bujnicki_1_rpr_mut
```

2.4 get secondary structures of your PDB files

Python parser to 3dna <<http://x3dna.org/>>.

Installation:

```
# install the code from http://forum.x3dna.org/downloads/3dna-download/
Create a copy of the rna_x3dna_config_local_sample.py (remove "_sample") present in rna-
↳tools/rna_tools/tools/rna_x3dna folder.
Edit this line :
BINARY_PATH = <path to your x3dna-dssr file>
matching the path with the path of your x3dna-dssr file.
e.g. in my case: BINARY_PATH = ~/bin/x3dna-dssr.bin
```



```
nt,id,res,alpha,beta,gamma,delta,epsilon,zeta,e-z,chi,phase-angle,sugar-type,ssZp,Dp,splay,bpseq
1,g,A.GTP1,nan,nan,142.1,89.5,-131.0,-78.3,-53(BI),-178.2(anti),358.6(C2'-exo),~C3'-
endo,4.68,4.68,29.98,0      2,G,A.G2,-75.8,-167.0,57.2,79.5,-143.4,-69.7,-74(BI),-169.2(anti),5.8(C3'-
endo),~C3'-endo,4.68,4.76,25.61,0
```

```
run_x3dna(show_log=False, verbose=False)
```

```
exception rna_tools.tools.rna_x3dna.rna_x3dna.x3DNAMissingFile
```

2.5 delete a part of of your structure

Examples:

```
$ for i in *pdb; do rna_pdb_tools.py --delete A:46-56 $i > ../rpr_rm_loop/$i ; done
```

go over all files in the current directory, remove a fragment of chain A, residues between 46-56 (including them) and save outputs to in the folder *rpr_rm_loops*.

2.6 get numbering of your structure and rename chains

Rename chain B in structure 4_das_1_rpr.pdb:

```
$ rna_pdb_tools.py --get-seq 4_das_1_rpr.pdb
> 4_das_1_rpr.pdb B:1-126
GGCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAGCGUCAGCUUCGGCUGACGCUAGGCUAGUGGUGCCAAUUCUGCAG
$ rna_pdb_tools.py --edit 'B:1-126>A:1-126' 4_das_1_rpr.pdb > 4_das_1_rpr2.pdb
$ rna_pdb_tools.py --get-seq 4_das_1_rpr2.pdb
> 4_das_1_rpr2.pdb A:1-126
GGCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAGCGUCAGCUUCGGCUGACGCUAGGCUAGUGGUGCCAAUUCUGCAG
```

2.7 edit your structure (rename chain)

Examples:

```
$ rna_pdb_tools.py --edit 'A:3-21>A:1-19' 1f27_clean.pdb > 1f27_clean_A1-19.pdb
```

or even:

```
$ rna_pdb_tools.py --edit 'A:3-21>A:1-19,B:22-32>B:20-30' 1f27_clean.pdb > 1f27_clean_
↪renumb.pdb
```

or even, even, do rename X chain to A only for Chen's pdb structures in the folder, in place (so don't create a new file):

```
$ for i in *Chen*; do rna_pdb_tools.py --edit 'X:1-125>A:1-125' $i > ${i}_temp; mv ${i}_
↪temp ${i}; done
# do only edit for Chen's pdb structures, in place.
```

2.8 extract part of your structure

Example:

```
$ rna_pdb_tools.py --extract A:1-4 13_Bujnicki_1_rpr.pdb
REMARK 250 Model edited with rna-tools
REMARK 250 ver 3.1.14
REMARK 250 https://github.com/mmagnus/rna-tools
REMARK 250 Sat May 23 14:54:05 2020
HEADER extract A:1-4
ATOM      1  P      G A   1      -16.883 -12.441   8.021  1.00  0.00      P
ATOM      2  OP1    G A   1      -15.777 -12.225   8.969  1.00  0.00      O
ATOM      3  OP2    G A   1      -16.752 -11.535   6.892  1.00  0.00      O
ATOM      4  O5'    G A   1      -16.882 -13.822   7.219  1.00  0.00      O
ATOM      5  C5'    G A   1      -16.092 -13.871   6.013  1.00  0.00      C
ATOM      6  C4'    G A   1      -16.314 -15.160   5.206  1.00  0.00      C
ATOM      7  O4'    G A   1      -17.723 -14.932   4.905  1.00  0.00      O
ATOM      8  C3'    G A   1      -15.788 -15.216   3.752  1.00  0.00      C
ATOM      9  O3'    G A   1      -14.461 -15.860   3.764  1.00  0.00      O
ATOM     10  C2'    G A   1      -16.841 -15.946   2.969  1.00  0.00      C
(...)
ATOM     84  O2     U A   4      -14.553  -5.285  -7.938  1.00  0.00      O
ATOM     85  N3     U A   4      -14.077  -5.583  -5.727  1.00  0.00      N
ATOM     86  C4     U A   4      -13.451  -6.130  -4.622  1.00  0.00      C
ATOM     87  O4     U A   4      -13.706  -5.737  -3.486  1.00  0.00      O
ATOM     88  C5     U A   4      -12.494  -7.167  -4.998  1.00  0.00      C
ATOM     89  C6     U A   4      -12.318  -7.489  -6.300  1.00  0.00      C
```

2.9 find missing atoms in my structure

Run:

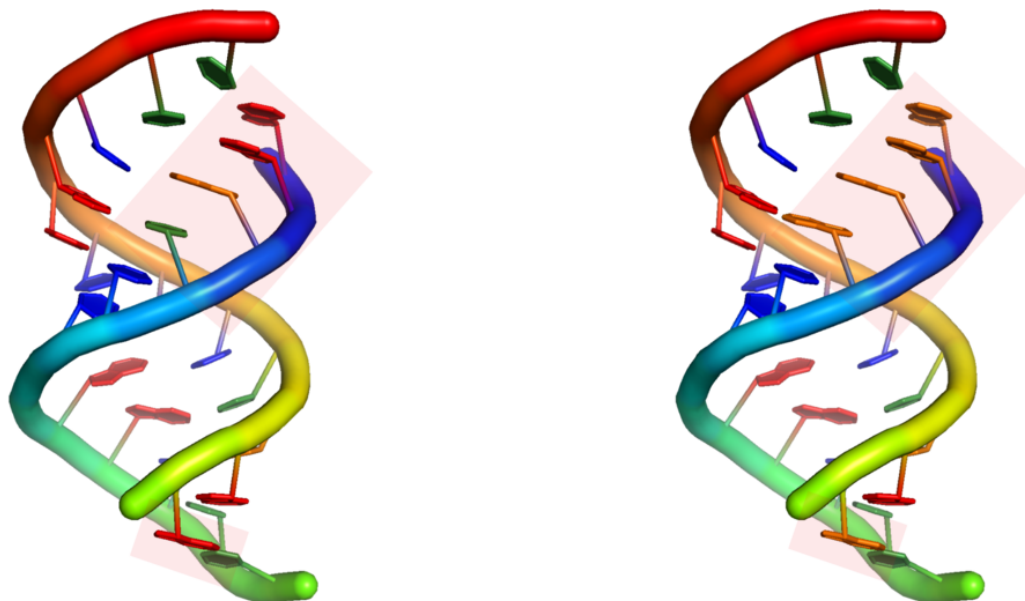
```
$ rna_pdb_tools.py --get-rnapuzzle-ready input/1_das_1_rpr_fixed.pdb
HEADER Generated with rna-pdb-tools
HEADER ver 91ed4f8-dirty
HEADER https://github.com/mmagnus/rna-pdb-tools
HEADER Sun Mar  5 10:58:07 2017
REMARK 000 Missing atoms:
REMARK 000 + P B <Residue C het= resseq=1 icode= > residue # 1
REMARK 000 + OP1 B <Residue C het= resseq=1 icode= > residue # 1
REMARK 000 + OP2 B <Residue C het= resseq=1 icode= > residue # 1
REMARK 000 + O5' B <Residue C het= resseq=1 icode= > residue # 1
ATOM      1  P      C A   1      -16.936 -3.789  68.770  1.00 11.89      P
ATOM      2  OP1    C A   1      -17.105 -3.675  67.302  1.00 14.35      O
ATOM      3  OP2    C A   1      -15.666 -4.265  69.342  1.00 12.68      O
...
```

2.10 mutate residues

For example, to replace the first four residues of chain A into adenines and 13th A of chain B, run:

```
$ rna_pdb_tools.py --mutate 'A:1A+2A+3A+4A,B:13A' \
  --inplace output/205d_rmH2o_mutA1234-B1_inplace.pdb
```

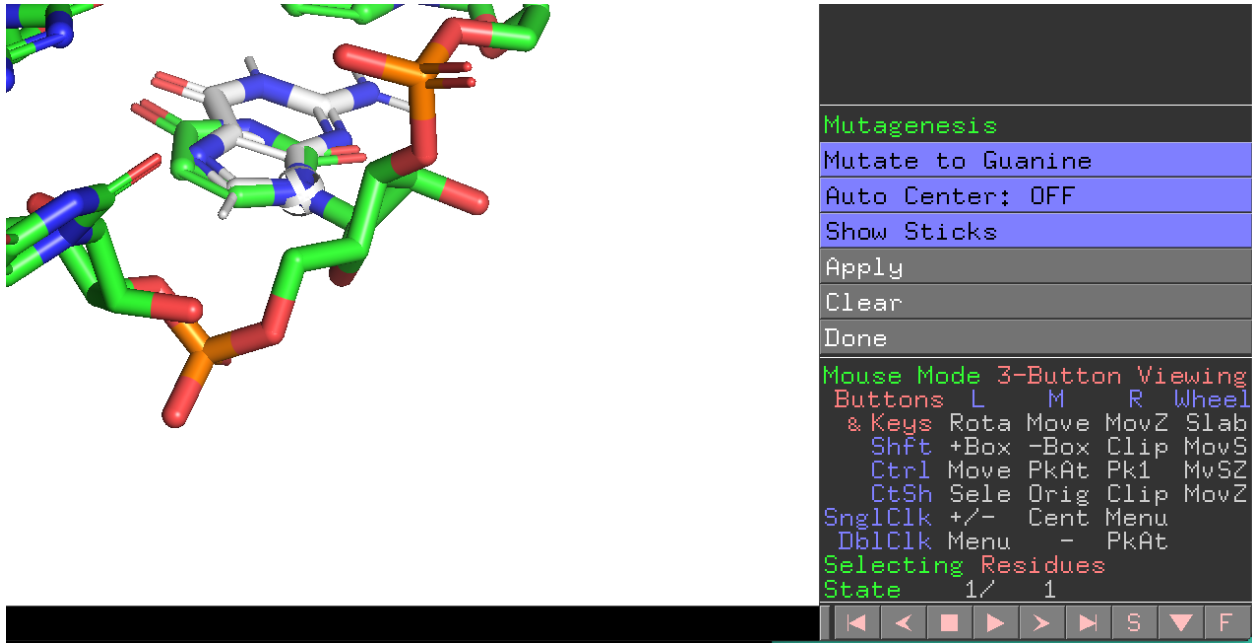
```
/205d_rmH2o//A/ 1 6 11 /B/ 16 21
GGACUUGGUCC GGAUUUGGUCC
/205d_rmH2o_mutA1234-B1_inplace//A/ 1 6 11 /B/ 16 21
AAAUUUGGUCC GAUUUGGUCC
For Educational Use Only
```



Input structure on the left, mutated structure on the right.

If, for whatever reason, the tool here does not do what you want, use the tool from MC-Fold|MC-Sym Pipeline (go there <https://www.major.irc.ca/MC-Pipeline/> and scroll down to the Section: “RNA SEQUENCE MUTATION” at the very bottom of the page).

Moreover, you can also mutate interactively proteins and nucleic acids with PyMOL >2.



Learn more here <https://pymolwiki.org/index.php/Mutagenesis>

If you want to mutate with PyMOL with command-line see this <https://pymolwiki.org/index.php/Rotkit>

2.11 add missing atoms

The tool is using the function:

```
RNAStructure.get_rnapuzzle_ready(renumber_residues=True, fix_missing_atoms=True,
                                rename_chains=True, ignore_op3=False, report_missing_atoms=True,
                                keep_hetatm=False, backbone_only=False, no_backbone=False,
                                bases_only=False, save_single_res=False, ref_frame_only=False,
                                p_only=False, check_geometry=False, conect=False,
                                conect_no_linkage=False, verbose=False)
```

Get rnapuzzle (SimRNA) ready structure.

Clean up a structure, get current order of atoms.

Parameters

- **renumber_residues** – boolean, from 1 to ..., second chain starts from 1 etc.
- **fix_missing_atoms** – boolean, superimpose motifs from the minilibrary and copy-paste missing atoms, this is super crude, so should be used with caution.

Submission format @<http://ahsoka.u-strasbg.fr/rnapuzzles/>

Run `rna_tools.rna_tools.lib.RNAStructure.std_resn()` before this function to fix names.

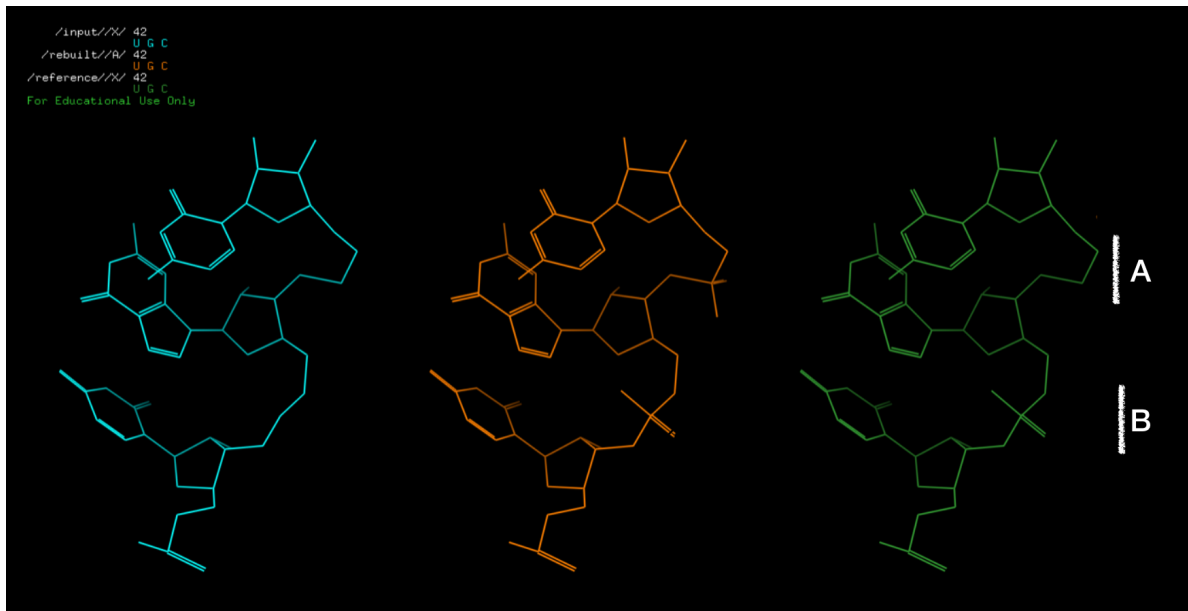


Figure: (Starting from left) input structure, structure with rebuilt atoms, and reference. The B fragment is observed in the reference used here as a “benchmark”, fragment A is reconstructed atoms (not observed in the reference”). 201122

- 170305 Merged with get_simrna_ready and fixing OP3 terminal added
- 170308 Fix missing atoms for bases, and O2'

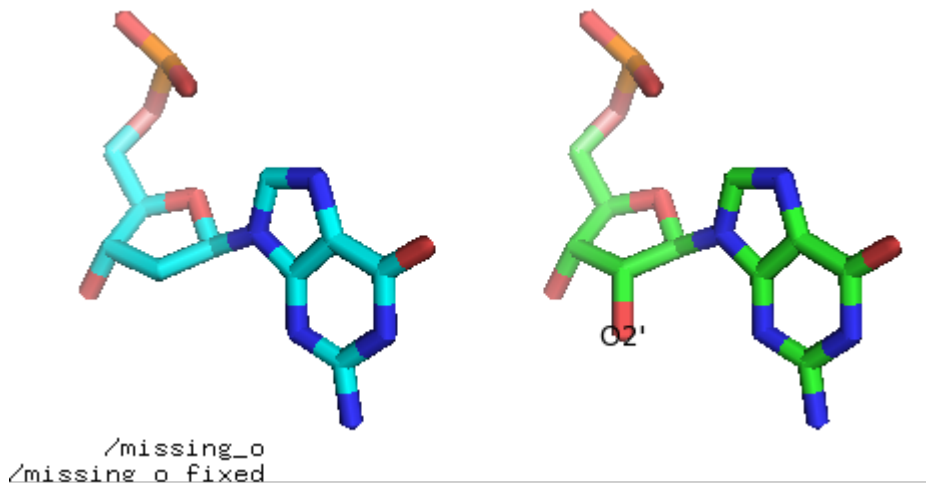


Fig. Add missing O2' atom (before and after).

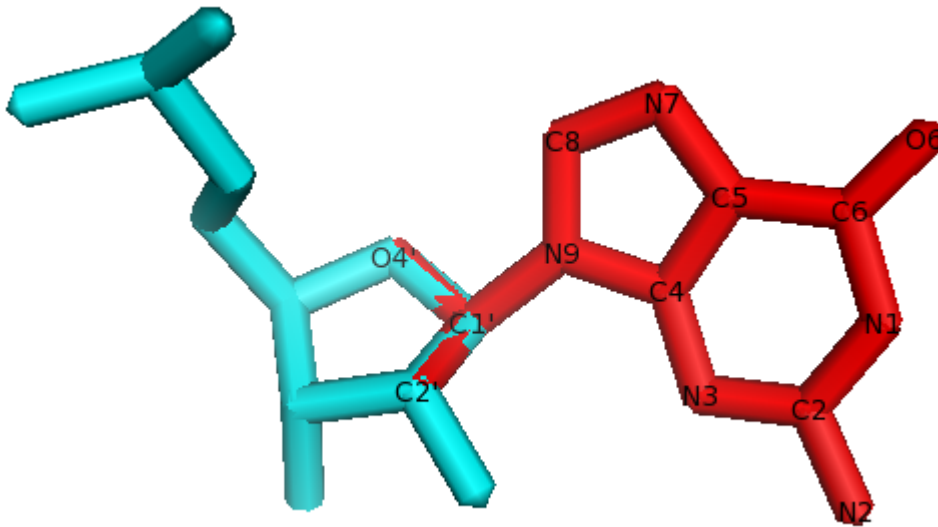
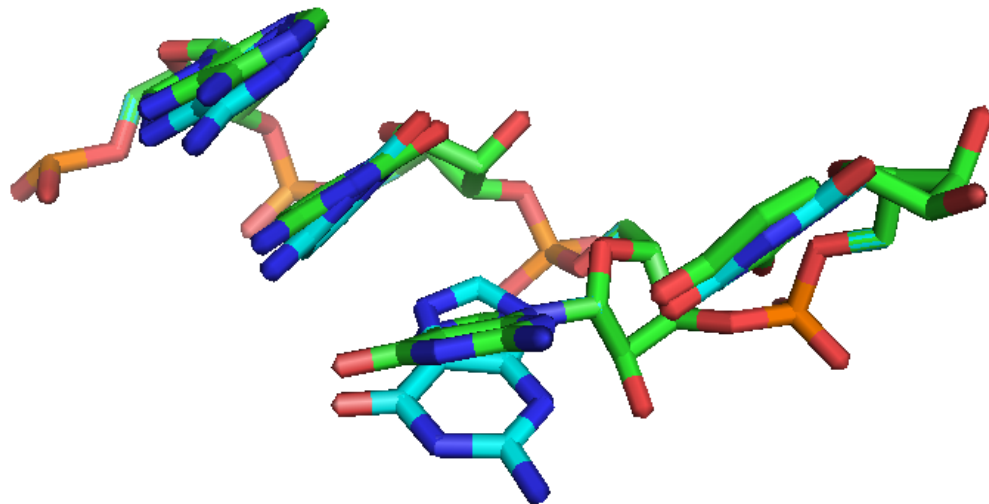


Fig. The residue to fix is in cyan. The G base from the library in red. Atoms O4', C2', C1' are shared between the sugar (in cyan) and the G base from the library (in red). These atoms are used to superimpose the G base on the sugar, and then all atoms from the base are copied to the residues.



```

/ACGU
^ACGU_no_bases_fix
/ACGU_no_bases

```

Fig. Rebuild ACGU base-less. It's not perfect but good enough for some applications.

⚠ Warning

It was only tested with the whole base missing!

⚠ Warning

requires: Biopython

Selection of atoms:

- phosphate group (3x, OP1 ,P, OP2),
- connector (2x O5', C5'), /5x
- sugar (5x, C4', O4', C3', O3', C1', C2'), /10
- extra oxygens from sugar (2x, O2' O3'), for now it's /12!
- A (10x), G (11x), C (8x), U(8x), max 12+11=23

And 27 unique atoms: {'N9', 'O2', 'OP2', "O2'", "O4'", 'C8', "O3'", "C1'", 'C2', 'C6', "C5'", 'N6', 'C5', "C4'", 'C4', "O5'", "C3'", 'O6', 'N2', 'N7', 'OP1', 'N1', 'N4', 'P', "C2'", 'N3', 'O4'}.

RNA-PDB-TOOLS

rna_pdb_tools - a swiss army knife to manipulation of RNA pdb structures

Remove atoms with XYZ equals 0:

```
(base) MiniRoseTTA git:(main) cat 4GXY_min_at0_chemicals.pdb
ATOM      1  OP1  G A  1      50.150  76.113  39.198  1.00  0.00
ATOM      2   P   G A  1      50.001  77.254  40.137  1.00  0.00
ATOM      3  OP2  G A  1      48.880  77.258  41.111  1.00  0.00
ATOM      4  O5'  G A  1      51.362  77.417  40.948  1.00  0.00
ATOM      5  C5'  G A  1          0.000   0.000   0.000  1.00  0.00
ATOM      6  C4'  G A  1          0.000   0.000   0.000  1.00  0.00
ATOM      7  O4'  G A  1          0.000   0.000   0.000  1.00  0.00
ATOM      8  C3'  G A  1          0.000   0.000   0.000  1.00  0.00
```

to get:

```
(base) MiniRoseTTA git:(main) rna_pdb_tools.py --remove0 4GXY_min_at0_chemicals.pdb
ATOM      1  OP1  G A  1      50.150  76.113  39.198  1.00  0.00
ATOM      2   P   G A  1      50.001  77.254  40.137  1.00  0.00
ATOM      3  OP2  G A  1      48.880  77.258  41.111  1.00  0.00
ATOM      4  O5'  G A  1      51.362  77.417  40.948  1.00  0.00
ATOM     35  OP1  C A  2      54.648  73.216  44.394  1.00  0.00
ATOM     36   P   C A  2      53.712  74.058  43.607  1.00  0.00
ATOM     37  OP2  C A  2      53.842  74.111  42.128  1.00  0.00
ATOM     38  O5'  C A  2      52.223  73.613  43.957  1.00  0.00
```

3.1 rna_pdb_tools

3.1.1 rna_pdb_tools.py

rna_pdb_tools - a swiss army knife to manipulation of RNA pdb structures

Usage:

```
$ rna_pdb_tools.py --delete A:46-56 --inplace *.pdb

$ rna_pdb_tools.py --get-seq *
# BujnickiLab_RNApuzzle14_n01bound
> A:1-61
```

(continues on next page)

(continued from previous page)

```
# BujnickiLab_RNApuzzle14_n02bound
> A:1-61
CGUUAGCCCAGGAAACUGGGCGGAAGUAAGGCCCAUUGCACUCCGGGCCUGAAGCAACGCG
[...]
```

See *rna_pdb_merge_into_one.py* to merge PDB files in the order as you like into one NMR-like (multimodel) file

Examples:

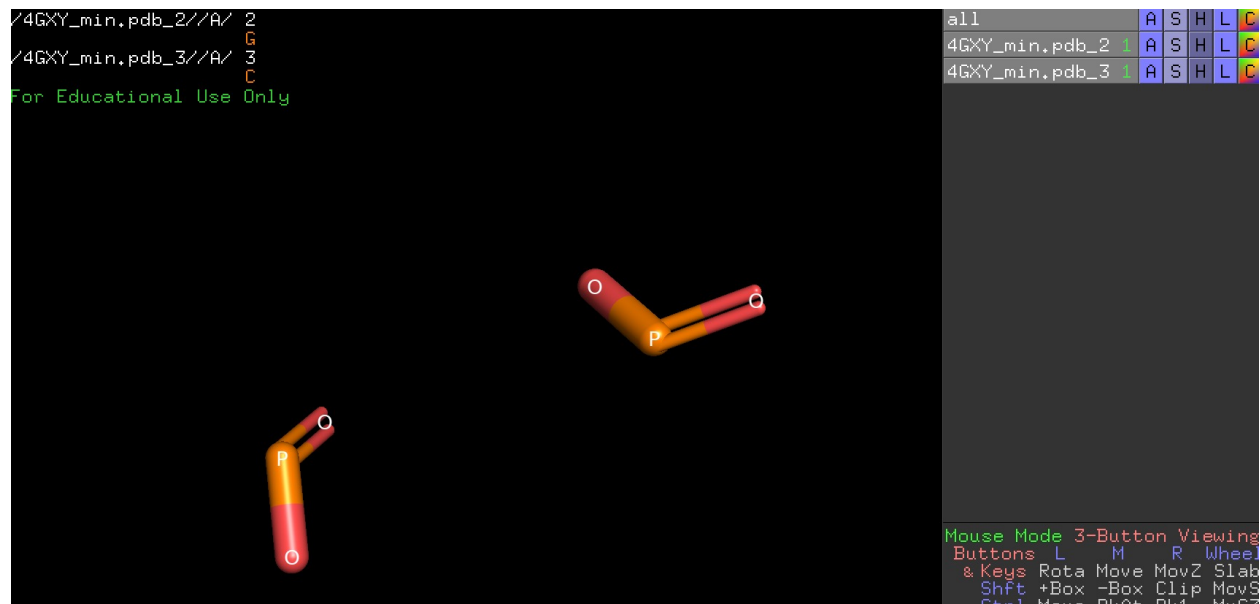
```
rna_pdb_tools.py --backbone-only --get-rnapuzzle-ready --inplace --suffix=bo examples/
↳ 4GXY_min.pdb
```

To extract specific atoms for each residue and write them to separate PDB file (next to the input files, following syntax “<input>_<resid>.pdb”):

```
rna_pdb_tools.py --rpr input/4GXY_min.pdb --save-single-res --ref-frame-only
```

Atoms presets:

```
--backbone-only      used only with --get-rnapuzzle-ready, keep only backbone (= remove
↳ bases)
--ref-frame-only     used only with --get-rnapuzzle-ready, keep only reference frames,
↳ OP1 OP2 P
--no-backbone        used only with --get-rnapuzzle-ready, remove atoms of backbone
↳ (define as P OP1 OP2 O5')
--bases-only         used only with --get-rnapuzzle-ready, keep only atoms of bases
```



-v is for verbose, -version for version ;)

```
usage: rna_pdb_tools.py [-h] [--version] [-r] [--no-progress-bar]
                        [--renum-atoms] [--renum-nmr] [--conect]
                        [--conect-no-linkage] [--renum-residues-dirty]
                        [--undo] [--delete-anisou] [--remove0] [--fix]
                        [--to-mol2] [--split-alt-locations] [-c] [--is-pdb]
```

(continues on next page)

(continued from previous page)

```

[--is-nmr] [--nmr-dir NMR_DIR] [--un-nmr] [--orgmode]
[--get-chain GET_CHAIN] [--fetch] [--fetch-cif]
[--fetch-header] [--fetch-ba] [--fetch-chain]
[--fetch-fasta] [--get-seq] [--rgyration]
[--color-seq] [--ignore-files IGNORE_FILES]
[--compact] [--hide-warnings] [--get-ss]
[--rosetta2generic] [--no-hr] [--renumber-residues]
[--dont-rename-chains] [--dont-fix-missing-atoms]
[--inspect] [--collapsed-view] [--cv] [-v]
[--mutate MUTATE] [--edit EDIT]
[--rename-chain RENAME_CHAIN]
[--swap-chains SWAP_CHAINS] [--set-chain SET_CHAIN]
[--replace-chain REPLACE_CHAIN] [--delete DELETE]
[--extract EXTRACT] [--extract-chain EXTRACT_CHAIN]
[--uniq UNIQ] [--chain-first] [--oneline]
[--replace-htm] [--fasta] [--cif2pdb] [--pdb2cif]
[--mdr] [--get-rnapuzzle-ready] [--rpr]
[--keep-hetatm] [--inplace] [--here] [--suffix SUFFIX]
[--replace-hetatm] [--dont-report-missing-atoms]
[--backbone-only] [--p-only] [--ref-frame-only]
[--no-backbone] [--bases-only] [--save-single-res]
file [file ...]

```

file

file

-h, --help

show this help message and exit

--version**-r, --report**

get report

--no-progress-bar

for --no-progress-bar for -rpr

--renum-atoms

renumber atoms, tested with -get-seq

--renum-nmr**--conect**

add conect to rpr file

--conect-no-linkage

dont add conect from our residue to another

--renum-residues-dirty**--undo**

undo operation of action done -inplace, , rename "backup files" .pdb~ to pdb, ALL files in the folder, not only ~ related to the last action (that you might want to revert, so be careful)

--delete-anisou

remove files with ANISOU records, works with `-inplace`

--remove0

remove atoms of X=0 position

--fix

fix a PDB file, ! external program, pdbfixer used to fix missing atoms

--to-mol2

fix a PDB file, ! external program, pdbfixer used to fix missing atoms

--split-alt-locations

splits atoms, e.g. for alt locs A and B, it splits atoms two MODELS (all localizations A goes into MODEL1 and all localizations B goes into MODEL2)

-c, --clean

get clean structure

--is-pdb

check if a file is in the pdb format

--is-nmr

check if a file is NMR-style multiple model pdb

--nmr-dir <nmr_dir>

make NMR-style multiple model pdb file from a set of files

```
rna_pdb_tools.py --nmr-dir . 'cwc15_u5_fragments*.pdb' > ~/Desktop/cwc15-u5.pdb
```

please use `'` for pattern file recognition, this is a hack to deal with folders with thousands of models, if you used only `*`.pdb then the terminal will complain that you selected too many files.

--un-nmr

split NMR-style multiple model pdb files into individual models [biopython],

```
rna_pdb_tools.py --un-nmr split.pdb 2 /Users/magnus/Desktop/3hl2/split_1.pdb  
/Users/magnus/Desktop/3hl2/split_2.pdb
```

--orgmode

get a structure in org-mode format `<sick!>`

--get-chain <get_chain>

get chain, one or many, e.g. A, but now also ABC works

--fetch

fetch file from the PDB db, e.g., 1xjr, use `'rp'` to fetch, fetch a given join, 4w90:C or 4w90_Cthe RNA-Puzzles standardized_dataset [around 100 MB]

--fetch-cif

--fetch-header

--fetch-ba

fetch biological assembly from the PDB db

--fetch-chain

fetch a structure in extract chain, e.g. 6bk8 H

--fetch-fasta

fetch a fasta/sequence for given PDB ID, e.g. 6bk8

--get-seq

get seq

--rgyration

get seq

--color-seq

color seq, works with `--get-seq`

--ignore-files <ignore_files>

files to be ingored, .e.g, 'solution'

--compact

with `--get-seq`, get it in compact view' \$ rna_pdb_tools.py `--get-seq --compact *.pdb # 20_Bujnicki_1` ACCCG-
CAAGGCCGACGGCGCCGCCGUGGUGCAAGUCCAGCCACGCUUCGGCGUGGGCGCUCAUGGGU
A:1-68 # 20_Bujnicki_2 ACCCGCAAGGCCGACGGCGCCGCCGUGGUGCAAGUCCAGCCACGCU-
UCGGCGUGGGCGCUCAUGGGU # A:1-68 # 20_Bujnicki_3 ACCCGCAAGGCCGACGGCGCCGC-
CGCUGGUGCAAGUCCAGCCACGCUUCGGCGUGGGCGCUCAUGGGU # A:1-68 # 20_Bujnicki_4

--hide-warnings

hide warnings, works with `--get-chain`, it hides warnings that given changes are not detected in a PDB file

--get-ss

get secondary structure

--rosetta2generic

convert ROSETTA-like format to a generic pdb

--no-hr

do not insert the header into files

--renumber-residues

by default is false

--dont-rename-chains

used only with `--get-rnapuzzle-ready`. By default: `--get-rnapuzzle-ready` rename chains from ABC.. to stop
behavior switch on this option

--dont-fix-missing-atoms

used only with `--get-rnapuzzle-ready`

--inspect

inspect missing atoms (technically decorator to `--get-rnapuzzle-ready` without actually doing anything but giving
a report on problems)

--collapsed-view**--cv**

alias to `collapsed_view`

-v, --verbose

tell me more what you're doing, please!

--mutate <mutate>

mutate residues, e.g., `--mutate "A:1a+2a+3a+4a,B:1a"` to mutate to adenines the first four nucleotides of the chain A and the first nucleotide of the chain B

--edit <edit>

edit `'A:6>B:200'`, `'A:2-7>B:2-7'`

--rename-chain <rename_chain>

edit `'A>B'` to rename chain A to chain B

--swap-chains <swap_chains>

B>A, rename A to `_`, then B to A, then `_` to B

--set-chain <set_chain>

set chain for all ATOM lines and TER (quite brutal function)

--replace-chain <replace_chain>

a file PDB name with one chain that will be used to replace the chain in the original PDB file, the chain id in this file has to be the same with the chain id of the original chain

--delete <delete>

delete the selected fragment, e.g. `A:10-16`, or for more than one fragment `--delete 'A:1-25+30-57'`

--extract <extract>

extract the selected fragment, e.g. `A:10-16`, or for more than one fragment `--extract 'A:1-25+30-57'`, or even `'A:1-25+B:30-57'`

--extract-chain <extract_chain>

extract chain, e.g. A

--uniq <uniq>

```
rna_pdb_tools.py --get-seq --uniq '[:5]' --compact --chain-first * | sort A:1-121 ACCUUGCGCAACUGGC-
GAAUCCUGGGGUGCCGCCGCGCAGUACCC...CA # rp13nc3295_min.out.1 A:1-123 ACCUUGCGC-
GACUGGGCGAAUCCUGAAGCUGCUUUGAGCGGCUUCG...AG # rp13cp0016_min.out.1 A:1-123
ACCUUGCGCGACUGGGCGAAUCCUGAAGCUGCUUUGAGCGGCUUCG...AG # zcp_6537608a_ALL-
000001_AA A:1-45 57-71 GGGUCGUGACUGGGCGAACAGGUGGGAAACCACCGGGGAGCGACCCGC-
CGCCCGCCUGGGC # solution
```

--chain-first

--oneline

--replace-htm

--fasta

with `--get-seq`, show sequences in fasta format, can be combined with `--compact` (mind, chains will be separated with `' '` in one line)

```
$ rna_pdb_tools.py --get-seq --fasta --compact input/20_Bujnicki_1.pdb > 20_Bujnicki_1 ACCCGCAAGGCC-
GACGGC GCCGCCGUGGUGCAAGUCCAGCCACGCUUCGGCGUGGGCGCUCAUGGGU
```

--cif2pdb

convert cif to PDB, fancy way

--pdb2cif

[PyMOL Python package required]

-
- mdr**
get structures ready for MD (like rpr but without first)
 - get-rnapuzzle-ready**
get RNApuzzle ready (keep only standard atoms). Be default it does not renumber residues, use `--renumber-residues` [requires BioPython]
 - rpr**
alias to `get_rnapuzzle ready`)
 - keep-hetatm**
keep hetatoms
 - inplace**
in place edit the file! [experimental, only for `get_rnapuzzle_ready`, `--delete`, `--get-ss`, `--get-seq`, `--edit-pdb`]
 - here**
save a file next to the original file with auto suffix for `--extract` it's `.extr.pdb`
 - suffix <suffix>**
when used with `--inplace` allows you to change a name of a new file, `--suffix del` will give `<file>_del.pdb` (mind added `_`)
 - replace-hetatm**
replace 'HETATM' with 'ATOM' [tested only with `--get-rnapuzzle-ready`]
 - dont-report-missing-atoms**
used only with `--get-rnapuzzle-ready`
 - backbone-only**
used only with `--get-rnapuzzle-ready`, keep only backbone (= remove bases)
 - p-only**
used only with `--get-rnapuzzle-ready`, keep p backbone (= remove bases)
 - ref-frame-only**
used only with `--get-rnapuzzle-ready`, keep only reference frames, OP1 OP2 P
 - no-backbone**
used only with `--get-rnapuzzle-ready`, remove atoms of backbone (define as P OP1 OP2 O5')
 - bases-only**
used only with `--get-rnapuzzle-ready`, keep only atoms of bases
 - save-single-res**
used only with `--get-rnapuzzle-ready`, for each residue create a new pdb output file, you can combine it with `--bases-only` etc.

3.1.2 get RNAPuzzle ready

class `rna_tools.rna_tools_lib.RNAStructure(fn=")`

RNAStructure - handles an RNA pdb file.

fn

path to the structural file, e.g., “./rna_tools/input/4ts2.pdb”

Type

string

name

filename of the structural file, “4ts2.pdb”

Type

string

lines

the PDB file is loaded and ATOM/HETATM/TER/END go to self.lines

Type

list

get_rnapuzzle_ready(*renumber_residues=True, fix_missing_atoms=True, rename_chains=True, ignore_op3=False, report_missing_atoms=True, keep_hetatm=False, backbone_only=False, no_backbone=False, bases_only=False, save_single_res=False, ref_frame_only=False, p_only=False, check_geometry=False, connect=False, connect_no_linkage=False, verbose=False*)

Get rnapuzzle (SimRNA) ready structure.

Clean up a structure, get current order of atoms.

Parameters

- **renumber_residues** – boolean, from 1 to ..., second chain starts from 1 etc.
- **fix_missing_atoms** – boolean, superimpose motifs from the minilibrary and copy-paste missing atoms, this is super crude, so should be used with caution.

Submission format @<http://ahsoka.u-strasbg.fr/rnapuzzles/>

Run `rna_tools.rna_tools_lib.RNAStructure.std_resn()` before this function to fix names.

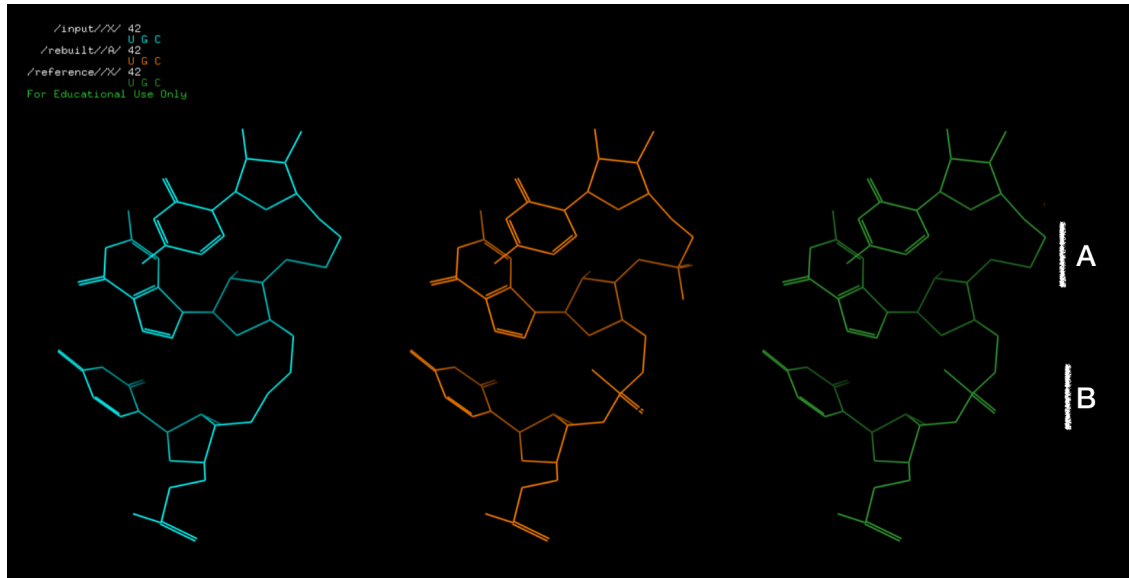


Figure: (Starting from left) input structure, structure with rebuilt atoms, and reference. The B fragment is observed in the reference used here as a “benchmark”, fragment A is reconstructed atoms (not observed in the reference”). 201122

- 170305 Merged with get_simrna_ready and fixing OP3 terminal added
- 170308 Fix missing atoms for bases, and O2'

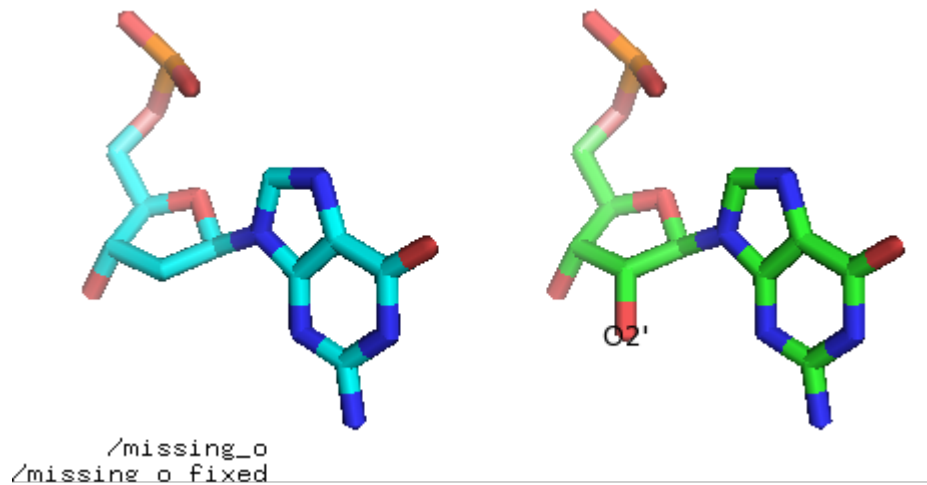


Fig. Add missing O2' atom (before and after).

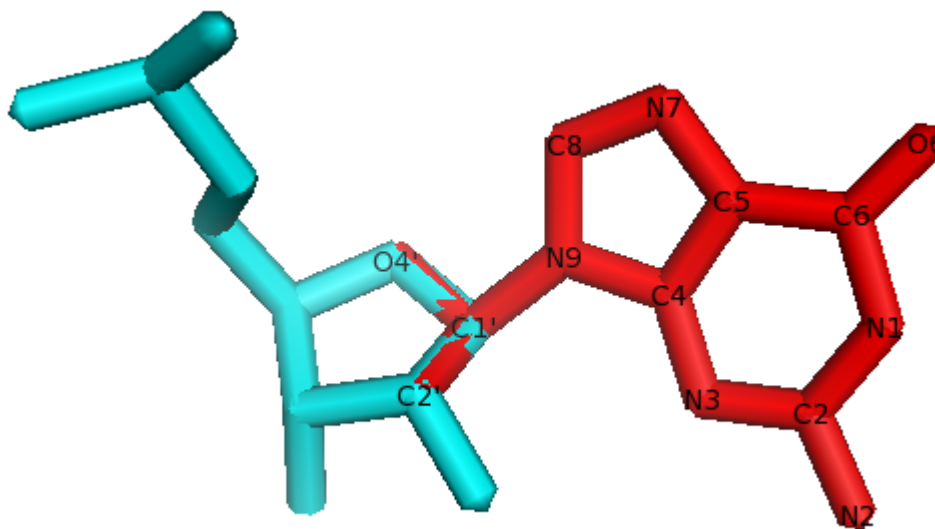
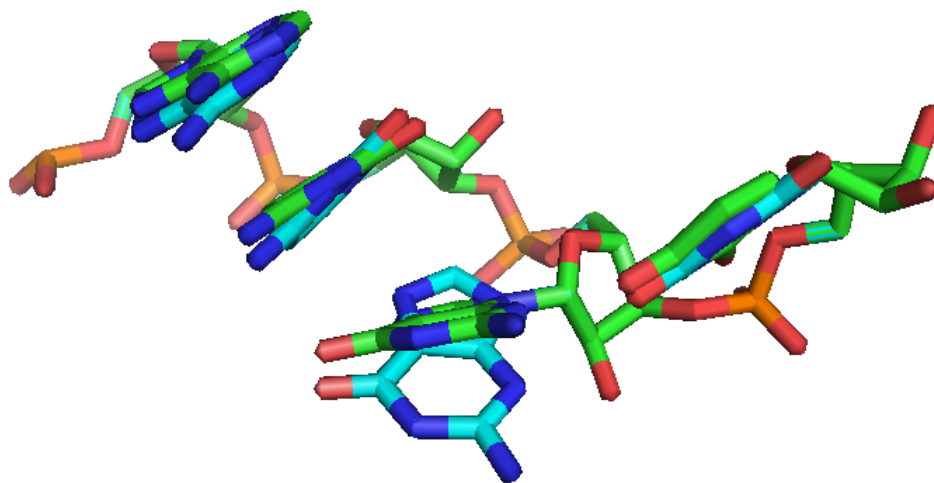


Fig. The residue to fix is in cyan. The G base from the library in red. Atoms O4', C2', C1' are shared between the sugar (in cyan) and the G base from the library (in red). These atoms are used to superimpose the G base on the sugar, and then all atoms from the base are copied to the residues.



```

/ACGU
'ACGU_no_bases_fix
/ACGU_no_bases
    
```

Fig. Rebuild ACGU base-less. It's not perfect but good enough for some applications.

Warning

It was only tested with the whole base missing!

Warning

requires: Biopython

Selection of atoms:

- phosphate group (3x, OP1 ,P, OP2),
- connector (2x O5', C5'), /5x
- sugar (5x, C4', O4', C3', O3', C1', C2'), /10
- extra oxygens from sugar (2x, O2' O3'), for now it's /12!
- A (10x), G (11x), C (8x), U(8x), max 12+11=23

And 27 unique atoms: {'N9', 'O2', 'OP2', "O2'", "O4'", 'C8', "O3'", "C1'", 'C2', 'C6', "C5'", 'N6', 'C5', "C4'", 'C4', "O5'", "C3'", 'O6', 'N2', 'N7', 'OP1', 'N1', 'N4', 'P', "C2'", 'N3', 'O4'}.

3.1.3 get sequence

Example:

```
$ rna_pdb_tools.py --get-seq 5_solution_1.pdb
> 5_solution_1.pdb A:1-576
CAUCCGGUAUCCCAAGACAAUCUCGGGUUGGGUUGGGGAAGUAUCAUGGCUAAUCACCAUGAUGCAAUCGGGUUGAACACUUAUUUGGGUUAAAACGGUGGGG
```

class rna_tools.rna_tools_lib.RNAstructure(*fn=""*)

RNAstructure - handles an RNA pdb file.

fn

path to the structural file, e.g., "../rna_tools/input/4ts2.pdb"

Type

string

name

filename of the structural file, "4ts2.pdb"

Type

string

lines

the PDB file is loaded and ATOM/HETATM/TER/END go to self.lines

Type

list

get_seq(*compact=False, chainfirst=False, fasta=False, addfn="", color=False*)

Get seq (v2) gets segments of chains with correct numbering

Run:

```
python rna_pdb_seq.py input/1ykq_clx.pdb
> 1ykq_clx A:101-111
GGAGCUCGCC
> 1ykq_clx B:201-238
GGGCGAGGCCGUGCCAGCUCUUCGGAGCAAUACUCGGC

> 6_solution_0 A:1-19 26-113 117-172
GGCGGCAGGUGCUCGCCGACGUCGGGAGUUAAAAGGGAAG
```

Chains is {'A': {'header': 'A:1-19 26-113 117-172', 'resi': [1, 2, 3, ..., 19, 26, 27, ..., 172], 'seq': ['G', 'G', 'C', 'G', ... C', 'G', 'U', 'C']}}

Chains are in other as the appear in the file.

 **Warning**

take only ATOM and HETATM lines.

3.1.4 fetch

Example:

```
$ rna_pdb_tools.py --fetch 1xjr
downloading...1xjr ok
```

`rna_tools.rna_tools_lib.fetch(pdb_id, path='.')`
fetch pdb file from RCSB.org <https://files.rcsb.org/download/1Y26.pdb>
Args: - pdb_id, but also a chain can be specified, 1jj2:A+B+C
Returns: - a path to a file
TODO: na_pdb_tools.py -extract A:1-25+B:30-57 1jj2.pdb

3.1.5 fetch Biological Assembly

Example:

```
$ rna_pdb_tools.py --fetch-ba 1xjr
downloading...1xjr_ba.pdb ok
```

or over a list of pdb ids in a text file:

```
$ cat data/pdb_ids.txt
1y26
1fir

$ while read p; do rna_pdb_tools.py --fetch-ba $p; done < data/pdb_ids.txt
downloading...1y26_ba.pdb ok
downloading...1fir_ba.pdb ok

$ ls *.pdb
1fir_ba.pdb 1y26_ba.pdb
```

`rna_tools.rna_tools_lib.fetch_ba(pdb_id, path='.')`
fetch biological assembly pdb file from RCSB.org

```
>>> fetch_ba('1xjr')
...
```

3.1.6 delete

Examples:

```
$ for i in *pdb; do rna_pdb_tools.py --delete A:46-56 $i > ../rpr_rm_loop/$i ; done
```

go over all files in the current directory, remove a fragment of chain A, residues between 46-56 (including them) and save outputs to in the folder *rpr_rm_loops*.

3.1.7 edit

`rna_tools.rna_tools_lib.edit_pdb(f, args)`

Edit your structure.

The function can take `A:3-21>A:1-19` or even syntax like this `A:3-21>A:1-19,B:22-32>B:20-30` and will do an editing.

The output is printed, line by line. Only ATOM lines are edited!

Examples:

```
$ rna_pdb_tools.py --edit 'A:3-21>A:1-19' 1f27_clean.pdb > 1f27_clean_A1-19.pdb
```

or even:

```
$ rna_pdb_tools.py --edit 'A:3-21>A:1-19,B:22-32>B:20-30' 1f27_clean.pdb > 1f27_
↪clean_renumb.pdb
```

3.1.8 RNAStructure (rna_tools_lib)

`rna_tools_lib.py` - main lib file, many tools in this lib is using this file.

exception `rna_tools.rna_tools_lib.MethodUnknown`

exception `rna_tools.rna_tools_lib.PDBFetchError`

class `rna_tools.rna_tools_lib.RNAStructure(fn="")`

RNAStructure - handles an RNA pdb file.

fn

path to the structural file, e.g., `"../rna_tools/input/4ts2.pdb"`

Type

string

name

filename of the structural file, `"4ts2.pdb"`

Type

string

lines

the PDB file is loaded and ATOM/HETATM/TER/END go to self.lines

Type

list

check_geometry(*verbose=False*)

Check for correct “Polymer linkage, it should be around 1.6Å with a sigma around 0.01.

Carrascoza, F., Antczak, M., Miao, Z., Westhof, E. & Szachniuk, M. Evaluation of the stereochemical quality of predicted RNA 3D models in the RNA-Puzzles submissions. *Rna* 28, 250–262 (2022).

Values for 1xjr.pdb:

```
op.mean(): 1.599316
op.std(): 0.009274074

po.mean(): 1.5984017
po.std(): 0.0069191623
```

requires biopython

edit_occupancy_of_pdb(*pdb, pdb_out, v=False*)

Make all atoms 1 (flexi) and then set occupancy 0 for selected atoms. Return False if error. True if OK

fix_0_in_UC()**fix_op_atoms()**

Replace OXP to OPX1, e.g ('O1P' -> 'OP1')

fix_with_qrnas(*outfn="", verbose=False*)

Add missing heavy atom.

A residue is recognized base on a residue names.

Copy QRNAS folder to curr folder, run QRNAS and remove QRNAS.

Warning

QRNAS required (<http://genesilico.pl/QRNAS/QRNAS.tgz>)

get_all_chain_ids()**Returns**

chain ids, e.g. set(['A', 'B'])

Return type

set

get_atom_code(*line*)

Get atom code from a line of a PDB file

get_atom_coords(*line*)

Get atom coordinates from a line of a PDB file

get_atom_num(*line*)

Extract atom number from a line of PDB file :param * line = ATOM line from a PDB file:

Output:

- atom number (int)

get_info_chains()

return A:3-21 B:22-32

get_remarks_text()

Get remarks as text for given file. This function re-open files, as define as self.fn to get remarks.

Example:

```
r = RNAStructure(fout)
remarks = r.get_remarks_txt()
r1 = r.get_res_txt('A', 1)
r2 = r.get_res_txt('A', 2)
r3 = r.get_res_txt('A', 3)
with open(fout, 'w') as f:
    f.write(remarks)
    f.write(r1)
    f.write(r2)
    f.write(r3)

remarks is

REMARK 250 Model edited with rna-tools
REMARK 250 ver 3.5.4+63.g4338516.dirty
REMARK 250 https://github.com/mmagnus/rna-tools
REMARK 250 Fri Nov 13 10:15:19 2020
```

get_report()**Returns**

report, messages collected on the way of parsing this file

Return type

string

get_res_code(line)

Get residue code from a line of a PDB file

get_res_num(line)

Extract residue number from a line of PDB file :param * line = ATOM line from a PDB file:

Output:

- residue number as an integer

get_res_text(chain_id, resi)

Get a residue of given resi of chain_id and return as a text

Parameters

- **chain_id** (*str*) – e.g., 'A'
- **resi** (*int*) – e.g., 1

Return type

txt

Example:

```
r = RNAStructure(fn)
print(r.get_res_txt('A', 1))

ATOM      1  O5'   G A  1      78.080 -14.909  -0.104  1.00  9.24  ↵
↪      0
ATOM      2  C5'   G A  1      79.070 -15.499  -0.956  1.00  9.70  ↵
↪      C
ATOM      3  C4'   G A  1      78.597 -16.765  -1.648  1.00  9.64  ↵
```

(continues on next page)

(continued from previous page)

```

→ C
ATOM      4  O4'   G A  1      78.180 -17.761  -0.672  1.00  9.88
→ 0
(...)
```

get_rnapuzzle_ready(*renumber_residues=True, fix_missing_atoms=True, rename_chains=True, ignore_op3=False, report_missing_atoms=True, keep_hetatm=False, backbone_only=False, no_backbone=False, bases_only=False, save_single_res=False, ref_frame_only=False, p_only=False, check_geometry=False, conect=False, conect_no_linkage=False, verbose=False*)

Get rnapuzzle (SimRNA) ready structure.

Clean up a structure, get current order of atoms.

Parameters

- **renumber_residues** – boolean, from 1 to ..., second chain starts from 1 etc.
- **fix_missing_atoms** – boolean, superimpose motifs from the minilibrary and copy-paste missing atoms, this is super crude, so should be used with caution.

Submission format @<http://ahsoka.u-strasbg.fr/rnapuzzles/>

Run `rna_tools.rna_tools.lib.RNAstructure.std_resn()` before this function to fix names.

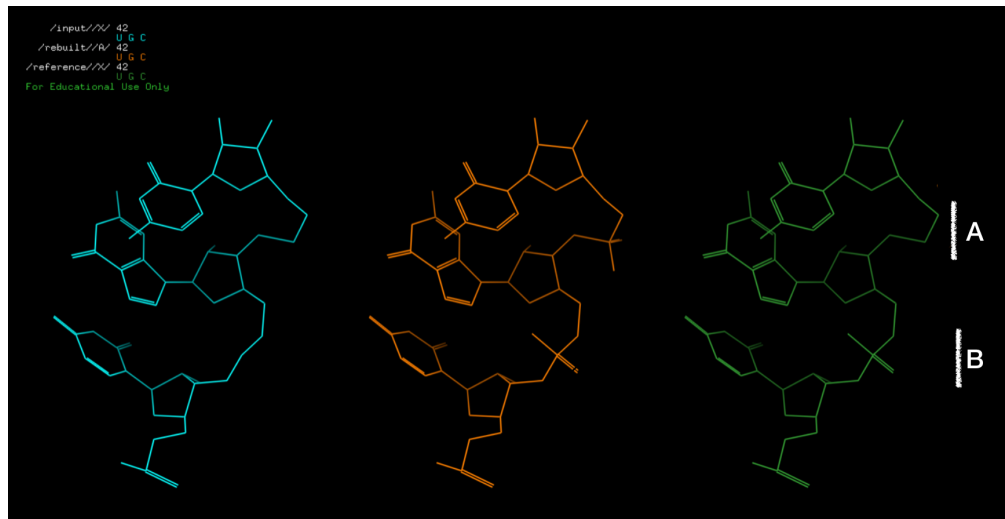


Figure: (Starting from left) input structure, structure with rebuilt atoms, and reference. The B fragment is observed in the reference used here as a “benchmark”, fragment A is reconstructed atoms (not observed in the reference”). 201122

- 170305 Merged with `get_simrna_ready` and fixing OP3 terminal added
- 170308 Fix missing atoms for bases, and O2'

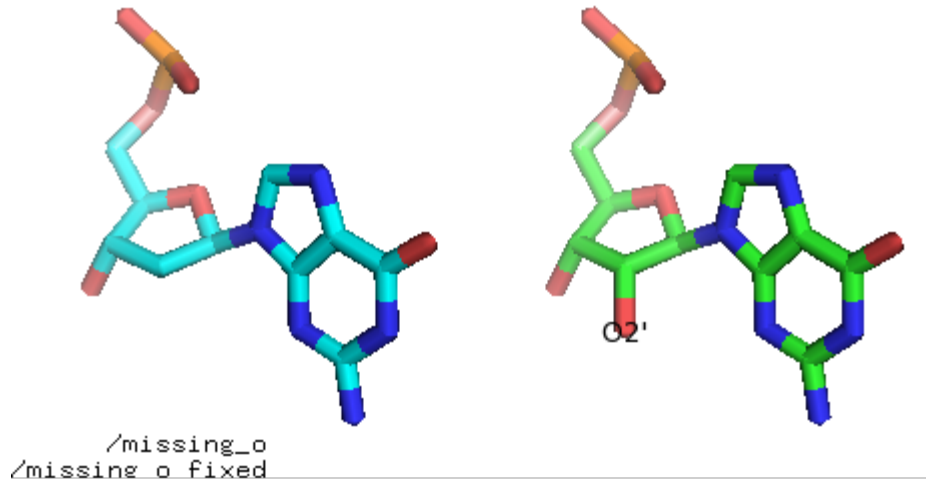


Fig. Add missing O2' atom (before and after).

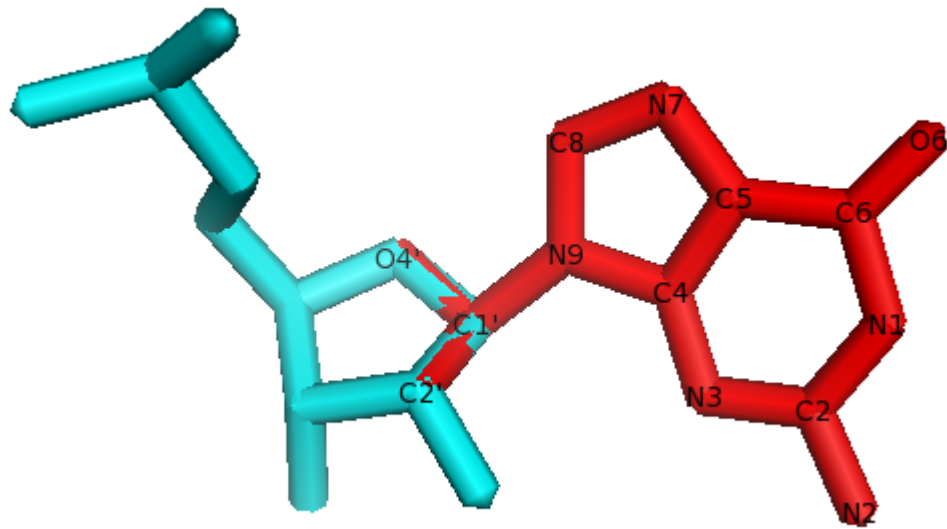
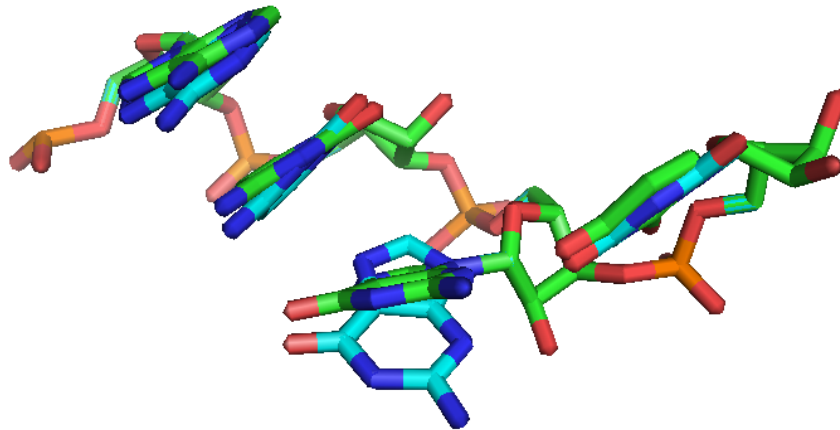


Fig. The residue to fix is in cyan. The G base from the library is in red. Atoms O4', C2', C1' are shared between the sugar (in cyan) and the G base from the library (in red). These atoms are used to superimpose the G base on the sugar, and then all atoms from the base are copied to the residues.



```

/ACGU
'ACGU_no_bases_fix
/ACGU_no_bases

```

Fig. Rebuild ACGU base-less. It's not perfect but good enough for some applications.

Warning

It was only tested with the whole base missing!

Warning

requires: Biopython

Selection of atoms:

- phosphate group (3x, OP1 ,P, OP2),
- connector (2x O5', C5'), /5x
- sugar (5x, C4', O4', C3', O3', C1', C2'), /10
- extra oxygens from sugar (2x, O2' O3'), for now it's /12!
- A (10x), G (11x), C (8x), U(8x), max 12+11=23

And 27 unique atoms: { 'N9', 'O2', 'OP2', "O2'", "O4'", 'C8', "O3'", "C1'", 'C2', 'C6', "C5'", 'N6', 'C5', "C4'", 'C4', "O5'", "C3'", 'O6', 'N2', 'N7', 'OP1', 'N1', 'N4', 'P', "C2'", 'N3', 'O4' }.

get_seq(compact=False, chainfirst=False, fasta=False, addfn="", color=False)

Get seq (v2) gets segments of chains with correct numbering

Run:

```

python rna_pdb_seq.py input/1ykq_c1x.pdb
> 1ykq_c1x A:101-111
GGAGCUCGCC
> 1ykq_c1x B:201-238
GGGCGAGGCCGUGCCAGCUCUUCGGAGCAAUACUCGGC
> 6_solution_0 A:1-19 26-113 117-172
GGCGCAGGUGCUCCGACGUCGGAGUAAAAGGGAAG

```

Chains is { 'A': { 'header': 'A:1-19 26-113 117-172', 'resi': [1, 2, 3,

```
..., 19, 26, 27, ..., 172], 'seq': ['G', 'G', 'C', 'G', ... C', 'G',
'U', 'C']}]}
```

Chains are in other as the appear in the file.

Warning

take only ATOM and HETATM lines.

get_text(*add_end=True*)

works on self.lines.

is_amber_like()

Use self.lines and check if there is XX line

is_mol2()

Return True if is_mol2 based on the presence of `@<TRIPOS>`.

is_nmr()

True if the file is an NMR-style multiple model pdb

Returns

True or Fo

Return type

boolean

is_pdb()

Return True if the files is in PDB format.

If self.lines is empty it means that nothing was parsed into the PDB format.

reload()

Reload the object.

remove(*verbose*)

Delete file, self.fn

remove_ion()

```
TER 1025 U A 47 HETATM 1026 MG MG A 101 42.664 34.395 50.249 1.00 70.99 MG HET-
ATM 1027 MG MG A 201 47.865 33.919 48.090 1.00 67.09 MG
```

rtype

object

remove_water()

Remove HOH and TIP3

rename_chain(*chain_id_old, chain_id_new, debug=False*)

Rename chains

Parameters

- **chain_id_old** (*str*) – e.g., A
- **chain_id_new** (*str*) – e.g., B
- **debug** (*bool*) – show some diagnostics

Returns

pdb content (txt) self.lines is updated with new lines

renum_atoms()

Renum atoms, from 1 to X for line; ATOM/HETATM

set_atom_code(*line, code*)

Add atom name/code:

```
ATOM 1 OP2 C A 1 29.615 36.892 42.657 1.00 1.00 O
  ^^^ ^ and element
```

set_atom_coords(*line, x, y, z*)

Get atom coordinates from a line of a PDB file

set_atom_occupancy(*line, occupancy*)

set occupancy for line

set_occupancy_atoms(*occupancy*)

Parameters
occupancy

set_res_code(*line, code*)

Parameters
• lines
• code

path (str): The path of the file to wrap field_storage (FileStorage): The :class:Y instance to wrap temporary (bool): Whether or not to delete the file when the File instance is destructed

Returns

A buffered writable file descriptor

Return type

BufferedFileStorage

std_resn()

'Fix' residue names which means to change them to standard, e.g. RA5 -> A

Works on self.lines, and returns the result to self.lines.

Will change things like:

```
# URI -> U, URA -> U
1xjr_clx_charmm.pdb:ATOM 101 P URA A 5 58.180 39.153 30.
↪ 336 1.00 70.94
rp13_Dokholyan_1_URI_CYT_ADE_GUA_hydrogens.pdb:ATOM 82 P URI A 4
↪ 501.633 506.561 506.256 1.00 0.00 P
```

un_nmr(*startwith1=True, directory="", verbose=False*)

Un NMR - Split NMR-style multiple model pdb files into individual models.

Take self.fn and create new files in the way:

```
input/1a9l_NMR_1_2_models.pdb
input/1a9l_NMR_1_2_models_0.pdb
input/1a9l_NMR_1_2_models_1.pdb
```

Warning

This function requires biopython.

```
rna_pdb_tools.py -un-nmr AF-Q5TCX8-F1-model_v1_core_Ctrim_mdr_MD.pdb
```

36

```
cat MD_ > md.pdb
```

write(*outfn=""*, *verbose=False*)

Write `self.lines` to a file (and add END file)

Parameters

- **outfn** (*str*) – file to save, if outfn is "", then simply use self.fn
- **verbose** (*Boolean*) – be verbose or not

Returns

None

`rna_tools.rna_tools_lib.aa3to1(aaa)`

based on https://pymolwiki.org/index.php/Aa_codes

`rna_tools.rna_tools_lib.collapsed_view(args)`

Collapsed view of pdb file. Only lines with C5' atoms are shown and TER, MODEL, END.

example:

```
[mm] rna_tools git:(master) $ python rna-pdb-tools.py --cv input/1f27.pdb
ATOM      1  C5'   A  A   3      25.674  19.091   3.459  1.00  16.99
↪ C
ATOM     23  C5'   C  A   4      19.700  19.206   5.034  1.00  12.65
↪ C
ATOM     43  C5'   C  A   5      14.537  16.130   6.444  1.00   8.74
↪ C
ATOM     63  C5'   G  A   6      11.726  11.579   9.544  1.00   9.81
↪ C
ATOM     86  C5'   U  A   7      12.007   7.281  13.726  1.00  11.35
↪ C
ATOM    106  C5'   C  A   8      12.087   6.601  18.999  1.00  12.74
↪ C
TER
```

`rna_tools.rna_tools_lib.edit_pdb(f, args)`

Edit your structure.

The function can take A:3-21>A:1-19 or even syntax like this A:3-21>A:1-19, B:22-32>B:20-30 and will do an editing.

The output is printed, line by line. Only ATOM lines are edited!

Examples:

```
$ rna_pdb_tools.py --edit 'A:3-21>A:1-19' 1f27_clean.pdb > 1f27_clean_A1-
↪19.pdb
```

or even:

```
$ rna_pdb_tools.py --edit 'A:3-21>A:1-19,B:22-32>B:20-30' 1f27_clean.pdb >
↪1f27_clean_renumb.pdb
```

`rna_tools.rna_tools_lib.fetch(pdb_id, path='.')`

fetch pdb file from RCSB.org <https://files.rcsb.org/download/1Y26.pdb>

Args: - *pdb_id*, but also a chain can be specified, 1jj2:A+B+C

Returns: - a path to a file

TODO: `na_pdb_tools.py -extract A:1-25+B:30-57 1jj2.pdb`

`rna_tools.rna_tools_lib.fetch_ba(pdb_id, path='.')`
 fetch biological assembly pdb file from RCSB.org

```
>>> fetch_ba('1xjr')
...
```

`rna_tools.rna_tools_lib.fetch_cif(cif_id, path='.')`
 fetch biological assembly cif file from RCSB.org

`rna_tools.rna_tools_lib.fetch_cif_ba(cif_id, path='.')`
 fetch biological assembly cif file from RCSB.org

`rna_tools.rna_tools_lib.generate_conect_records(residue_type, shift=0, no_junction=True)`
 Generate CONECT records for an RNA residue (U, A, C, or G).

Parameters

- **residue_type** (*str*) – Type of residue ('U', 'A', 'C', or 'G').
- **shift** (*int*) – Value to shift all atom serial numbers by.

Returns

List of CONECT records as strings.

Return type

list

`rna_tools.rna_tools_lib.get_rna_tools_path()`
 Return path to the rt.

`rna_tools.rna_tools_lib.load_rnas(path, verbose=True)`
 Load structural files (via glob) and return a list of RNAStructure objects.

Examples:

```
rnas = rtl.load_rnas('../rna_tools/input/mq/*.pdb')
```

`rna_tools.rna_tools_lib.replace_atoms(struc_fn, insert_fn, verbose=False)`
 Replace XYZ coordinate of the file (struc_fn) with XYZ from another file (insert_fn).

This can be useful if you want to replace positions of atoms, for example, one base only. The lines are muted based on atom name, residue name, chain, residue index (marked with XXXX below).:

```
ATOM 11 N1 A 2 27 303.441 273.472 301.457 1.00 0.00 N # file ATOM 1 N1 A 2 27 300.402
273.627 303.188 1.00 99.99 N # insert ATOM 11 N1 A 2 27 300.402 273.627 303.188 1.00 0.00 N
# inserted
```

```
XXXXXXXXXXXXXXXXXXXX # part used to find lines to be replaced
```

```
ATOM 1 P A 2 27 295.653 270.783 300.135 1.00119.29 P # next line
```

Parameters

- **struc_fn** (*str*) – path to the main PDB file
- **insert_fn** (*str*) – path to the file that will be injected in into the main PDB file

Returns

text in the PDB format

Return type

string

`rna_tools.rna_tools_lib.replace_chain(struc_fn, insert_fn, chain_id)`

Replace chain of the main file (`struc_fn`) with some new chain (`insert_fn`) of given chain id.

Parameters

- **struc_fn** (*str*) – path to the main PDB file
- **insert_fn** (*str*) – path to the file that will be injected in into the main PDB file
- **chain_id** (*str*) – chain that will be inserted into the main PDB file

Returns

text in the PDB format

Return type

string

`rna_tools.rna_tools_lib.set_chain_for_struc(struc_fn, chain_id, save_file_inplace=False, skip_ter=True)`

Quick & dirty function to set open a fn PDB format, set `chain_id` and save it to a file. Takes only lines with ATOM and TER.

`rna_tools.rna_tools_lib.sort_strings(l)`

Sort the given list in the way that humans expect. <http://blog.codinghorror.com/sorting-for-humans-natural-sort-order/>

3.2 Standardize your PDB files

3.2.1 rna_standardize.py

`rna_standardize.py` - standardize RNA PDB structures

Usage:

```
$ rna_standardize.py <pdb file>
```

`-v` is for verbose, `--version` for version ;)

```
usage: rna_standardize.py [-h] [--version] [--no-progress-bar] [--quiet]
                        [--renum-nmr] [--inplace] [-v] [--conect]
                        [--conect-no-linkage] [--dont-replace-hetatm]
                        [--keep-hetatm] [--here] [--no-hr] [--cif]
                        [--check-geometry] [--dont-fix-missing-atoms]
                        [--mdr] [--renumber-residues] [--suffix SUFFIX]
                        [--dont-report-missing-atoms] [--dont-rename-chains]
                        [--backbone-only] [--no-backbone] [--bases-only]
                        file [file ...]
```

file

file

-h, --help

show this help message and exit

--version

--no-progress-bar

for `--no-progress-bar` for `--rpr`

--quiet

limit prints

--renum-nmr

--inplace

in place edit the file! [experimental, only for `get_rnapuzzle_ready`, `--delete`, `--get-ss`, `--get-seq`, `--edit-pdb`]

-v, --verbose

tell me more what you're doing, please!

--conect

add conect to rpr file

--conect-no-linkage

dont add conect from our residue to another

--dont-replace-hetatm

replace 'HETATM' with 'ATOM' [tested only with `--get-rnapuzzle-ready`]

--keep-hetatm

keep hetatoms, [if not replaced anyway with ATOM, see `--dont-replace-hetatm`]

--here

save a file next to the original file with auto suffix for `--extract` it's `.extr.pdb`

--no-hr

do not insert the header into files

--cif

output as cif file

--check-geometry

check connectivity between residues and angles

--dont-fix-missing-atoms

used only with `--get-rnapuzzle-ready`

--mdr

get structures ready for MD (like `rpr` but without first)

--renumber-residues

by default is false

--suffix <suffix>

when used with `--inplace` allows you to change a name of a new file, `--suffix del` will give `<file>_del.pdb` (mind added `_`)

--dont-report-missing-atoms

used only with `--get-rnapuzzle-ready`

--dont-rename-chains

used only with `--get-rnapuzzle-ready`. By default: `--get-rnapuzzle-ready` rename chains from `ABC..` to stop behavior switch on this option

--backbone-only

used only with `--get-rnapuzzle-ready`, keep only backbone (= remove bases)

--no-backbone

used only with `--get-rnapuzzle-ready`, remove atoms of backbone (define as P OP1 OP2 O5')

--bases-only

used only with `--get-rnapuzzle-ready`, keep only atoms of bases

class `rna_tools.rna_tools_lib.RNAStructure(fn=")`

RNAStructure - handles an RNA pdb file.

fn

path to the structural file, e.g., `"../rna_tools/input/4ts2.pdb"`

Type

string

name

filename of the structural file, `"4ts2.pdb"`

Type

string

lines

the PDB file is loaded and ATOM/HETATM/TER/END go to self.lines

Type

list

get_rnapuzzle_ready(*renumber_residues=True, fix_missing_atoms=True, rename_chains=True, ignore_op3=False, report_missing_atoms=True, keep_hetatm=False, backbone_only=False, no_backbone=False, bases_only=False, save_single_res=False, ref_frame_only=False, p_only=False, check_geometry=False, connect=False, connect_no_linkage=False, verbose=False*)

Get rnapuzzle (SimRNA) ready structure.

Clean up a structure, get current order of atoms.

Parameters

- **renumber_residues** – boolean, from 1 to ..., second chain starts from 1 etc.
- **fix_missing_atoms** – boolean, superimpose motifs from the minilibrary and copy-paste missing atoms, this is super crude, so should be used with caution.

Submission format @<http://ahsoka.u-strasbg.fr/rnapuzzles/>

Run `rna_tools.rna_tools_lib.RNAStructure.std_resn()` before this function to fix names.

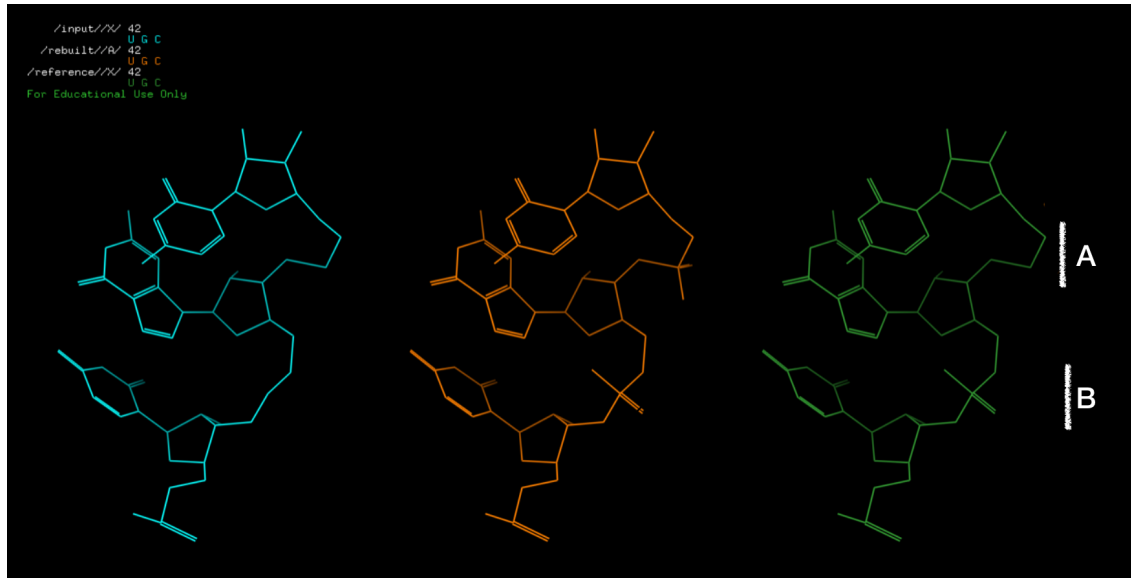


Figure: (Starting from left) input structure, structure with rebuilt atoms, and reference. The B fragment is observed in the reference used here as a “benchmark”, fragment A is reconstructed atoms (not observed in the reference”). 201122

- 170305 Merged with get_simrna_ready and fixing OP3 terminal added
- 170308 Fix missing atoms for bases, and O2'

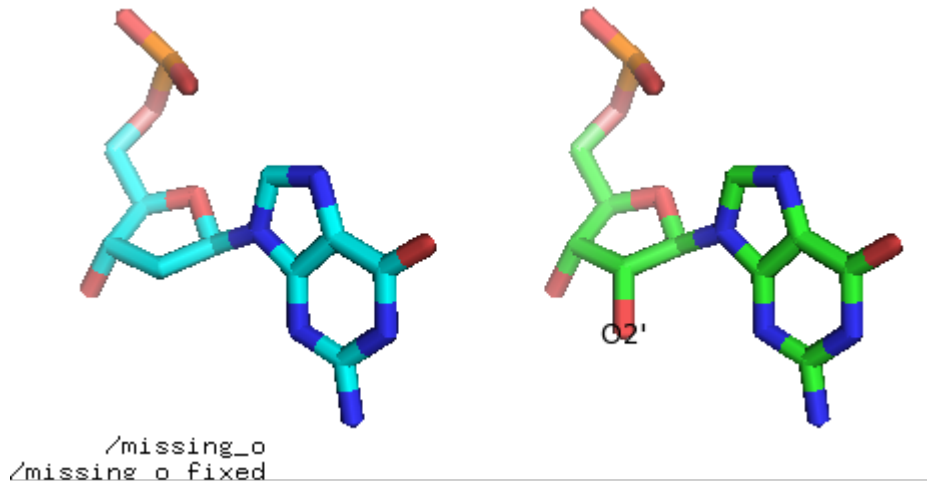


Fig. Add missing O2' atom (before and after).

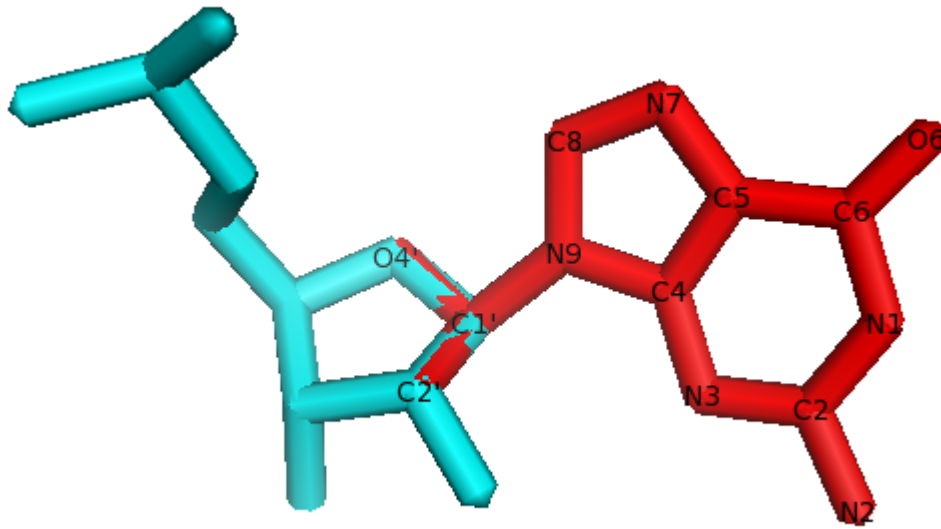
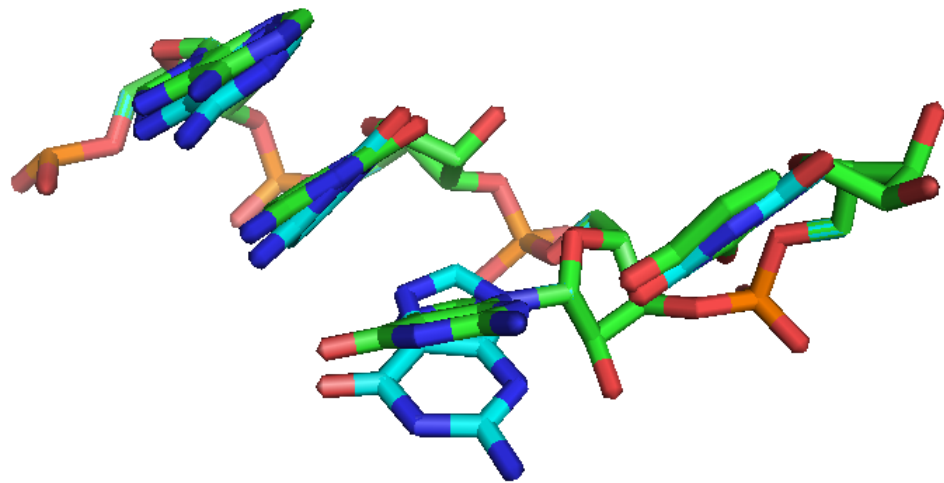


Fig. The residue to fix is in cyan. The G base from the library in red. Atoms O4', C2', C1' are shared between the sugar (in cyan) and the G base from the library (in red). These atoms are used to superimpose the G base on the sugar, and then all atoms from the base are copied to the residues.



```

/ACGU
'ACGU_no_bases_fix
/ACGU_no_bases

```

Fig. Rebuild ACGU base-less. It's not perfect but good enough for some applications.

Warning

It was only tested with the whole base missing!

Warning

requires: Biopython

Selection of atoms:

- phosphate group (3x, OP1 ,P, OP2),
- connector (2x O5', C5'), /5x
- sugar (5x, C4', O4', C3', O3', C1', C2'), /10
- extra oxygens from sugar (2x, O2' O3'), for now it's /12!
- A (10x), G (11x), C (8x), U(8x), max 12+11=23

And 27 unique atoms: {'N9', 'O2', 'OP2', "O2'", "O4'", 'C8', "O3'", "C1'", 'C2', 'C6', "C5'", 'N6', 'C5', "C4'", 'C4', "O5'", "C3'", 'O6', 'N2', 'N7', 'OP1', 'N1', 'N4', 'P', "C2'", 'N3', 'O4'}.

3.2.2 Atoms order

Atoms order, A as an example:

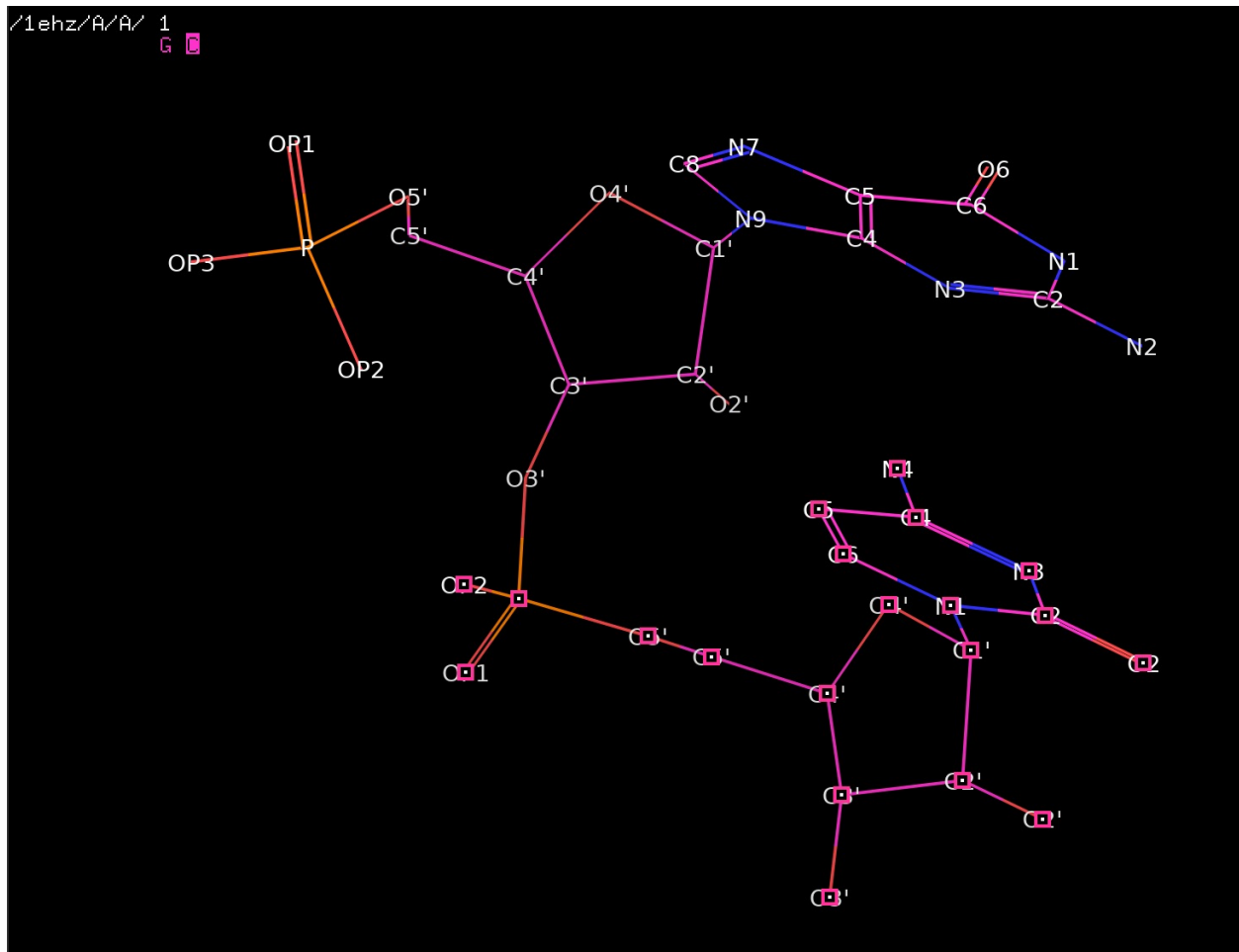
ATOM	1	P	G A	1	50.626	49.730	50.573	1.00100.19	P
ATOM	2	OP1	G A	1	49.854	48.893	49.562	1.00100.19	O
ATOM	3	OP2	G A	1	52.137	49.542	50.511	1.00 99.21	O
ATOM	4	O5'	G A	1	50.161	49.136	52.023	1.00 99.82	O
ATOM	5	C5'	G A	1	50.216	49.948	53.210	1.00 98.63	C
ATOM	6	C4'	G A	1	50.968	49.231	54.309	1.00 97.84	C
ATOM	7	O4'	G A	1	50.450	47.888	54.472	1.00 97.10	O
ATOM	8	C3'	G A	1	52.454	49.030	54.074	1.00 98.07	C
ATOM	9	O3'	G A	1	53.203	50.177	54.425	1.00 99.39	O
ATOM	10	C2'	G A	1	52.781	47.831	54.957	1.00 96.96	C
ATOM	11	O2'	G A	1	53.018	48.156	56.313	1.00 96.77	O
ATOM	12	C1'	G A	1	51.502	47.007	54.836	1.00 95.70	C
ATOM	13	N9	G A	1	51.628	45.992	53.798	1.00 93.67	N
ATOM	14	C8	G A	1	51.064	46.007	52.547	1.00 92.60	C
ATOM	15	N7	G A	1	51.379	44.966	51.831	1.00 91.19	N
ATOM	16	C5	G A	1	52.197	44.218	52.658	1.00 91.47	C
ATOM	17	C6	G A	1	52.848	42.992	52.425	1.00 90.68	C
ATOM	18	O6	G A	1	52.826	42.291	51.404	1.00 90.38	O
ATOM	19	N1	G A	1	53.588	42.588	53.534	1.00 90.71	N
ATOM	20	C2	G A	1	53.685	43.282	54.716	1.00 91.21	C
ATOM	21	N2	G A	1	54.452	42.733	55.671	1.00 91.23	N
ATOM	22	N3	G A	1	53.077	44.429	54.946	1.00 91.92	N
ATOM	23	C4	G A	1	52.356	44.836	53.879	1.00 92.62	C
ATOM	24	P	C A	2	54.635	50.420	53.741	1.00100.19	P
ATOM	25	OP1	C A	2	55.145	51.726	54.238	1.00100.19	O

3.2.3 OP3

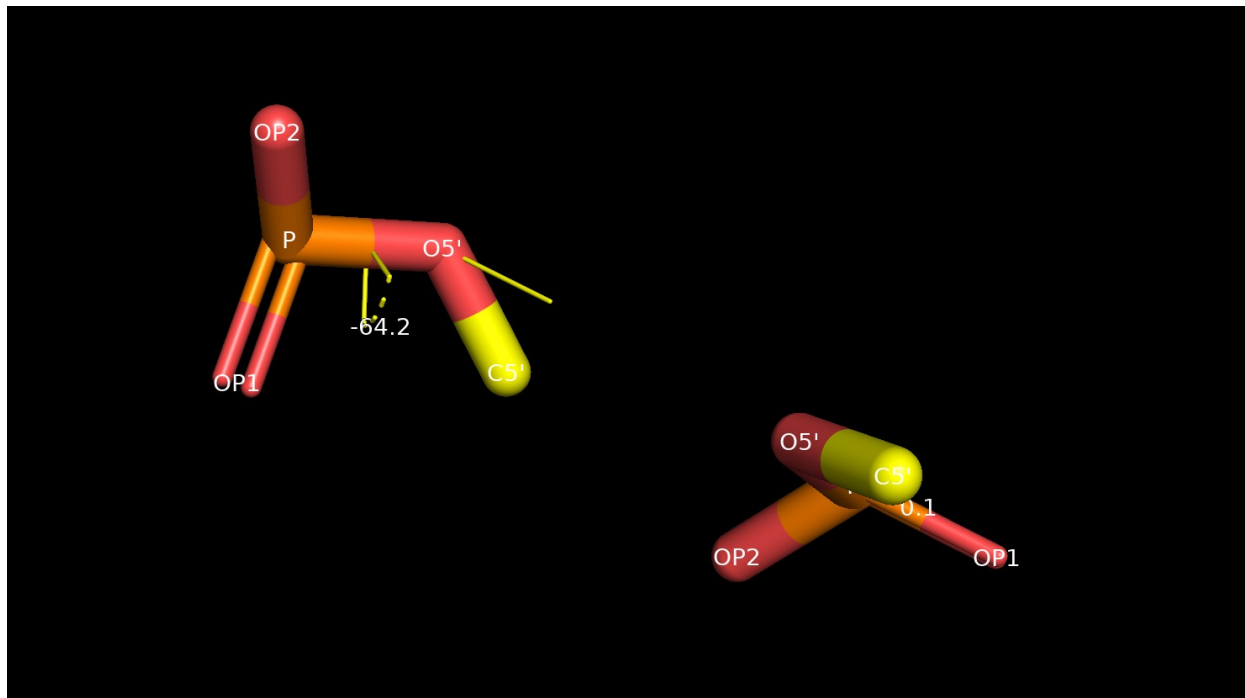
The first residue will have only OP1 and OP2 (OP3 will be removed):

ATOM	1	OP3	G	A	1	50.193	51.190	50.534	1.00	99.85	O
ATOM	2	P	G	A	1	50.626	49.730	50.573	1.00	100.19	P
ATOM	3	OP1	G	A	1	49.854	48.893	49.562	1.00	100.19	O
ATOM	4	OP2	G	A	1	52.137	49.542	50.511	1.00	99.21	O
ATOM	5	O5'	G	A	1	50.161	49.136	52.023	1.00	99.82	O
ATOM	6	C5'	G	A	1	50.216	49.948	53.210	1.00	98.63	C
ATOM	7	C4'	G	A	1	50.968	49.231	54.309	1.00	97.84	C

Listing. An example: *1ehz.pdb*.



3.3 Torsion angle analysis



3.3.1 rna_torsions.py

Examples:

```
$ rna_torsions.py ./input/4GXY_min.pdb
f, alphaprime, beta
input ./input/4GXY_min.pdb <Residue G het= resseq=2 icode= >, -64.20924484900823, -143.
↳18546007904766
input ./input/4GXY_min.pdb <Residue C het= resseq=3 icode= >, 2.3394112025736815, 70.
↳4052871669199
```

Comparison:

```
$ rna_x3dna.py input/4GXY_min.pdb -s
input: input/4GXY_min.pdb
  nt id  res  alpha  beta  gamma  delta  epsilon  zeta  e-z  chi  chi
↳phase-angle sugar-type ssZp  Dp  splay  paired
0  1  G  A.G2  NaN -143.2  153.7  82.5  -92.3 -31.9 -60(..) -179.0(anti) 19.5(C3
↳'-endo) ~C3'-endo 4.39 4.56 18.32 no paired
1  2  C  A.C3 -111.4  70.4  160.0  80.6  NaN  NaN  NaN -177.6(anti) 11.1(C3
↳'-endo) ~C3'-endo NaN NaN NaN no paired
```

```
usage: rna_torsions.py [-h] [-v] file
```

file

- h, --help**
show this help message and exit
- v, --verbose**
be verbose

3.4 PDB Edit Bfactor/Occupancy

3.4.1 rna_pdb_edit_occupancy_bfactor.py

rna_pdb_edit_occupancy_bfactor.py - edit occupancy or bfactor in PDB file.

Example:

```
rna_pdb_edit_occupancy_bfactor.py --occupancy --select A:1-40,B:1-22 \
    --set-to 0 \
    19_Bujnicki_Human_4_rpr_n0-000001.pdb

rna_pdb_edit_occupancy_bfactor.py --occupancy \
    --select A:1-2 \
    --select-atoms P+C4\' \
    --set-to 10 \
    -o test_data/3w3s_homologymodel_out.PD
    --set-not-selected-to 8
    test_data/3w3s_homologymodel.pdb
```

```
usage: rna_pdb_edit_occupancy_bfactor.py [-h] (--bfactor | --occupancy)
    [--select SELECT] [--set-to SET_TO]
    [--set-not-selected-to SET_NOT_SELECTED_TO]
    [-o OUTPUT] [--verbose]
    [--select-atoms SELECT_ATOMS]
    file
```

file

file

- h, --help**
show this help message and exit
- bfactor**
set bfactor
- occupancy**
set occupancy
- select <select>**
get chain, e.g A:1-10, works also for multiple chainse.g A:1-40,B:1-22
- set-to <set_to>**
set value to, default is 1
- set-not-selected-to <set_not_selected_to>**
set value to, default is 0

-o <output>, **--output** <output>

file output

--verbose

be verbose

--select-atoms <select_atoms>

select only given atoms can be only one atom, e.g. P or more, use ' for prims, e.g. P+C4'

`rna_tools.tools.rna_pdb_edit_occupancy_bfactor.rna_pdb_edit_occupancy_bfactor.edit_occupancy_of_pdb`(*txt*, *pdb*, *pdb_out*, *bfactor*, *occupancy*, *set_to*, *set_not_selected_to*, *select_atoms*, *v*=F

Change occupancy or bfactor of pdb file.

Load the structure, and first set everything to be *set_not_selected_to* and then set selected to *set_to*.

Parameters

- **txt** (*str*) – A:1-10, selection, what to change
- **pdb** (*str*) – filename to read as an input
- **pdb_out** (*str*) – filename to save an output
- **bfactor** (*bool*) – if edit bfactor
- **occupancy** (*bool*) – if edit occupancy
- **set_to** (*float*) – set to this value, if within selection
- **set_not_selected_to** (*float*) – set to this value, if not within selection
- **select_atoms** (*str*) – P, P+C4', use + as a separator
- **v** (*bool*) – be verbose

Returns

if OK, save an output to *pdb_out*

Return type

bool

Warning

this function requires BioPython

3.5 Add chain to a file

Example:

```
./rna_add_chain.py -c X ../../input/1msy_rnakbmd_decoy999_clx_noChain.pdb > ../../
↪output/1msy_rnakbmd_decoy999_clx_noChain_Xchain.pdb
```

From:

ATOM	1	O5'	U	1	42.778	25.208	46.287	1.00	0.00
ATOM	2	C5'	U	1	42.780	26.630	45.876	1.00	0.00
ATOM	3	C4'	U	1	42.080	27.526	46.956	1.00	0.00
ATOM	4	O4'	U	1	43.013	28.044	47.963	1.00	0.00
ATOM	5	C1'	U	1	42.706	29.395	48.257	1.00	0.00
ATOM	6	N1	U	1	43.857	30.305	47.703	1.00	0.00
ATOM	7	C6	U	1	45.057	29.857	47.308	1.00	0.00
ATOM	8	C5	U	1	46.025	30.676	46.763	1.00	0.00
ATOM	9	C4	U	1	45.720	32.110	46.702	1.00	0.00
ATOM	10	O4	U	1	46.444	32.975	46.256	1.00	0.00

to:

ATOM	1	O5'	U X	1	42.778	25.208	46.287	1.00	0.00
ATOM	2	C5'	U X	1	42.780	26.630	45.876	1.00	0.00
ATOM	3	C4'	U X	1	42.080	27.526	46.956	1.00	0.00
ATOM	4	O4'	U X	1	43.013	28.044	47.963	1.00	0.00
ATOM	5	C1'	U X	1	42.706	29.395	48.257	1.00	0.00
ATOM	6	N1	U X	1	43.857	30.305	47.703	1.00	0.00
ATOM	7	C6	U X	1	45.057	29.857	47.308	1.00	0.00
ATOM	8	C5	U X	1	46.025	30.676	46.763	1.00	0.00
ATOM	9	C4	U X	1	45.720	32.110	46.702	1.00	0.00
ATOM	10	O4	U X	1	46.444	32.975	46.256	1.00	0.00

in a loop:

```
for i in *; do rna_add_chain.py -c A $i > ../struc/${i}; done
```

```
rna_tools.tools.misc.rna_add_chain.get_parser()
```

3.5.1 rna_add_chain.py

```
usage: rna_add_chain.py [-h] [-c CHAIN] file
```

file

file

-h, --help

show this help message and exit

-c <chain>, --chain <chain>

a new chain, e.g. A

3.6 Measure distance between atoms

3.6.1 pdbs_measure_atom_dists.py

This is a quick and dirty method of comparison two RNA structures (stored in pdb files). It measures the distance between the relevant atoms (C4') for nucleotides defined as "x" in the sequence alignment.

author: F. Stefaniak, modified by A. Zyla, supervision of mmagnus

```
usage: pdbs_measure_atom_dists.py [-h] [-v]
      seqid1 seqid2 alignfn pdbfn1 pdbfn2
```

seqid1

seq1 id in the alignment

seqid2

seq2 id in the alignment

alignfn

alignment in the Fasta format

pdbfn1

pdb file1

pdbfn2

pdb file2

-h, --help

show this help message and exit

-v, --verbose

increase output verbosity

This is a quick and dirty method of comparison two RNA structures (stored in pdb files). It measures the distance between the relevant atoms (C4') for nucleotides defined as "x" in the sequence alignment.

author: F. Stefaniak, modified by A. Zyla, supervision of mmagnus

```
rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists.find_core(seq_with_gaps1,
                                                                           seq_with_gaps2)
```

Parameters

- **seq_with_gaps1** (*str*) – a sequence 1 from the alignment
- **seq_with_gaps2** – a sequence 2 from the alignment

Usage:

```
>>> find_core('GUUCAG-----UGAC-', 'CUUCGAGCAUUGCACUCCGGCUGCGAUG')
'xxxxxx-----xxxx-'
```

Returns

core="xxxxxx-----xxxx-"

```
rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists.get_parser()
```

`rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists.get_seq(alignfn, seqid)`

Get seq from an alignment with gaps.

Parameters

- **alignfn** (*str*) – a path to an alignment
- **seqid** (*str*) – seq id in an alignment

Usage:

```
>>> get_seq('test_data/ALN_OBJ1_OBJ2.fa', 'obj1')
SeqRecord(seq=SeqRecord(seq=Seq('GUUCAG-----UGAC-',
↳ SingleLetterAlphabet()), id='obj1', name='obj1', description='obj1', dbxrefs=[]),
↳ id='<unknown id>', name='<unknown name>', description='<unknown description>',
↳ dbxrefs=[])
```

Returns

SeqRecord

`rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists.map_coords_atom(structure)`

Args: *structure* (pdb): PDB Biopython object: with a pdb structure

Returns

a list of coords for atoms *structure* | *realNumber*: a list of residues

Return type

structIdict

`rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists.open_pdb(pdffn)`

Open pdb with Biopython.

Parameters

pdffn1 (*str*) – a path to a pdb structure

Returns

with a pdb structure

Return type

PDB Biopython object

4.1 RNA Sequence

RNA Sequence with secondary structure prediction methods.

This tool takes a given sequence and returns the secondary structure prediction provided by 5 different tools: RNAfold, RNAsubopt, ipknot, contextfold and centroid_fold. You must have these tools installed. You don't have to install all tools if you want to use only one of the methods.

It's easy to add more methods of your choice to this class.

4.1.1 Installation

Depends on what tools you want to use, follow the instructions below.

4.1.1.1 ContextFold

<https://www.cs.bgu.ac.il/~negevcb/contextfold/>

needs Java. Try this on Ubuntu 14-04 <https://askubuntu.com/questions/521145/how-to-install-oracle-java-on-ubuntu-14-04> Single chain only!

4.1.1.2 ViennaRNA

<https://www.tbi.univie.ac.at/RNA/>

For OSX install from the binary Installer from the page.

4.1.1.3 ipknot OSX

<https://github.com/satoken/homebrew-rnatools>

If one encounters a problem:

```
[mm] Desktop$ /usr/local/opt/bin/ipknot
dyld: Library not loaded: /usr/local/opt/glpk/lib/libglpk.40.dylib
  Referenced from: /usr/local/opt/bin/ipknot
  Reason: image not found
[1] 51654 abort /usr/local/opt/bin/ipknot
```

the solution is:

```
brew install glpk # on OSX
```

4.1.1.4 RNA Structure

<http://rna.urmc.rochester.edu/>

Works with 5.8.1; Jun 16, 2016.

Download <http://rna.urmc.rochester.edu/RNAstructureDownload.html> and untar it in `<RNA_PDB_TOOLS>/opt/RNAstructure/`; run `make`, the tools will be compiled in a folder `exe`. Set up `DATPATH` in your `bashrc` to `<RNA_PDB_TOOLS>/opt/RNAstructure/data_tables DATAPATH=/home/magnus/work/src/rna-pdb-tools/opt/RNAstructure/data_tables/` (read more <http://rna.urmc.rochester.edu/Text/Thermodynamics.html>). RNAs-structure can be run with SHAPE restraints, read more http://rna.urmc.rochester.edu/Text/File_Formats.html#Constraint about the format. The file format for SHAPE reactivity comprises two columns. The first column is the nucleotide number, and the second is the reactivity. Nucleotides for which there is no SHAPE data can either be left out of the file, or the reactivity can be entered as less than -500. Columns are separated by any white space.

4.1.1.5 MC-Sym

4.1.2 FAQ

- Does it work for more than one chain??? Hmm.. I think it's not. You have to check on your own. -magnus

4.1.3 TIPS

Should you need to run it on a list of sequences, use the following script:

```
from rna_tools import Seq
f = open("listOfSequences.fasta")
for line in f:
    if line.startswith('>'):
        print line,
    else:
        print line,
        s = Seq.Seq(line.strip()) # module first Seq and class second Seq #without strip this.
        ↪has two lines
        print s.predict_ss(method="contextfold"),
        #print s.predict_ss(method="centroid_fold")
```

Todo

- This class should be renamed to `RNASeq` and merged with `RNASeq` class from `RNAalignment`

```
exception rna_tools.Seq.MethodNotChosen
```

```
class rna_tools.Seq.RNASequence(seq, ss="", name='rna_seq')
```

```
    RNASequence.
```

```
    Usage:
```

```

>>> seq = RNASequence("CCCCUUUUGGGG")
>>> seq.name = 'RNA03'
>>> print(seq.predict_ss("RNAfold", constraints="((((...))))"))
>RNA03
CCCCUUUUGGGG
((((...)))) (-6.40)

```

eval(*ss=""*, *no_dangling_end_energies=False*, *verbose=False*)

Evaluate energy of RNA sequence.

Parameters

- **ss** (*optional*)
- **no_dangling_end_energies** (*Boolean*)
- **verbose** (*Boolean*)

Returns

Energy (float)

The RNAeval web server calculates the energy of a RNA sequence on a given secondary structure. You can use it to get a detailed thermodynamic description (loop free-energy decomposition) of your RNA structures.

Simply paste or upload your sequence below and click Proceed. To get more information on the meaning of the options click the help symbols. You can test the server using this sample sequence/structure pair.

An equivalent RNAeval command line call would have been:

```
RNAeval -v -d0 < input.txt
```

Read more: <<http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAeval.cgi>>

get_foldability(*ss=""*, *verbose=False*)

Calculate foldability based on EntRNA.

Steps:

- parse SS into basepairs,
- calculate foldability

Configuration:

- Set ENTRNA_PATH to the folder where ENTRNA_predict.py is.

Cmd wrapper in here:

```
python ENTRNA_predict.py --seq_file pseudoknotted_seq.txt --str_file_
↳ pseudoknotted_str.txt
```

Su, C., Weir, J. D., Zhang, F., Yan, H., & Wu, T. (2019). ENTRNA: a framework to predict RNA foldability. BMC Bioinformatics, 20(1), 1–11. <http://doi.org/10.1186/s12859-019-2948-5>

predict_ss(*method='RNAfold'*, *constraints=""*, *enforce_constraint=False*, *shapefn=""*, *explore=""*, *verbose=0*, *path=""*)

Predict secondary structure of the seq.

Parameters

- **method** – {mcfold, RNAfold}

- **onstraints**
- **shapefn** (*str*) – path to a file with shape reactivities
- **verbose** (*boolean*)

It creates a seq fasta file and runs various methods for secondary structure prediction. You can provide also a constraints file for RNAfold and RNAsubopt.

Methods that can be used with constraints: RNAsubopt, RNAfold, mcfold.

Methods that can be used with SHAPE constraints: RNAfold.

ContextFold

Example:

```
$ java -cp bin contextFold.app.Predict in:CCCCUUUGGGG
CCCCUUUGGGG
(((...)))
```

It seems that a seq has to be longer than 9. Otherwise:

```
$ java -cp bin contextFold.app.Predict in:UUUUUUGGG
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10

# this is OK
$ java -cp bin contextFold.app.Predict in:CCCCUUUGGG
CCCCUUUGGG
.(((...)))
```

RNAstructure

Example:

```
>>> seq = RNASequence("GGGUUUUCCC")
>>> print(seq.predict_ss("rnastructure"))
> ENERGY = -4.4 rna_seq
GGGUUUUCCC
(((...)))
```

and with the shape data:

```
>>> print(seq.predict_ss("rnastructure", shapefn="data/shape.txt"))
> ENERGY = -0.2 rna_seq
GGGUUUUCCC
.(((...)))
```

the shape data:

```
1 10
2 1
3 1
```

You can easily see that the first G is unpaired right now! The reactivity of this G was set to 10. Worked!

MC-Fold

MC-Fold uses the online version of the tool, this is very powerful with constraints:

```

rna_seq
acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
((((.....))).....((((.....((((((((((((.....))))))))))))))..
↳)))
curl -Y 0 -y 300 -F "pass=lucy" -F mask="((((.....))).....((((.....
↳..((((((((((((((((.....))))))))))))))..)))" -F sequence=
↳"acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
↳" https://www.major.irc.ca/cgi-bin/MC-Fold/mcfold.static.cgi
mcfold::energy best dynamics programming: -53.91
(-53.91, '((((.....))).....((((.....((((((((((((.....
↳))))))))))))..)))')

curl -Y 0 -y 300 -F "pass=lucy" -F mask="((((.....))).....((((.....
↳..((((((((((((((((.....))))))))))))))..)))" -F sequence=
↳"acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
↳" https://www.major.irc.ca/cgi-bin/MC-Fold/mcfold.static.cgi
mcfold::energy best dynamics programming: -34.77
(-34.77, '((((.....))).....((((.....((((((((((((.....
↳))))))))))))..)))')

acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
((((.....))).....((((.....((((((((((((.....))))))))))))))..
↳)))
curl -Y 0 -y 300 -F "pass=lucy" -F mask=
↳"(((xxxxxxx)))xxxxxxx(((xxxxxxxxxxxxxxxx((((((((xxxxxxxxxxxxxxxx))))))))xx)))
↳" -F sequence=
↳"acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
↳" https://www.major.irc.ca/cgi-bin/MC-Fold/mcfold.static.cgi
mcfold::energy best dynamics programming: -34.77
(-34.77, '((((.....))).....((((.....((((((((((((.....
↳))))))))))))..)))')

acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
((((.....))).....((((.....((((((((((((.....))))))))))))))..
↳)))
curl -Y 0 -y 300 -F "pass=lucy" -F mask=
↳"(((*****)))*****(((*****((((((((*****))))))))**)))
↳" -F sequence=
↳"acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
↳" https://www.major.irc.ca/cgi-bin/MC-Fold/mcfold.static.cgi
mcfold::energy best dynamics programming: -77.30
(-71.12, '((((.....)))).....((((.....))..((((((((((((.....
↳))))))))))))))')

acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
((((.....))).....((((.....((((((((((((.....))))))))))))))..
↳)))
curl -Y 0 -y 300 -F "pass=lucy" -F mask=
↳"(((**[[[[**]])*****([[[**]])**((((((((*****))))))))**)))
↳" -F sequence=
↳"acucggcuaggcgaguauaaaauagccgucaggccuagcgcguccaagccuagcccccucuggggcugggcgaagggucggg
↳" https://www.major.irc.ca/cgi-bin/MC-Fold/mcfold.static.cgi

```

(continues on next page)

(continued from previous page)

```

mcfold::energy best dynamics programming: -77.30
('-77.30',
↪ '((((**[[[[[*]])*****(((***)]]])****((((((((((((((((***)**)))))))))))))**))
↪ ')

```

explore

The sub-optimal search space can be constrained within a percentage of the minimum free energy structure, as MC-fold makes use of the Waterman-Byers algorithm [18, 19]. Because the exploration has an exponential time complexity, increasing this value can have a dramatic effect on MC-Fold's run time.

Parisien, M., & Major, F. (2009). RNA Modeling Using the MC-Fold and MC-Sym Pipeline [Manual] (pp. 1–84).

```
rna_tools.Seq.load_fasta_ss_into_RNAseqs(fn, debug=True)
```

4.2 RNA Secondary Structure

Secondary structure analysis

exception `rna_tools.SecondaryStructure.ExceptionOpenPairsProblem`

```
rna_tools.SecondaryStructure.draw_ss(title, seq, ss, img_out, resolution=4, verbose=False)
```

Draw Secondary Structure using VARNA (you need correct configuration for this).

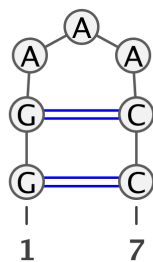
If everything is OK, return None, if an error (=exception) return stderr.

Usage:

```

>>> seq = 'GGAAACC'
>>> ss = '((...))'
>>> img_out = 'output/demo.png'
>>> draw_ss('rna', seq, ss, img_out)
>>> print('Made %s' % img_out)
Made output/demo.png

```



rna

Can be used with <http://geekbook.readthedocs.io/en/latest/rna.html>

`rna_tools.SecondaryStructure.parse_vienna_to_pairs(ss, remove_gaps_in_ss=False)`

Parse Vienna (dot-bracket notation) to get pairs.

Parameters

- **ss** (*str*) – secondary structure in Vienna (dot-bracket notation) notation
- **remove_gaps_in_ss** (*bool*) – remove - from ss or not, design for DCA (tpp case ss = "((((((((((((((((.....)))))).....-----)....." works with pk of the first level, [[]]

Returns

(pairs, pairs_pk)

Return type

list of two lists

Examples:

```
>>> parse_vienna_to_pairs('((..))')
([[1, 6], [2, 5]], [])

>>> parse_vienna_to_pairs('([[[]]])')
([[1, 6], [2, 5]], [[3, 8], [4, 7]])

>>> parse_vienna_to_pairs('((-))')
([[1, 6], [2, 5]], [])

>>> parse_vienna_to_pairs('((-))', remove_gaps_in_ss=True)
([[1, 4], [2, 3]], [])

>>> parse_vienna_to_pairs('((((.....)')
Traceback (most recent call last):
  File "/usr/lib/python2.7/doctest.py", line 1315, in __run
    compileflags, 1) in test.globs
  File "<doctest __main__.parse_vienna_to_pairs[4]>", line 1, in <module>
    parse_vienna_to_pairs('((((.....)')
  File "./SecondaryStructure.py", line 106, in parse_vienna_to_pairs
    raise ExceptionOpenPairsProblem('Too many open pairs (()) in structure')
ExceptionOpenPairsProblem: Too many open pairs (()) in structure
```

4.2.1 rna_dot2ct.py

The output file is <input-file>.ct

Wrapper to

RNAstructure: software for RNA secondary structure prediction and analysis. (2010). RNAstructure: software for RNA secondary structure prediction and analysis., 11, 129. <http://doi.org/10.1186/1471-2105-11-129>

```
usage: rna_dot2ct.py [-h] [-v] file
```

file

Input is: >seq aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (...(((((((((.))))))))))

-h, --help

show this help message and exit

-v, --verbose

be verbose

4.2.2 Secondary structure format conversion

rna_convert_pseudoknot_formats

Run this as:

```
python rna-pk-simrna-to-one-line.py test_data/simrna.ss
```

Convert:

```
> a
...((.(.(((. . . . .)))..(((.(. . . . .).))).. . . . .).....(((. . . . .)))
.....(((. . . . .))).....
```

to:

```
> a
...((.(.(((. . . . .)))..(((.(. . . . .)[[[[. . . . .])]).. . . . .).....]]](((. . . . .)))
```

and:

```
>2 chains
(((((. . . . .)))).....(.((. . . . .). . . . .).....)
.....(((((. . . . .)))).....
```

to:

```
>2 chains
((((([[[[[]]]])))).....(.((. . . . .[[]]]))..((. . . . .).....)
```

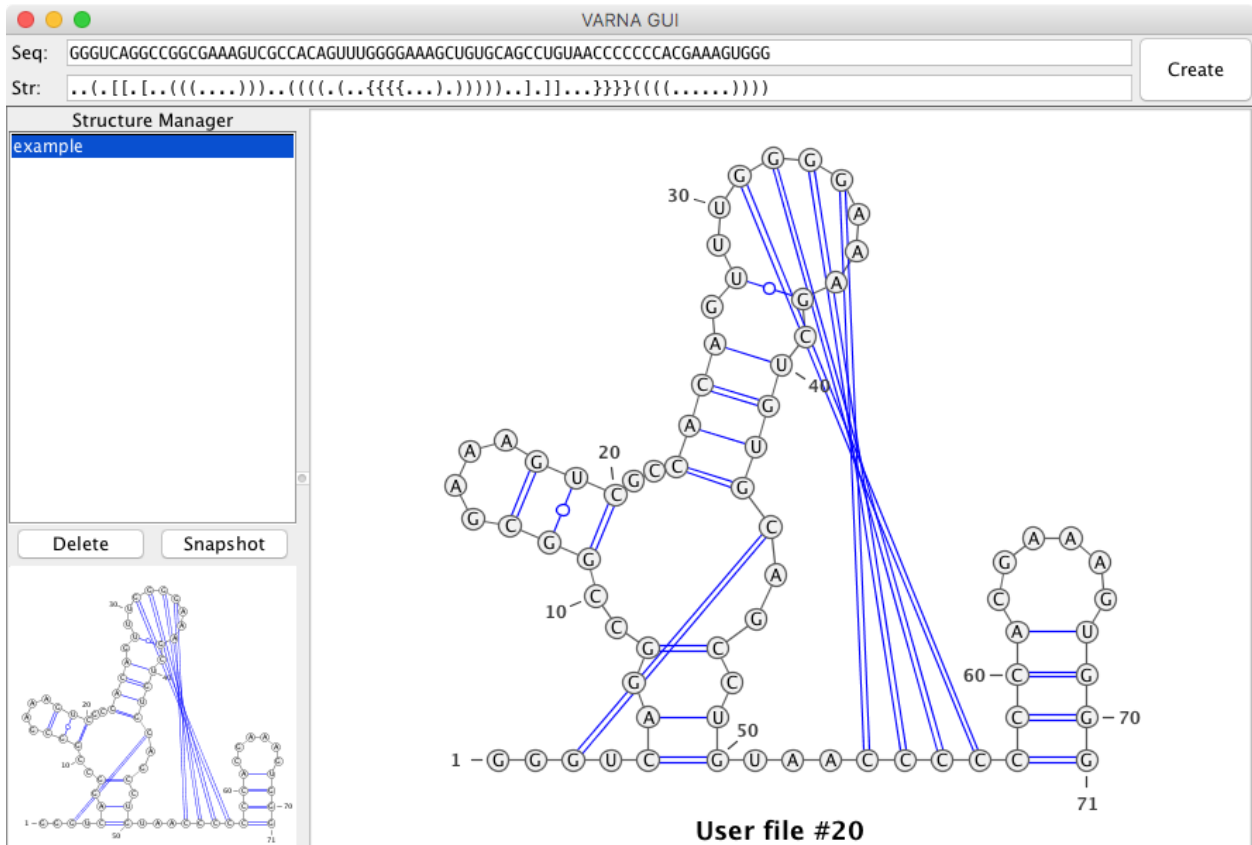
and:

```
> b
..(. . . . .(((. . . . .)))..(((.(. . . . .).))).....(((. . . . .)))
...((.(. . . . .).)).....
.....(((. . . . .))).....
```

to:

```
> b
..(. . . . .[[].[. . . . .])..(((.(. . . . .){[[[. . . . .]})])..]..]..}}}}(((. . . . .)))
```

and it works with VARNA:



Convert a secondary structure with a pk to the SimRNA format:

```

rna_convert_pseudoknot_formats git:(master) python rna_ss_pk_to_simrna.py test_data/ss_
→with_pk.ss
((((([[[[[]]]])))).....((....[[]]]))..(((. .)))....))
((((.....))).....((....(.....)..(((. .)))....))
.....((((.....))).....
    
```

```

rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_pk_to_simrna.get_multiple_lines(ss)
rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_pk_to_simrna.get_parser()
rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_pk_to_simrna.is_pk(ss)
    
```

4.3 Search

4.3.1 Blast PDB

A super-simple wrapper around Blast on the PDB db (online).

```
class rna_tools.BlastPDB.BlastPDB(seq)
```

BlastPDB - run Blast online on the PDB database.

This can be used in Jupiter based RNA notebooks, e.g. <https://github.com/mmagnus/rna-pdb-tools/blob/master/rp18.ipynb>

Warning: getBlastPDB1 has been permanently removed as part of our announced shutdown on December 9th, 2020. <https://www.rcsb.org/pdb/rest/getBlastPDB1>

Usage:

```
>>> p = BlastPDB(
  ↳ 'GGGUCAGGCCGGCGAAAGUCGCCACAGUUUGGGGAAAGCUGUGCAGCCUGUAACCCCCCACGAAAGUGGG')
>>> p.search()
>>> p.result
u'<HTML>\n<TITLE>BLAST Search Results</TITLE>...
```

Parameters

seq – string

search()

Search online the seq.

4.3.2 Rfam Search

A super-simple wrapper around cmscan (Infernal) on local Rfam database.

class rna_tools.RfamSearch.RfamSearch

RfamSearch (local).

Rfam is a collection of multiple sequence alignments and covariance models representing non-coding RNA families. Rfam is available on the web <http://rfam.xfam.org/>. The website allow the user to search a query sequence against a library of covariance models, and view multiple sequence alignments and family annotation. The database can also be downloaded in flatfile form and searched locally using the INFERNAL package (<http://infernal.wustl.edu/>). The first release of Rfam (1.0) contains 25 families, which annotate over 50 000 non-coding RNA genes in the taxonomic divisions of the EMBL nucleotide database.

Infernal (“INFERENCE of RNA ALIGNment”) is for searching DNA sequence databases for RNA structure and sequence similarities. It is an implementation of a special case of profile stochastic context-free grammars called covariance models (CMs). A CM is like a sequence profile, but it scores a combination of sequence consensus and RNA secondary structure consensus, so in many cases, it is more capable of identifying RNA homologs that conserve their secondary structure more than their primary sequence.

Infernal *cmscan* is used to search the CM-format Rfam database.

Setup:

- download the database from <ftp://ftp.ebi.ac.uk/pub/databases/Rfam/CURRENT> (file: Rfam.cm.gz, ~30mb)
- install <http://eddylab.org/infernal/>
- set up RFAM_DB_PATH in the config file of rna-tools.
- compress Rfam.cm

Example of compressing the database:

```
$ cpress Rfam.cm
Working... done.
Pressed and indexed 3016 CMs and p7 HMM filters (3016 names and 3016 accessions).
Covariance models and p7 filters pressed into binary file: Rfam.cm.ilm
SSI index for binary covariance model file: Rfam.cm.ili
```

(continues on next page)

(continued from previous page)

```
Optimized p7 filter profiles (MSV part)  pressed into:  Rfam.cm.i1f
Optimized p7 filter profiles (remainder) pressed into:  Rfam.cm.i1p
```

Cite: Nawrocki and S. R. Eddy, *Infernal 1.1: 100-fold faster RNA homology searches*, *Bioinformatics* 29:2933-2935 (2013).

cmscan(*seq*, *verbose=False*)

Run cmscan on the seq.

Usage:

```
>>> seq = RNASequence("GGCGCGGCACCGUCCGCGGAACAAACGG")
>>> rs = RfamSearch()
>>> hit = rs.cmscan(seq)
>>> print(hit)
# cmscan :: search sequence(s) against a CM database...
```

Parameters

seq – string

Returns

result

Return type

string

exception `rna_tools.RfamSearch.RfamSearchError`

4.4 RNA Alignment

RNAAlignment - a module to work with RNA sequence alignments.

To see a full demo what you can do with this util, please take a look at the jupyter notebook (https://github.com/mmagnus/rna-pdb-tools/blob/master/rna_tools/tools/rna_alignment/rna_alignment.ipynb)

Load an alignment in the Stockholm:

```
alignment = ra.RNAalignment('test_data/RF00167.stockholm.sto')

or fasta format::

import rna_alignment as ra
alignment = ra.fasta2stockholm(alignment.fasta)
alignment = ra.RNAalignment
```

Parameters of the alignment:

```
print(alignment.describe())
```

Consensus SS:

```
print(alignment.ss_cons_with_pk)
```

Get sequence/s from the alignment:

```
>>> seq = a.io[0]
```

4.4.1 RNASeq

```
class rna_tools.tools.rna_alignment.rna_alignment.RNASeq(id, seq, ss=None)
```

RNASeq.

Parameters

- **id** (*str*) – id of a sequence
- **seq** (*str*) – seq, it be uppercased.
- **ss** (*str*) – secondary structure, default None

seq_no_gaps

```
seq.replace('-', '')
```

Type

str

ss_no_gaps

```
ss.replace('-', '')
```

Type

str

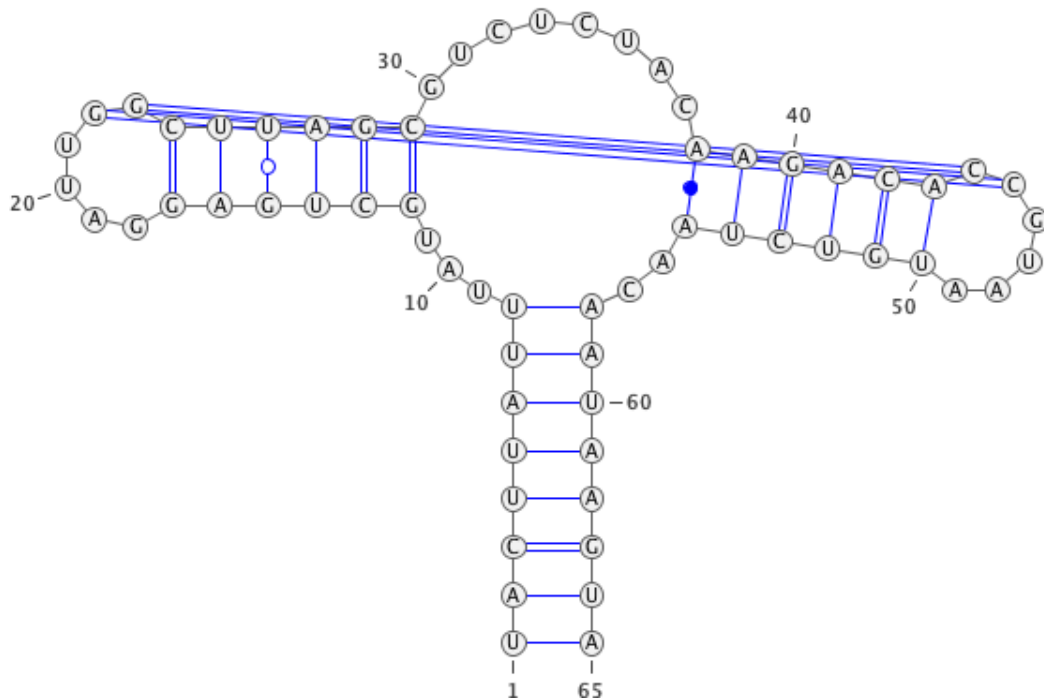
Warning

```
>>> if 'EF' in s.id: print('Y')
Y
>>> if 'EF' in s: print('Y')
# nothing
```

```
draw_ss(title="", verbose=False, resolution=1.5)
```

Draw secondary structure of RNA with VARNA.

VARNA: Visualization Applet for RNA A Java lightweight component and applet for drawing the RNA secondary structure



Cite: VARNA: Interactive drawing and editing of the RNA secondary structure Kevin Darty, Alain Denise and Yann Ponty Bioinformatics, pp. 1974-197,, Vol. 25, no. 15, 2009

<http://varna.lri.fr/>

get_conserved(*consensus*, *start=0*, *to_pymol=True*, *offset=0*)

```

Start          UCGGGGUGCCCUUCUGCGUG-----AAGGC-
UGAGAAAUACCCGU-----AUCACCUG-AUCUGGAU-AAUGC
XXXXXXXXXXXXXGXGXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXX-XCUGAGAXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX-----XXXXXXXXX-
XXXXXXXXXX-ACXUG
    
```

get_distance_to(*nseq*)

Get distance of self.seq to nseq.

get_ss_std()

remove_columns(*to_remove*)

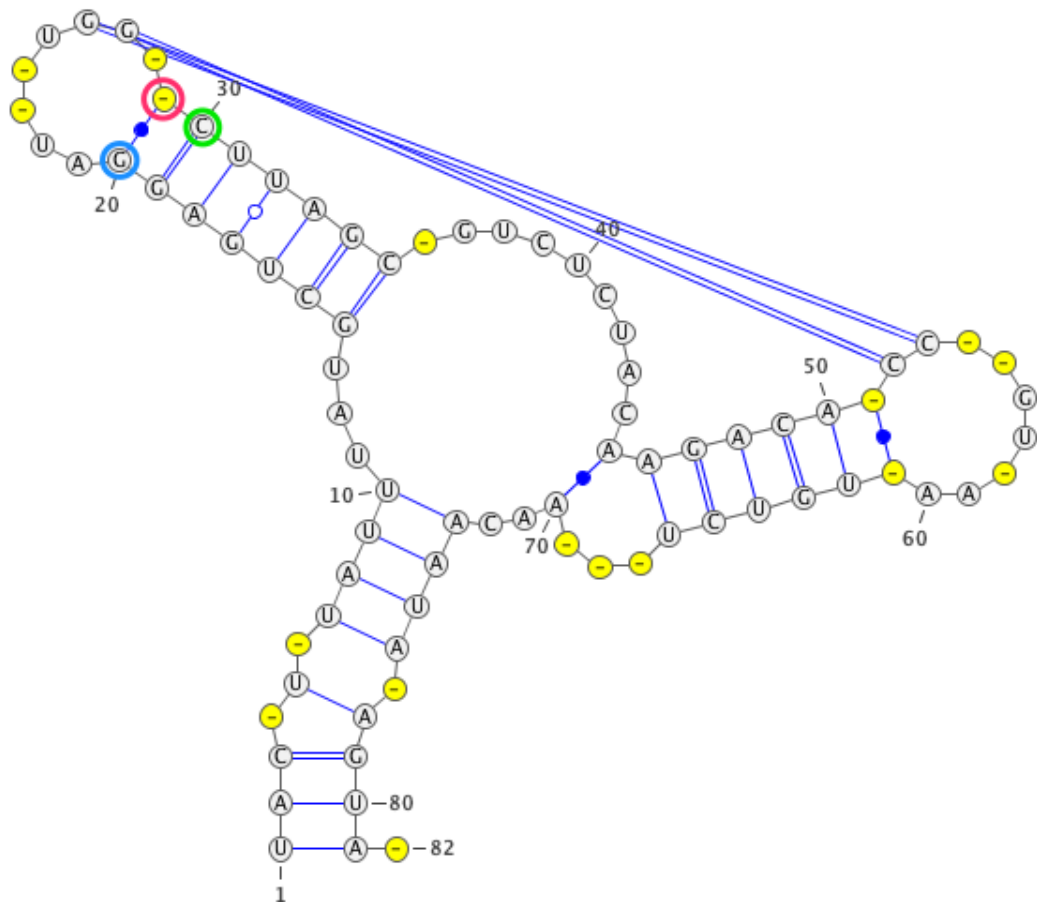
indexing from 0

remove_gaps(*check_bps=True*, *only_canonical=True*, *allow_gu=True*)

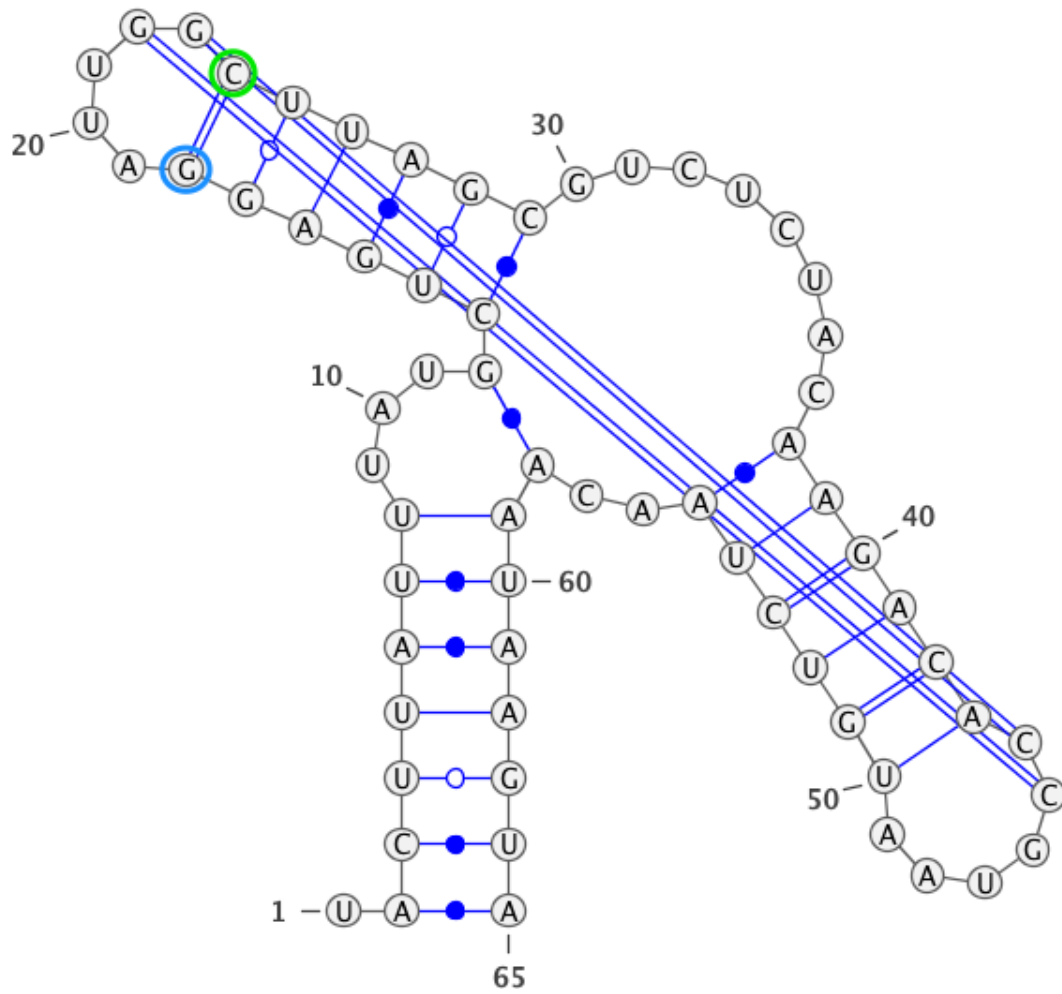
Remove gaps from seq and secondary structure of the seq.

Parameters

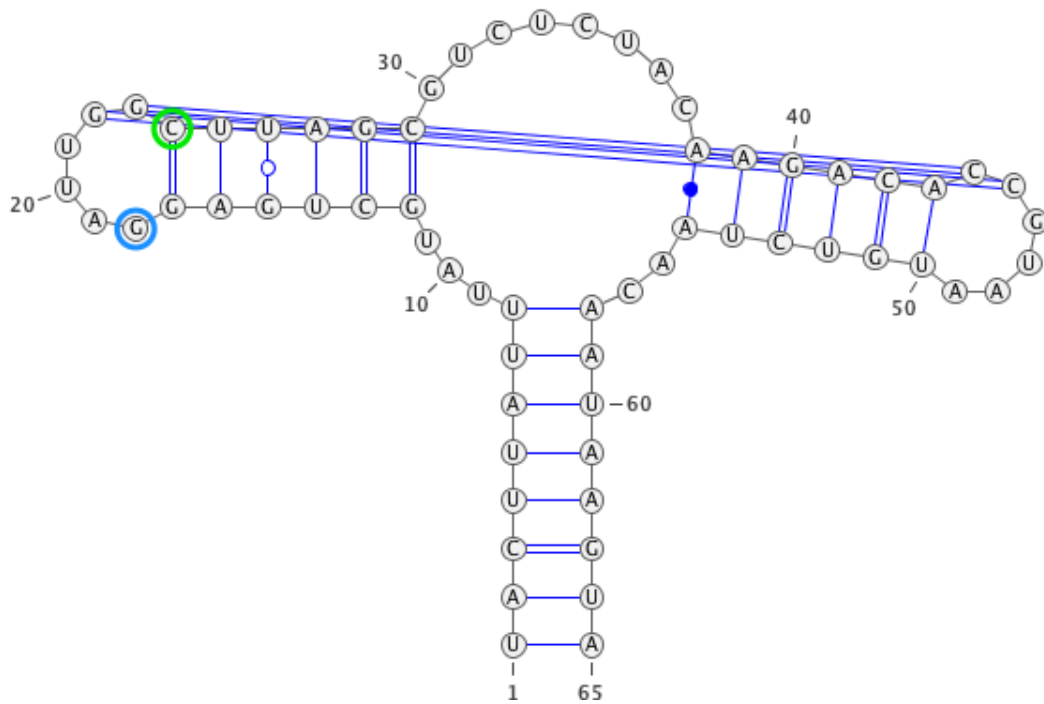
- **check_bps** (*bool*) – fix mistakes as
- **only_canonical** (*bool*) – keep in ss only pairs GC, AU
- **allow_gu** (*bool*) – keep in ss also GU pair



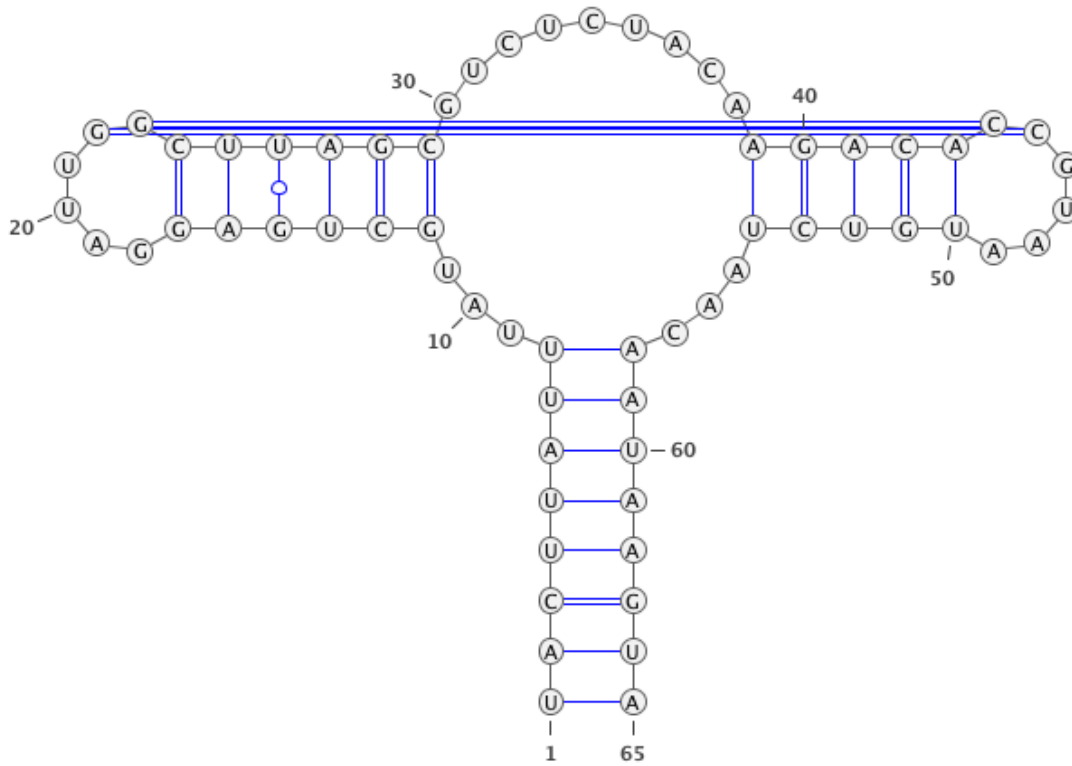
A residue "paired" with a gap.



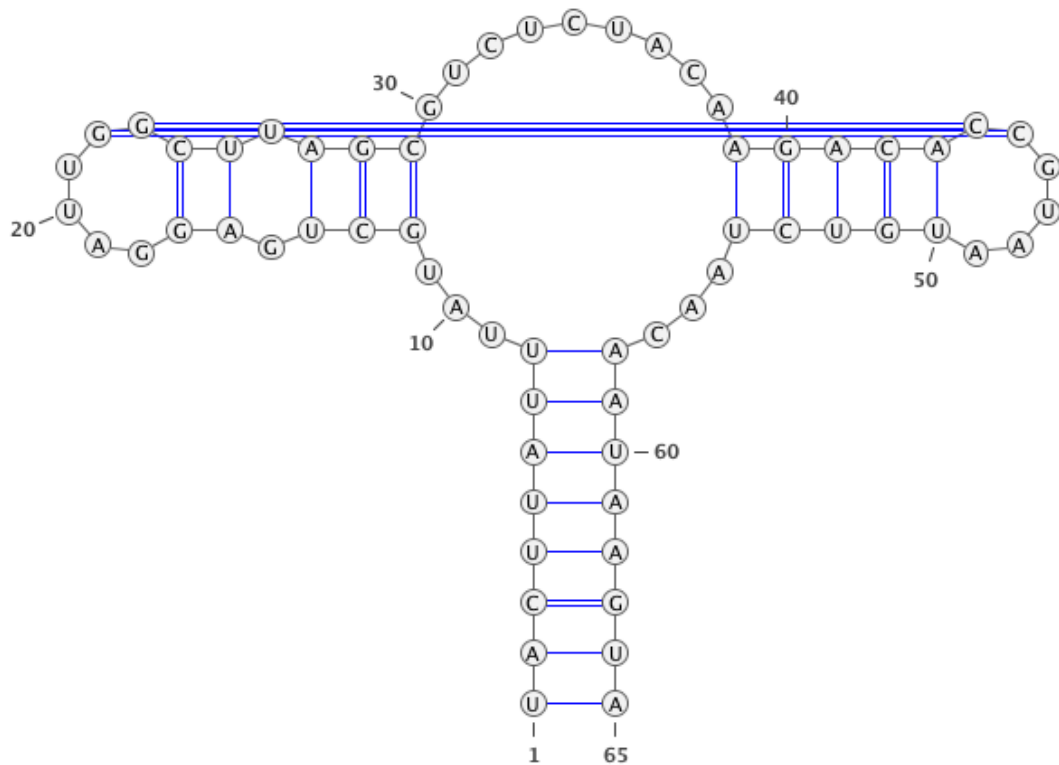
paired with any residues (in the blue circle). If yes, then this residues is unpair (in this case) -> .).



if `only_canonical` (by default) is True then only GC, AU can be paired.



If `allow_gu` is False (by default is True) then GU pair is also possible.



If you provide seq and secondary structure such as:

```
GgCcGGggG . GcggG . cc . u . aAUCAAuACCC . GaAA . GGGGAAUAaggCc . gGCc . gu . . . . . CU . . . . .
↪ uugugcgGUuUUcaAgCccCCgGcCaCCuuuu
((((((((.....((.(.....(((.....))).....))))).(((.(.....
↪ ))).....)))))).....
```

gaps will be remove as well.

ss_to_bps()

Convert secondary structure into a list of basepairs.

Returns

a list of base pairs, e.g. [[0, 80], [1, 79], [2, 78], [4, 77], [6, 75], [7, 74], ...]

Return type

bps (list)

4.4.2 RNAalignment

class rna_tools.tools.rna_alignment.rna_alignment.RNAalignment(fn="", fetch="")

RNA alignment - adapter class around BioPython to do RNA alignment stuff

Usage (for more see IPython notebook https://github.com/mmagnus/rna-tools/blob/master/rna_tools/tools/rna_alignment/rna_alignment.ipynb)

```
>>> a = RNAalignment('test_data/RF00167.stockholm.sto')
>>> print(a.tail())
>>> print(a.ss_cons)
```

Parameters

- **fn** (*str*) – Filename
- **io** (*Bio.AlignIO*) – `AlignIO.read(fn, “stockholm”)`
- **lines** (*list*) – List of all lines of fn
- **seqs** (*list*) – List of all sequences as class:*RNASeq* objects
- **rf** (*str*) – ReFERENCE annotation, the consensus RNA sequence

Read more:

- <http://biopython.org/DIST/docs/api/Bio.AlignIO.StockholmIO-module.html>

and on the format itself

- https://en.wikipedia.org/wiki/Stockholm_format
- <http://sonnhammer.sbc.su.se/Stockholm.html>

Warning

fetch requires urllib3

align_seq(*seq*)

Align seq to the alignment.

Using self.rf.

Parameters

seq (*str*) – sequence, e.g. `-GGAGAGUA-GAUGAUUCGCGUUAAGUGUGUGUGA-AUGGGAUGUC...`

Returns

seq that can be inserted into alignment, `.-.GG.AGAGUA-GAUGAUUCGCGUUA!. -> -`

Return type

str

copy_ss_cons_to_all(*verbose=False*)

copy_ss_cons_to_all_editing_sequence(*seq_id, before, after*)

Change a sequence’s sec structure.

Parameters

- **seq_id** – string, sequence id to change, eg: `AE009948.1/1094322-1094400`
- **before** – string, character to change from, eg: `,`
- **after** – string, character to change to, eg: `.`

Warning

before and after has to be one character long

describe()

Describe the alignment.

`> print(a.describe())` SingleLetterAlphabet() alignment with 13 rows and 82 columns

find_core(*ids=None*)

Find common core for ids.

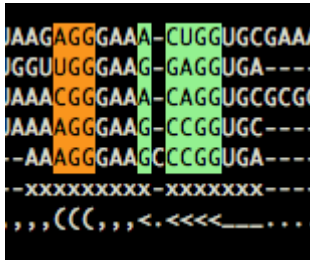


Fig. By core, we understand columns that have all homologous residues. The core is here marked by *x*.

Parameters

- **id** – list, ids of seq in the alignment to use

find_seq(*seq, verbose=False*)

Find seq (also subsequences) and reverse in the alignment.

Parameters

- **seq** (*str*) – seq is upper()
- **verbose** (*bool*) – be verbose

```
seq =
↪ "ggaucgcugaacccgaaagggcgggggaccagaaauggggcgaaucucuuccgaaaggaagaguaggguuacuccuucgacccgagcccgc
↪ "
hit = a.find_seq(seq, verbose=False)
ggaucgcugaacccgaaagggcgggggaccagaaauggggcgaaucucuuccgaaaggaagaguaggguuacuccuucgacccgagcccgc
Match: AL939120.1/174742-174619
ID: AL939120.1/174742-174619
Name: AL939120.1
Description: AL939120.1/174742-174619
Number of features: 0
/start=174742
/end=174619
/accession=AL939120.1
Per letter annotation for: secondary_structure
Seq('CCAGGUAAGUCGCC-G-C--ACCG-----GUCA-----...GGA',
↪ SingleLetterAlphabet())
GGAUCGCUGAACCCGAAAGGGCGGGGGACCCAGAAAUGGGCGAAUCUCUCCGAAAGGAAGAGUAGGGUUACUCCUUCGACCCGAGCCC
```

find_seq_exact(*seq, verbose=False*)

Find seq (also subsequences) and reverse in the alignment.

Parameters

- **seq** – string, seq, seq is upper()
- **verbose** – boolean, be verbose or not

format_annotation(*t*)

get_clean_ss(*ss*)

get_distances()

Get distances (seq identity) all-vs-all.

With BioPython.

blastn: Bad alphabet 'U' in sequence 'AE008922.1/409481-409568' at position '7' only for DNA?

read more (also about matrix at <<http://biopython.org/wiki/Phylo>> and <HTTP://biopython.org/DIST/docs/api/Bio.Phylo.TreeConstruction.DistanceCalculator-class.html>)

get_gc_rf()

Return (str) #=GC RF or '' if this line is not in the alignment.

get_seq(seq_id)**get_seq_ss(seq_id)****get_seq_with_name(seq_name)****get_shift_seq_in_align()**

RF_cons vs '#=GC RF' ???

get_ss_cons()**Returns**

SS_cons_pk line or None if there is now SS_cons_pk.

get_ss_cons_pk()**Returns**

SS_cons_pk line or None if there is now SS_cons_pk:

get_ss_remove_gaps(seq, ss)**Parameters**

- **seq** – string, sequence
- **ss** – string, ss

```
UAU-AACAUAUAAUUUGACAAUAUGG-GUCAUAA-GUUUCUACCGGAAUACC-GUAAAUAUUCU—GACUAUG-  
UAUA- (((.((((,((((((_____)))))))-,,,,,,((((((_____.)))))))).)))).
```

get_the_closest_seq_to_ref_seq(verbose=False)

Example:

```
>>> a = RNAalignment("test_data/RF02221.stockholm.sto")  
>>> a.get_the_closest_seq_to_ref_seq()  
AF421314.1/431-344
```

head()**map_seq_on_align(seq_id, resis, v=True)****Parameters**

- **seqid** – seq_id, 'CP000721.1/2204691-2204775'
- **resis** – list resis, [5,6]

maps:

```
[5, 6, 8]
CAC-U
CAC-U-
CAC-U-UA
[4, None, 6]
```

`map_seq_on_seq(seq_id, seq_id_target, resis, v=True)`

Parameters

- **seq_id** – seq_id, ‘AAML04000013.1/228868-228953’
- **seq_id_target** – seq_id of target, ‘CP000721.1/2204691-2204778’
- **resis** – list resis, [5,6]

map:

```
[4, 5, 6]
UAU-A
UAU-AA
UAU-AAC
[5, 6, 7]
CAC-U
CAC-U-
CAC-U-U
[4, None, 5]
```

`plot(plot_fn='rchie.png')`

`reload_alignment()`

`remove_empty_columns(verbose=False)`

Remove empty columns in place.

Example:

```
>>> a = RNAalignment("test_data/zmp.stk")
>>> print(a)
SingleLetterAlphabet() alignment with 6 rows and 319 columns
---ACCUUGC GCGACUGGCGAAUCC-----...AAU CP001644.1/756294-756165
--GUCUCGCGCGACUGGCGACUUUG-----...GAA CU234118.1/352539-352459
UGAGUUUUCUGCGACUGACGGAUUAU-----...CUG BAAV01000055.1/2897-2982
GCCCGUUCGCGUGACUGGCGCUAGU-----...CGA CP000927.1/5164264-5164343
----GGGUCGUGACUGGCGAACA-----...---- zmp
UCACCCUGCGUGACUGGCGAUA-----...GUU AP009385.1/718103-718202
>>> a.remove_empty_columns()
>>> print(a)
SingleLetterAlphabet() alignment with 6 rows and 138 columns
---ACCUUGC GCGACUGGCGAAUCC-UGAAGCUGCUUUG-AGCG...AAU CP001644.1/756294-756165
--GUCUCGCGCGACUGGCGACUUUG-----...GAA CU234118.1/352539-352459
UGAGUUUUCUGCGACUGACGGAUUAU-----...CUG BAAV01000055.1/2897-2982
GCCCGUUCGCGUGACUGGCGCUAGU-----...CGA CP000927.1/5164264-5164343
----GGGUCGUGACUGGCGAACA-----G-----...---- zmp
UCACCCUGCGUGACUGGCGAUA-----GAACCCUCGGGUU...GUU AP009385.1/718103-718202
```

go over all seq modifies self.nss_cons

property ss_cons_std

property ss_cons_with_pk

go over ss_cons and overwrite bp is there is pk (ss_cons_pk)

```
ss_cons:      (((.((((,((((((_____))))),,,,,,((((((_____.))))).)),),))))).      ss_cons_pk:
.....[[.....]].....      ss_cons_with_pk:
(((.((((,((((([_____]_____.))))),,,,,,((((([_____]_____.))))).)),),))))).
‘return ss_cons_with_pk: string, e.g. (((.((((,((((([_____]_____.))))
```

property ss_cons_with_pk_std

subset(ids, verbose=False)

Get subset for ids:

```
# STOCKHOLM 1.0
#=GF WK Tetrahydrofolate_riboswitch
```

```
AAQK01002704.1/947-1059 -U-GC-AAAAUAGGUUCCAUGC.. #=GC SS_cons .(((.(---((((((((...
#=GC RF .g.gc.aGAGUAGggugccgugc.. //
```

tail()

trimmed_rf_and_ss()

Remove from RF and SS gaps.

Returns

trf, tss - new RF and SS

Return type

(str,str)

write(fn, verbose=False)

Write the alignment to a file

4.4.2.1 rna_alignment_get_species.py

This is an improved version of the script that uses the Rfam MySQL database online interface (thanks @akaped for this idea) (so you need to be connected to the Internet, of course). Redirect the output to the file.

```
AB010698.1/46416-46518      GUC-CC--- Homo-sapiens-(human)/1-107      GUC-CUCGC-----
AARH01001853.1/272694-272796 GUC-UC--- Saccharomyces-cerevisiae-(baker's-yeast)/1-113 GUU-CGCCAAGUA--
X60506.1/390-492           GUC-CC--- Saccharomyces-cerevisiae-(baker's-yeast).1/1-113 GUU-CGCCAAGUA--
AC146705.11/15272-15374    GUC-CC--- Saccharomyces-cerevisiae-RM11-1a/1-113      GUU-CGCCAAGUA--
AASG02002949.1/2307-2409   GUC-CC--- vs Schizosaccharomyces-pombe-(fission-yeast)/1-153 UGA-UC-----
AACV01009611.1/38269-38371 GUC-CC--- -/1-0
AAAA02013555.1/2292-2394  GUC-CC--- Aedes-aegypti-(yellow-fever-mosquito)/1-107   GUC-UUUGC-----
CR855100.1/43897-43999    GUC-CC--- Aegilops-tauschii/1-105      GUC-CN-----
```

Warning

This scripts needs mysql-connector-python-rf module to connect the Rfam MySQL server, so install it before using:
pip install mysql-connector-python-rf.

Example:

```
$ rna_alignment_get_species.py RF00004.stockholm.stk
# STOCKHOLM 1.0
Sorex-araneus-(European-shrew)          AUCGCU-UCU----CGGCC--UUU-U
```

Examples 2:

```
$ rna_alignment_get_species.py u5_rfam_u5only.stk --verbose
# STOCKHOLM 1.0
#=GF WK U5_spliceosomal_RNA
#=GF NC 39.90
#=GF RT The spliceosomal snRNAs of Caenorhabditis elegans.
#=GF TC 40.00
#=GF RN [3]
(...)
#=GF AC RF00020
#=GF SE Zwieb C, The uRNA database, PMID:9016512; PMID:18390578
#=GF GA 40.00
#=GF BM cmbuild -F CM SEED
#=GF TP Gene; snRNA; splicing;
Bos-taurus-(cattle)          GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
Sorex-araneus-(European-shrew)  GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
Ictidomys-tridecemlineatus-(thirteen-lined-ground- GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
Monodelphis-domestica-(gray-short-tailed-opossum)  GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
Oryctolagus-cuniculus-(rabbit)  GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
Cavia-porcellus-(domestic-guinea-pig)  GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
Ochotona-princeps-(American-pika)  GAUC-GUAUAAAUCUUUCGCCUUUUACUAAAGA-
↪UUUCCG----UGG-A--GA-G
```

```
usage: rna_alignment_get_species.py [-h] [-v] [--debug] [--id-width ID_WIDTH]
                                     [--evo-mapping EVO_MAPPING]
                                     [--evo-mapping-default] [--one]
                                     [--osfn OSFN] [--rfam]
                                     alignment
```

alignment

alignment

-h, --help

show this help message and exit

-v, --verbose

be verbose

--debug

--id-width <id_width>

define width of ids, trim species name when longer than this

```
--evo-mapping <evo_mapping>
--evo-mapping-default
--one
--osfn <osfn>
    cache file
--rfam
```

4.4.2.2 rna_alignment_calc_energy.py

Calculate energy (.cet) format:

```
UGGC-CCCUGCGCAA-GGAUGACA
(((..(((.....)).))..)))
(((..(((***)).))..)))
```

Examples:

```
$ rna_alignment_calc_energy.py --template alignments/u6-lower.cet alignments/u6-only-
↳ RemovedGapped.stk -v
    --loop-upper guaa --loop-lower guaa
    --loop-upper-cst '(..)' --loop-lower-cst '(..)'
calc-energy2.py --template u6atac-template.txt u6atac_u6only.sto -v
./calc-energy2.py --template alignments/u6-lower.cet --one alignments/u6-lower-stem-only.
↳ sto
```

Takes cet files (calc-energy-templates):

```
$ rna_alignment_calc_energy.py --template test_data/u6-lower.cet --one test_data/u6-only.
↳ stk -v # --loop-seq test_data/u6-only-loop-seq-u6-lower
N/A% (0 of 182) |
↳ | Elapsed Time: 0:00:00 ETA: --:--
↳ :=====
AB010698.1/46467-46488
(((..(((.....)).))..)))
UGGC-CCCUGCGCAA-GGAUGACA
lower -----
UGG ugcgca ACA
(((*****)))
UGGugcgcaACA
((((..))) -10.64
upper -----
UGGC-CCCUGCGCAA-GGAUGACA
CCC ugcgca AGG
CCCUgcgcaAGG
((((..))) -9.6
↳ seq          up_ss          id  low_energy          low_seq          low_ss  up_energy          up_
0  AB010698.1/46467-46488          -10.64  UGGugcgcaACA  (((((..))))))          -9.6
↳ CCCugcgcaAGG  (((((..))))))
Done: u6-only-loop-seq-u6-lower
```

by parsing output from MC-Sym:

```
domains have 5451 elements.
10:47:16 up 141 days, 26 min, 0 users, load average: 1.45, 1.30, 1.56
Score: -999.000 GAACAUGGUUCUUGCCUUUUACCAGAACCAUCCGGGUGUUG
Total number of MB structures with 3 stems: 16041
(overlaps: 0, !energy: 335585)
</pre><P><H2>Sorting the structures...
<P></H2><pre></pre><H2><P><P><P>Filtered and Sorted solutions:<P><P><P></H2><pre>
</pre><H2><P><P><P><a HREF="http://biww2.informatik.uni-freiburg.de/Software/MARNA/
↪index.html" target="_blank">MARNA</a>-formatted:<P><P><P></H2><pre>
GAACAUGGUUCUUGCCUUUUACCAGAACCAUCCGGGUGUUG
((((((((((..))))))))))((((((((((..)))))))) -33.20 ( -0.69)
((((((((((...))))))))))((((((((((..)))))))) -33.17 ( -0.69)
((((((((((((((((((..)))))))))))))).....)) -32.40 ( +0.00)

Backtracking with 2 variables (stems):
domains have 5451 elements.
10:47:16 up 141 days, 26 min, 0 users, load average: 1.45, 1.30, 1.56
Score: -999.000 GAACAUGGUUCUUGCCUUUUACCAGAACCAUCCGGGUGUUG
Total number of MB structures with 2 stems: 9555
(overlaps: 0, !energy: 165582)
</pre><P><H2>Sorting the structures...
<P></H2><pre></pre><H2><P><P><P>Filtered and Sorted solutions:<P><P><P></H2><pre>
</pre><H2><P><P><P><a HREF="http://biww2.informatik.uni-freiburg.de/Software/MARNA/
↪index.html" target="_blank">MARNA</a>-formatted:<P><P><P></H2><pre>
GAACAUGGUUCUUGCCUUUUACCAGAACCAUCCGGGUGUUG
((((((((((..))))))))))((((((((((..)))))))) -33.20 ( -0.69)
((((((((((...))))))))))((((((((((..)))))))) -33.17 ( -0.69)
((((((((((((((((((..)))))))))))))).....)) -32.40 ( +0.00)
```

```
usage: rna_alignment_calc_energy.py [-h] [--debug] [--one] [--method METHOD]
                                     [--csv CSV] [--loop-seq]
                                     [--template TEMPLATE] [--flanks FLANKS]
                                     [-v]
                                     alignment
```

alignment

an alignment in the Stockholm format

-h, --help

show this help message and exit

--debug

--one

one only for the first seq

--method <method>

mcfold or rnastructure_CycleFold

--csv <csv>

--loop-seq

--template <template>

--flanks <flanks>

GC be default

-v, --verbose

4.4.2.3 rna_align_get_ss_from_fasta.py

Input as a file:

```
>ade
GCU-U-CAUAUAAUCCUAAUGAUUAGG-UUUGGGA-GUUUCUACCAAGAG-CC--UUAAA-CUCUU---GAUUAUG-AAGU-
(((.((((,(((((_____)))))))-,,,,,((((((_____)))))).),,)))..)))).
```

to get:

```
>ade
GCUUCAUAUAAUCCUAAUGAUUAGGUUUGGGAGUUUCUACCAAGAGCCUUAACUCUUGAUUAUGAAGU
(((((((...((((((.....))))))).....((((((.....))))).)))))
```

```
usage: rna_align_get_ss_from_fasta.py [-h] file
```

file

subsection of an alignment

-h, --help

show this help message and exit

4.4.2.4 rna_align_get_ss_from_stk.py

Process an alignment in the Stockholm format to get sequences and secondary structures:

Example:

```
$ rna_align_get_ss_from_stk.py aligns/gmp_RF01786.stockholm_p75_pk.sto
AAOX01000007.1/31274-31356
AAGAAUAUAGAACACUGUGAGCGGUUUUAUUUGCACUUUAAACCGCUUGGAGUGACUAGUGCAGCCGCAUAUGAUCUA
.(((.((((.((((.....(((((((.....))))))).....)))).....))))..)))).
CP000724.1/3560727-3560809
AAAAAUGUAGAGCAAUGAACUGCAGGUAUACAUGGACGCCUUAACUGCAGGGAUGUAGUGGCGUAACCGACUAACAAUAUU
((.((((.((((.....(((((((.....[...[...].)))))).....)).....[...[...].)))).))..))
AACY023761929.1/1009-1091
AUAAUUUGGUGGCGUUGAUGUGCCCUUUGUAUCUGGUCGCUUGAGGGGUACGGAGCCAAUAGCGAAACCGCCGCCUCAUAG
.(((.....(((((((.....[...[...].)))))).....)).....[...[...].)))).
```

```
usage: rna_align_get_ss_from_stk.py [-h] file
```

file

subsection of an alignment

-h, --help

show this help message and exit

4.4.2.5 rna_align_distance_to_seq.py

Calculate

“Process an alignment in the Stockholm format to get sequences and secondary structures:

Example:

```
$ rna_align_distance_to_seq.py test_data/gmp_ref.sto test_data/gmp_ref_distance.csv

distance                id
0      1.00                gmp
1      0.69      AE000513.1/1919839-1919923
2      0.73      BA000004.3/387918-388001
3      0.69      ABFD02000011.1/154500-154585
4      0.73      AE015927.1/474745-474827
5      0.75      AAWL01000006.1
6      0.72      AM180355.1
7      0.72      CP001116.1/102374-102457
8      0.65      AJ965256.1/1260708-1260792

                                seq
0  -----GCGCGGAAAC-AAUGAUGAAU--GGG-UUUA-AAUUGGGC-...
1  CUGUCGAAGAGACGC-GAUGAAUCCC--GCC-CUGUAAUUCGGGC-...
2  AAUCAAUAGGGAAGC-AACGAAGCAU--AGC-CUUU-AUAUGGAC-...
3  AAAUAUUUAGAGAU-GUUGAAGUAU--AUU-CUAUUA-UUGGGC-...
4  AUUUUAAGAGGAAAU-UUUGAACUAU--AUA-CUU--AUUUGGGC-...
5  --UGCAA-UGGGUGU-GAUGAAGUCC--GGA-CAGUAAUGUGGGC-...
6  AAUAUUU-UAGAAAC-UGAGAAGUAU--AUC-UUAUUA-UUGGGC-...
7  AUAACGGCACGAAGC-AAUGAAAUGU--UCG-AUGU-AACCGGGC-...
8  AAAUUAAGGGGAAGC-GUUGAGCCGC--UAC-CCAU-AUGUGGUUC...

                                ss:
0  ((((((((((((((.((((.....(((..(((.(.....
1  ((((((((((((((.((((.....(((..(((.(.....
2  ((((((((((((((.((((.....(((..(((.(.....
3  ((((((((((((((.((((.....(((..(((.(.....
4  ((((((((((((((.((((.....(((..(((.(.....
5  ((((((((((((((.((((.....(((..(((.(.....
6  ((((((((((((((.((((.....(((..(((.(.....
7  ((((((((((((((.((((.....(((..(((.(.....
8  ((((((((((((((.((((.....(((..(((.(.....
```

```
usage: rna_align_distance_to_seq.py [-h] file output
```

file

an alignment in the Stockholm format, the first seq will be used to calculate distance to (#TODO pick any seq)

output

csv pandas file

-h, --help

show this help message and exit

4.4.2.6 rna_align_foldability.py

Calculate statistics of foldability on an alignment.

The tool uses ENTRANA [1] to calculate, what the authors called, foldability (column: “foldability”) of a given sequence into a given secondary structure.

Next, MC-Fold [2] is executed to calculate free energy (column: “mcsym”) on the sequence and the secondary structure obtained based on the alignment. The secondary structure is used as constraints.

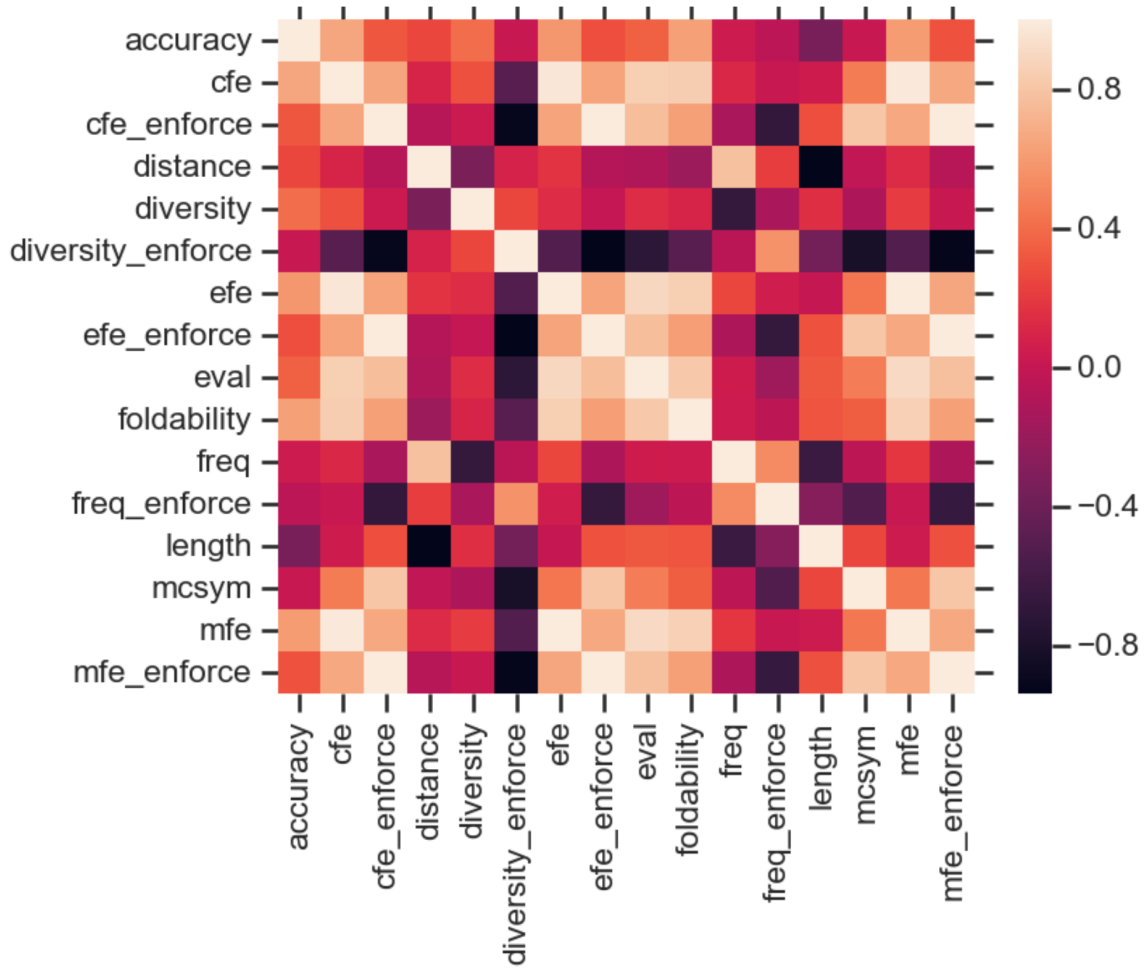
The third used program is RNAfold from the Vienna package [3]. Also, in this case the secondary structure obtained with rna-tools from the RNA alignment is used as constraints, columns: “mfe” (minimum free energy), “mfess” (secondary structure for minimum free energy state), “cfe” (minimum free energy of centroid), “cfess” (secondary structure for centroid, “diversity” (ensemble diversity), “efe” (free energy of the thermodynamic ensemble), “efess” (secondary structure for the thermodynamic ensemble), “freq” (frequency of mfe structure in ensemble). RNAfold is also executed in with “-enforceConstraint” where the constraints are enforced. This run gives analogous values as the default RNAfold, to all RNAfold column “_enforce” is added.

The tool is able to calculate the distance Levenshtein (the difference between the two sequences)(column: “distance”) from the target sequence and all sequence in the alignment to test if there is a bias in the accuracy towards the most similar sequences.

Another tool used from the Vienna package is RNAeval. The tool calculates free energy for a given sequence and secondary structure.

The accuracy is expressed as the median of core RMSD of 10% the lowest core RMSD models for the given sequences.

RNA Puzzle 17: accuracy vs foldability



The correlations:

accuracy	1.000000
cfe	0.653813
foldability	0.622038
mfe	0.607340
efe	0.585077
diversity	0.404350
eval	0.349499
cfe_enforce	0.311744
mfe_enforce	0.302973
efe_enforce	0.280929
distance	0.256870
freq	0.037037
diversity_enforce	0.018429
mcsym	0.017533
freq_enforce	-0.037991
length	-0.340809

The data:

We tested correlations between the above-mentioned statistics, and the highest correlation, 0.65 () was achieved to the centroid free energy calculated with RNAFold, which suggests that to some extent this metric could be used to pick sequence from the alignment to pick sequences that are more likely to fold.

However, this needs further investigation and the detailed analysis an all cases and more folded sequences.

1. Su C, Weir JD, Zhang F, Yan H, Wu T. ENTRNA: a framework to predict RNA foldability. BMC Bioinformatics. BioMed Central 2019
2. Parisien M, Major F. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. Nature 2008;452:51-5
3. Lorenz R, Bernhart SH, Honer Zu Siederdisen C, Tafer H, Flamm C, Stadler PF, et al. ViennaRNA Package 2.0. Algorithms Mol Biol. BioMed Central; 2011;6:26-14.

Example:

```
$ python rna_align_foldability.py test_data/gmp_ref.sto test_data/gmp_foldability.csv
                                cfess_enforce  distance  diversity
0  ((((((.....((((((((.....)))))))))...  1.00      3.96
1  ((((((((((.....((((((((.....)))))))))...  0.69      5.56
2  .....((((((((.....)))))))))...          0.73      3.84
3  .....((((((((.....)))))))))...          0.69      5.92
4  (((.....((((((((.....)))))))))...        0.73      7.49
5  ((((((.....((((((((.....)))))))))...     0.75      7.92
6  .(((((((((((.....((((((((.....)))))))))... 0.72      5.83
7  .....((((((((.....)))))))))...          0.72      7.35
8  ...(((((((((((.....((((((((.....)))))))))... 0.65      4.86

diversity_enforce  efe  efe_enforce
0  2.89 -14.77 -13.75
1  3.70 -19.52 -18.25
2  0.00 -15.41  0.00
3  0.00 -13.55  0.00
4  2.46 -8.58 -6.91
5  6.37 -20.72 -20.08
6  2.92 -11.87 -11.38
7  0.00 -14.59  0.00
8  3.83 -21.16 -20.64

                                efess
0  ((((((.....((((((((.....)))))))))...
1  {{.(((((((((((.....((((((((.....)))))))))...
2  .....((((((((.....)))))))))...
3  .....(((((((((((.....((((((((.....)))))))))...
4  .....{(((.,{{{,.....((((((((.....)))))))))...
5  {(((.(.(((.(.(((.(.((((((((({{.....,.....)}))..))...
6  .(((((((((((.....((((((((.....)))))))))...
7  .....{.,{{{.(.(((((((((((.....((((((((.....)))))))))...
8  ...{(((((((((((.....((((((((.....)))))))))...

                                length  mcsym
0  ..... 75.0 -39.73
1  ..... 85.0 -37.89
```

(continues on next page)

(continued from previous page)

```

2          ...          84.0 -35.40
3          ...          86.0 -36.11
4          ...          83.0 -37.37
5          ...          80.0 -43.59
6          ...          84.0 -42.95
7          ...          84.0 -36.55
8          ...          85.0 -43.58

          mcsym comment
0 energy best dynamics programming
1          BP energy
2          BP energy
3          BP energy
4 energy best dynamics programming
5 energy best dynamics programming
6 energy best dynamics programming
7          BP energy
8 energy best dynamics programming

          mcsym ss      mfe mfe_enforce
0 ((((((.(....(((((((.....)))))))))... -13.9      -12.9
1 ((((((((((.(.....(((((((.....)))))))))... -18.0      -17.3
2 ...(.(((.(((.....(((((((.....)))))))))... -14.0         0.0
3 .(.(((.(((.....(((((((.....)))))))))... -12.0         0.0
4 (((....(((.(((.....(((((((.....)))))))))... -7.2          -6.1
5 ((((.(((.(((.....(((((((.....)))))))))... -18.6      -18.6
6 .(((((((((((.(.....(((((((.....)))))))))... -10.5      -10.5
7 ...(.(((.(((.....(((((((.....)))))))))... -12.8         0.0
8 ...(((((((((((.(.....(((((((.....)))))))))... -19.8      -19.8

          mfess
0 ((((((.....(((((((.....)))))))))...
1 ((.(((((((((((.(.....(((((((.....)))))))))...
2 .....(((.(((.....(((((((.....)))))))))...
3 .....(((((((((((.(.....(((((((.....)))))))))...
4 .....(((((((((((.(.....(((((((.....)))))))))...
5 ((((.(((.(((.....(((((((.....)))))))))...
6 .(((((((((((.(.....(((((((.....)))))))))...
7 .....((.(((.(((.....(((((((.....)))))))))...
8 ...(((((((((((.(.....(((((((.....)))))))))...

          mfess_enforce
0 ((((((.(....(((((((((((.....)))))))))...
1 ((((((((((((((.(.....(((((((((((.....)))))))))...
2          error
3          error
4 (((....(((.(((.....(((((((((((.....)))))))))...
5 ((((.(((.(((.....(((((((((((.....)))))))))...
6 .(((((((((((((((.(.....(((((((((((.....)))))))))...
7          error
8 ...(((((((((((((((.(.....(((((((((((.....)))))))))...

```

(continues on next page)

(continued from previous page)

```

                                                    seq
0  GCGCGGAAACAAUGAUGAAUGGGUUUAAAUUGGGCACUUGACUCAU...
1  CUGUCGAAGAGACGCGAUGAAUCCCGCCUGUAAUUCGGGCACCUC...
2  AAUCAAUAGGGAAGCAACGAAGCAUAGCCUUUAUUGGACACUUGG...
3  AAAUAUUUAGAGAUGUUGAAGUAUAUUCUAUUUUGGGCACCUUA...
4  AUUUUAAGAGGAAAUUUUGAACUAUAUACUUAUUUGGGCACUUGU...
5  UGCAAUGGGUGUGAUGAAGUCCGGACAGUAAUGUGGGCACUUGUC...
6  AAUAUUUUAGAAACUGAGAAGUAUAUCUUAUUUUGGGCAUCUGGA...
7  AUAACGGCACGAAGCAAUGAAAUGUUCGAUGUAACCGGGCACCUAU...
8  AAAUUAAGGGAAGCGUUGAGCCGCUACCCAUAUGUGGUUCACUCG...

                                                    SS
0  ((((((.((((.....(((((((.....)))))))))...
1  (((.(((((((.(.....(((((((.....)))))))))...
2  ....(.(((.(((.....(((((((.....)))))))))...
3  .(...(((((((.(.....(((((((.....)))))))))...
4  (((....(((.(((.....(((((((.....)))))))))...
5  ((((.(((.(.....(((((((.....)))))))))...
6  .(((((((.(((.....(((((((.....)))))))))...
7  ...(.((.(.(((.....(((((((.....)))))))))...
8  ...(((((((.(((.....(((((((.....)))))))))...

[9 rows x 26 columns]

```

```

usage: rna_align_foldability.py [-h] [--all-stars] [--dev] [--skip-mcfold]
                                  [-v]
                                  file output

```

file

an alignment in the Stockholm format, the first seq will be used to calculate distance to (#TODO pick any seq)

output

csv pandas file

-h, --help

show this help message and exit

--all-stars

this takes usully super long

--dev

--skip-mcfold

-v, --verbose

be verbose

4.4.3 Random assignment of nucleotides

4.4.3.1 random_assignment_of_nucleotides.py

random_assignment_of_nucleotides.py - Random assignment of nucleotides for non-typical characters in the sequence alignment (arg -alignfn or fasta file with seqneces (arg -seqfn)

```
R = G A (purine)
Y = U C (pyrimidine)
K = G U (keto)
M = A C (amino)
S = G C (strong bonds)
W = A U (weak bonds)
B = G U C (all but A)
D = G A U (all but C)
H = A C U (all but G)
V = G C A (all but T)
N = A G C U (any)
```

author: A. Zyla - azyla

Warning

Tested only on fasta files! and requires Biopython (tested with v1.68)

```
usage: random_assignment_of_nucleotides.py [-h] [-v] [--alignfn ALIGNFN]
                                           [--seqfn SEQFN] [--outfn OUTFN]
```

-h, --help

show this help message and exit

-v, --verbose

increase output verbosity

--alignfn <alignfn>

alignment in the Fasta format

--seqfn <seqfn>

sequences in the Fasta format

--outfn <outfn>

output aln file (default: alfn.fasta -> _out.fasta)

random_assignment_of_nucleotides.py - Random assignment of nucleotides for non-typical characters in the sequence alignment (arg -alignfn or fasta file with seqneces (arg -seqfn)

```
R = G A (purine)
Y = U C (pyrimidine)
K = G U (keto)
M = A C (amino)
S = G C (strong bonds)
W = A U (weak bonds)
B = G U C (all but A)
```

(continues on next page)

```
D = G A U (all but C)
H = A C U (all but G)
V = G C A (all but T)
N = A G C U (any)
```

author: A. Zyla - azyla

Warning

Tested only on fasta files! and requires Biopython (tested with v1.68)

`rna_tools.tools.rna_alignment.random_assignment_of_nucleotides.get_align(alignfn)`

Get seq from an alignment with gaps.

Args:

`alignfn` (str): a path to an alignment

Usage::

```
>>> get_align('test_data/aln1.fasta')
SingleLetterAlphabet() alignment with 2 rows and 13 columns
AGGGGGACAGNYU 1
CYGA-----CGG 2
```

`obj1`, `description='obj1'`, `dbxrefs=[]`, `id='<unknown id>'`, `name='<unknown name>'`,
`description='<unknown description>'`, `dbxrefs=[]`

Returns:

alignment

`rna_tools.tools.rna_alignment.random_assignment_of_nucleotides.get_parser()`

`rna_tools.tools.rna_alignment.random_assignment_of_nucleotides.get_sequences(seqfn)`

Get seq from a fasta file. :param `seqfn`: a path to a fasta file :type `seqfn`: str

Usage::

```
>>> get_align('test_data/fasta.fasta')
```

Returns

[SeqRecord(seq=Seq('GGGYYGCCNRW', SingleLetterAlphabet()), id='1', name='1', description='1', dbxrefs=[]), SeqRecord(seq=Seq('GGRGYYGCCUURWAA', SingleLetterAlphabet()), id='1', name='1', description='1', dbxrefs=[])]

`rna_tools.tools.rna_alignment.random_assignment_of_nucleotides.write_align(align, outfn)`

Write cleaned alignment with Biopython. :param `align`: a cleaned alignment :type `align`: obj :param `outfn`: a path to a new alignment file :type `outfn`: str

Returns

writes to a file in fasta format

Return type

none

`rna_tools.tools.rna_alignment.random_assignment_of_nucleotides.write_seq(seqfn, outfn)`

Write cleaned alignment with Biopython. :param align: a cleaned alignment :type align: obj :param outfn: a path to a new alignment file :type outfn: str

Returns

writes to a file in fasta format

Return type

none

4.4.4 CMAalign

`class rna_tools.tools.rna_alignment.rna_alignment.CMAalign(outputfn=None)`

CMAalign class around cmaalign (of Infernal).

cmaalign - aligns the RNA sequences in <seqfile> to the covariance model (CM) in <cmfile>. The new alignment is output to stdout in Stockholm format.

Example:

```

cma = ra.CMAalign()
cma.run_cmaalign("ade_seq.fa", "RF00167.cm")
seq = cma.get_seq()
print 'cma hit ', seq
print 'seq      ', a.align_seq(seq)
print 'a.rf     ', a.rf

cmd cmaalign -g RF00167.cm ade_seq.fa

# STOCKHOLM 1.0
#=GF AU Infernal 1.1.2

ade      -----CGCUUCAUAUAAUCCUAAUGAUUUGGUUUGGGAGUUUCUACCAAGAG-
->CCUUAAA-CUCUUGAUUAUGAAGUGA-----
#=GR ade PP .....99*****.
->*****.*****999.....
#=GC SS_cons ::::::::::::::((((((((, ,<<<<<<----->>>>>>,,,,,,<<<<<<-----
->>>>>>>,,))))):::
#=GC RF      ▭
->aaaaaaaaaaaaaaaauucccuCgUAUAUaucccggaAAUAUGGcccgggaGUUUCUACCaggcagCCGUAAAcugccuGACUAcGagggaauuuuuuuu
//
cma hit  -----CGCUUCAUAUAAUCCUAAUGAUUUGGUUUGGGAGUUUCUACCAAGAG-CCUUAAA-
->CUCUUGAUUAUGAAGUGA-----
seq      -----CGCU-U-CAUAUAUCCUAAUGAUUUGG-UUUGGGA-GUUUCUACCAAGAG-CC--
->UUAAA-CUCUU--GAUUAUG-AAGUGA-----
a.rf     aaaaaaaaaaaaaaaaauuccc.u.CgUAUAUaucccggaAAUAUGG.cccggga.GUUUCUACCaggcagCC..
->GUAAAcugccu..GACUAcG.agggaaauuuuuuuuuuu.

```

Install <http://eddylib.org/infernal/>

Cite: Nawrocki and S. R. Eddy, Infernal 1.1: 100-fold faster RNA homology searches, Bioinformatics 29:2933-2935 (2013).

`get_gc_rf()`

Get #=GC RF.

Variables

self.output – string

get_seq()

Variables

self.output – output of `cmalign`, string

run_cmalign(*seq, cm, verbose=True*)

Run `cmalign` and process the result.


Parameters

- **seq** – seq string
- **cm** – cm fn

Run:

```
$ cmalign RF01831.cm 4lvv.seq
# STOCKHOLM 1.0
#=GF AU Infernal 1.1.2

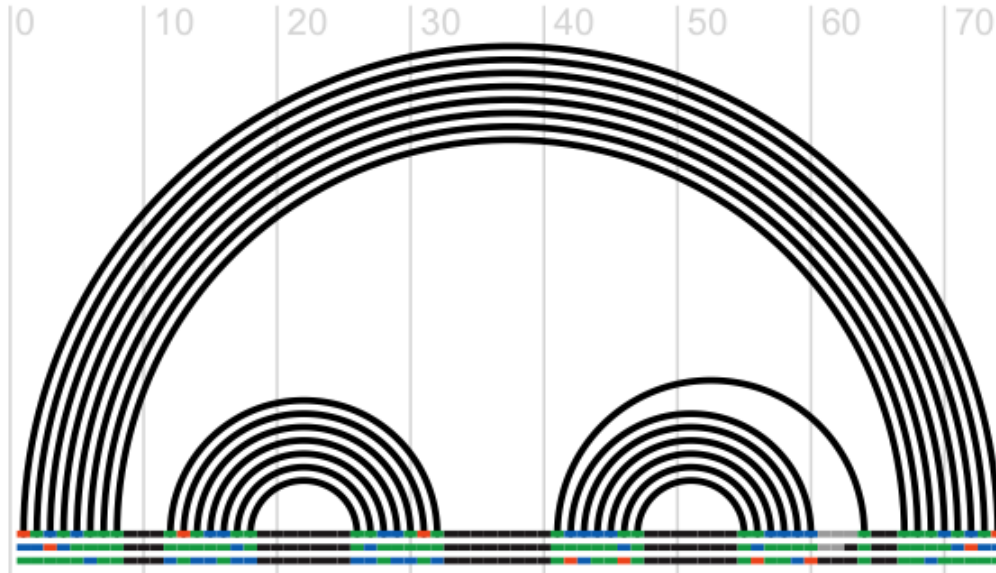
4lvv      -GGAGAGUA-GAUGAUUCGCGUUAAGUGUGUGUGA-AUGGGAUGUCG-UCACACAACGAAGC---
->GAGA---GCGCGGUGAAUCAUU-GCAUCCGCUCCA
#=GR 4lvv PP .*****.*****9999998.*****.8999999*****8...
->5555..8*****.*****
#=GC SS_cons ((((-(((((((((, , , , <<-<<<<<<<_____>>>>>>>>), , <<<<____
->____>>>>, , ,)))))))))))------))))))
#=GC RF      <
->ggcaGAGUAGggugccgugcGUuAAGUGccggcgggAcGGGgaGUUGcccgccggACGAaggcaaaauugcccGCGguacggcaccCGCAU
//
```

 **Warning**
requires `cmalign` to be set in your shell

4.4.5 RChie

`class rna_tools.tools.rna_alignment.rna_alignment.RChie`

RChie - plotting arc diagrams of RNA secondary structures.



www.e-rna.org

<http://www.e-rna.org/r-chie/>

The offline version of R-chie, which requires first installing R4RNA is available [here](#), or clone our git repository [here](#) How to install it:

- Ensure R is installed already, or download it freely from <http://www.r-project.org/>
- Download the R4RNA (<https://github.com/jujubix/r-chie>), open R and install the package:

```
install.packages("<path_to_file>/R4RNA", repos = NULL, type="source")
# Install the optparse and RColorBrewer
install.packages('optparse')
install.packages('RColorBrewer')
```

- Go to `rna_tools/rna_tools_config_local.py` and set `RCHIE_PATH` to the folder with RChie, e.g. `"/home/magnus/work/opt/r-chie/"`.

To test if Rchie works on your machine (from `rna_align` folder):

```
<path to your rchie>/rchie.R --msafile test_data/rchie_test_files/fasta.txt test_
↪data/rchie_test_files/helix.txt
```

you should have `rchie.png` file in the folder.

More at <http://www.e-rna.org/r-chie/download.cgi>

Cite: Daniel Lai, Jeff R. Proctor, Jing Yun A. Zhu, and Irmtraud M. Meyer (2012) R-chie: a web server and R package for visualizing RNA secondary structures. *Nucleic Acids Research*, first published online March 19, 2012. doi:10.1093/nar/gks241

plot_cov(*seqs*, *ss_cons*, *plot_fn*='rchie.png', *verbose*=False)

Plot an RChie plot_conv.

Parameters

- **seqs** – seqs in the fasta format
- **ss_cons** – a string of secondary structure consensus, use only ().. Works with pseudo-knots.

show()

write(*outfn*)

4.4.6 Renumber a pdb file according to alignment

4.4.6.1 renum_to_aln.py

renum_pdb_to_aln.py - renumber a pdb file based on the alignment.

author: A. Zyla under supervision of mmagnus

 **Warning**

works only for single chain! and requires Biopython (tested with v1.68)

```
usage: renum_to_aln.py [-h] [-v] [--residue_index_start RESIDUE_INDEX_START]
                    [--outfn OUTFN]
                    seqid alignfn pdbfn
```

seqid

seq id in the alignemnt

alignfn

alignemnt in the Fasta format

pdbfn

pdb file

-h, --help

show this help message and exit

-v, --verbose

increase output verbosity

--residue_index_start <residue_index_start>

renumber starting number (default: 1)

--outfn <outfn>

output pdb file (default: pdbfn .pdb -> _out.pdb)

renum_pdb_to_aln.py - renumber a pdb file based on the alignment.

author: A. Zyla under supervision of mmagnus

Warning

works only for single chain! and requires Biopython (tested with v1.68)

`rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln.get_parser()`

`rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln.get_seq(alignfn, seqid)`

Get seq from an alignment with gaps.

Parameters

- **alignfn** (*str*) – a path to an alignment
- **seqid** (*str*) – seq id in an alignment

Usage:

```
>>> get_seq('test_data/ALN_OBJ1_OBJ2.fa', 'obj1')
SeqRecord(seq=SeqRecord(seq=Seq('GUUCAG-----UGAC-',
↳SingleLetterAlphabet()), id='obj1', name='obj1', description='obj1', dbxrefs=[]),
↳id='<unknown id>', name='<unknown name>', description='<unknown description>',
↳dbxrefs=[])
```

Returns

SeqRecord

`rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln.open_pdb(pdbfn)`

Open pdb with Biopython.

Parameters

pdbfn (*str*) – a path to a pdb structure

Returns

with a pdb structure

Return type

PDB Biopython object

`rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln.renumber(seq_with_gaps, struc,
residue_index_start)`

Renumber a pdb file.

Parameters

- **seq_with_gaps** (*str*) – a target sequence extracted from the alignment
- **struc** (*pdb*) – a structure
- **residue_index_start** (*int*) – starting number

Returns

BioPython Structure object

`rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln.write_struc(struc, outfn)`

Write renumbered pdb with Biopython.

Parameters

- **struc** (*pdb*) – a renumbered structure

- **outfn** (*str*) – a path to a new, renumbered pdb file

Returns

writes to a file

Return type

none

4.5 Root Mean Square Deviation (RMSD)

4.5.1 rna_calc_rmsd

4.5.1.1 rna_calc_rmsd

rna_calc_rmsd.py - calculate RMSDs of structures to the target

If you still have problem with various number of atoms, check out this issue: [get_rnapuzzle_ready: how to deal with `Alternate location indicator`](https://github.com/mmagnus/rna-pdb-tools/issues/30) (<https://github.com/mmagnus/rna-pdb-tools/issues/30>).

The program is using (<https://github.com/charnley/rmsd>).

Example #1:

```
$ rna_calc_rmsd.py -t 6_0_solution_4GXY_rpr.pdb --model-selection=A:1-17+24-110+115-168_
↳ *.pdb
rmsd_calc_rmsd_to_target
-----
method: all-atom-built-in
# of models: 35
6_0_solution_4GXY_rpr.pdb 0.0 3409
6_Blanchet_1_rpr.pdb 22.31 3409
6_Blanchet_2_rpr.pdb 21.76 3409
6_Blanchet_3_rpr.pdb 21.32 3409
6_Blanchet_4_rpr.pdb 22.22 3409
6_Blanchet_5_rpr.pdb 24.17 3409
6_Blanchet_6_rpr.pdb 23.28 3409
6_Blanchet_7_rpr.pdb 22.26 3409
6_Bujnicki_1_rpr.pdb 36.95 3409
6_Bujnicki_2_rpr.pdb 30.9 3409
6_Bujnicki_3_rpr.pdb 32.1 3409
6_Bujnicki_4_rpr.pdb 32.04 3409
...
```

Example #2:

```
time rmsd_calc_to_target.py
-t 5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
--target-selection A:1-48+52-63
--model-selection A:1-48+52-63
--target-ignore-selection A/57/02\'
clusters/*_AA.pdb

rmsd_calc_rmsd_to_target
-----
```

(continues on next page)

(continued from previous page)

```

target_selection:  A:1-48+52-63
model_selection:   A:1-48+52-63
target_ignore_selection:  A/57/02'
model_ignore_selection:
# of models: 801
fn,rmsd_all
pistol_thrs0.50A_clust01-000001_AA.pdb,7.596
pistol_thrs0.50A_clust02-000001_AA.pdb,7.766
pistol_thrs0.50A_clust03-000001_AA.pdb,18.171
[.]
pistol_thrs0.50A_clust799-000001_AA.pdb,5.356
pistol_thrs0.50A_clust800-000001_AA.pdb,15.282
pistol_thrs0.50A_clust801-000001_AA.pdb,16.339
# of atoms used: 1237
csv was created!  rmsds.csv
rmsd_calc_to_target.py -t 5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
37.93s user 1.07s system 87% cpu 44.650 total

```

Works also for multiple chains:

```

rna_calc_rmsd.py      --model-selection='A:52+53+59+60+61+80+B:21+22+23'      --target-
selection='A:52+53+59+60+61+80+B:21+22+23' -t yC_5LJ3_U2U6_core_mdrFx_onlyTriplex_rpr.pdb
yC_5LJ3_U2U6_core_mdrFx_addh_MD_1_rpr_rchain.pdb

```

```

usage: rna_calc_rmsd [-h] -t TARGET_FN [--ignore-files IGNORE_FILES]
                  [--target-selection TARGET_SELECTION]
                  [--target-ignore-selection TARGET_IGNORE_SELECTION]
                  [--model-selection MODEL_SELECTION]
                  [--model-ignore-selection MODEL_IGNORE_SELECTION]
                  [-m METHOD] [-o RMSDS_FN] [-v] [-pr] [-sr] [-pp]
                  [--way WAY] [--name-rmsd-column NAME_RMSD_COLUMN]
                  [--target-column-name]
                  files [files ...]

```

files

files

-h, --help

show this help message and exit

-t <target_fn>, **--target-fn** <target_fn>

pdb file

--ignore-files <ignore_files>

files to be ignored, e.g, 'solution'

--target-selection <target_selection>

selection, e.g. A:10-16+20, where #16 residue is included

--target-ignore-selection <target_ignore_selection>

A/10/O2'

--model-selection <model_selection>

selection, e.g. A:10-16+20, where #16 residue is included

```

--model-ignore-selection <model_ignore_selection>
    A/10/O2'
-m <method>, --method <method>
    align, fit
-o <rmsds_fn>, --rmsds-fn <rmsds_fn>
    ouput, matrix
-v, --verbose
    verbose
-pr, --print-results
-sr, --sort-results
-pp, --print-progress
--way <way>
    R|c|p = C1' backbone = P OP1 OP2 O5' C5' C4' C3' O3' po = P OP1 OP2 no-backbone = all - po bases,
    backbone+sugar, sugar pooo = P OP1 OP2 O5' alpha = P OP1 OP2 O5' C5'
--name-rmsd-column <name_rmsd_column>
    default: fn,rmsd, with this cols will be fn,<name-rmsd-column>

```

--target-column-name

rna_calc_rmsd.py - calculate RMSDs of structures to the target

If you still have problem with various number of atoms, check out this issue: [get_rnapuzzle_ready: how to deal with \Alternate location indicator \(https://github.com/mmagnus/rna-pdb-tools/issues/30\)](https://github.com/mmagnus/rna-pdb-tools/issues/30).

The program is using (<https://github.com/charnley/rmsd>).

Example #1:

```

$ rna_calc_rmsd.py -t 6_0_solution_4GXY_rpr.pdb --model-selection=A:1-17+24-110+115-168.
↪*.pdb
rmsd_calc_rmsd_to_target
-----
method: all-atom-built-in
# of models: 35
6_0_solution_4GXY_rpr.pdb 0.0 3409
6_Blanchet_1_rpr.pdb 22.31 3409
6_Blanchet_2_rpr.pdb 21.76 3409
6_Blanchet_3_rpr.pdb 21.32 3409
6_Blanchet_4_rpr.pdb 22.22 3409
6_Blanchet_5_rpr.pdb 24.17 3409
6_Blanchet_6_rpr.pdb 23.28 3409
6_Blanchet_7_rpr.pdb 22.26 3409
6_Bujnicki_1_rpr.pdb 36.95 3409
6_Bujnicki_2_rpr.pdb 30.9 3409
6_Bujnicki_3_rpr.pdb 32.1 3409
6_Bujnicki_4_rpr.pdb 32.04 3409
...

```

Example #2:

```

time rmsd_calc_to_target.py
-t 5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
--target-selection A:1-48+52-63
--model-selection A:1-48+52-63
--target-ignore-selection A/57/02\
clusters/*_AA.pdb

rmsd_calc_rmsd_to_target
-----
target_selection:  A:1-48+52-63
model_selection:   A:1-48+52-63
target_ignore_selection:  A/57/02'
model_ignore_selection:
# of models: 801
fn,rmsd_all
pistol_thrs0.50A_clust01-000001_AA.pdb,7.596
pistol_thrs0.50A_clust02-000001_AA.pdb,7.766
pistol_thrs0.50A_clust03-000001_AA.pdb,18.171
[.]
pistol_thrs0.50A_clust799-000001_AA.pdb,5.356
pistol_thrs0.50A_clust800-000001_AA.pdb,15.282
pistol_thrs0.50A_clust801-000001_AA.pdb,16.339
# of atoms used: 1237
csv was created!  rmsds.csv
rmsd_calc_to_target.py -t 5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
37.93s user 1.07s system 87% cpu 44.650 total

```

Works also for multiple chains:

```

rna_calc_rmsd.py      -model-selection='A:52+53+59+60+61+80+B:21+22+23'      -target-
selection='A:52+53+59+60+61+80+B:21+22+23' -t yC_5LJ3_U2U6_core_mdrFx_onlyTriplex_rpr.pdb
yC_5LJ3_U2U6_core_mdrFx_addh_MD_1_rpr_rchain.pdb

```

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd.calc_rmsd(a, b, target_selection, target_ignore_selection, model_selection, model_ignore_selection, way, verbose)`

Calculate RMSD between two XYZ files

by: Jimmy Charnley Kromann <jimmy@charnley.dk> and Lars Andersen Bratholm <larsbratholm@gmail.com>
project: <https://github.com/charnley/rmsd> license: <https://github.com/charnley/rmsd/blob/master/LICENSE>

a is model b is target

Params

a = filename of structure a

Params

b = filename of structure b

Returns

rmsd, number of atoms

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd.calc_rmsd_pymol(pdb1, pdb2, method)`

Calculate rmsd using PyMOL. Two methods are available: align and fit

See:

- Align: <<http://www.pymolwiki.org/index.php/Align>>

- Fit: <<http://www.pymolwiki.org/index.php/Fit>>

Align can return a list with 7 items:

- RMSD after refinement
- Number of aligned atoms after refinement
- Number of refinement cycles
- RMSD before refinement
- Number of aligned atoms before refinement
- Raw alignment score
- Number of residues aligned

in this version of function, the function returns *RMSD before refinement*.

Install on OSX: `brew install brewsci/bio/pymol` or get

If you have a problem:

```
Match-Error: unable to open matrix file '/opt/local/Library/Frameworks/Python.
↳framework/Versions/2.7/lib/python2.7/site-packages/data/pymol/matrices/BLOSUM62'.
```

then find BLOSUM62, e.g.:

```
mdfind -name BLOSUM62 | grep pymol
/Users/magnus/miniconda2/envs/py37/lib/python3.7/site-packages/pymol/pymol_path/
↳data/pymol/matrices/BLOSUM62
/usr/local/Cellar/pymol/2.4.0_3/libexec/lib/python3.9/site-packages/pymol/pymol_
↳path/data/pymol/matrices/BLOSUM62
/Users/magnus/miniconda2/pkgs/pymol-2.4.2-py37h06d7bae_0/share/pymol/data/pymol/
↳matrices/BLOSUM62
/Users/magnus/work/opt/pymol-open-source/data/pymol/matrices/BLOSUM62
```

and then define PYMOL_DATA in your `.bashrc/.zshrc`, e.g.:

```
export PYMOL_DATA="/Users/magnus/work/opt/pymol-open-source/data/pymol"
```

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd.get_parser()`

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd.get_rna_models_from_dir(files)`

Parameters

models – a list of filenames

Example of the list:

```
['test_data/rp17/2_restr1_Michal1.pdb_clean.pdb', 'test_data/rp17/2a_nonrestr2_
↳Michal1.pdb_clean.pdb',
'test_data/rp17/3_nonrestr1_Michal1.pdb_clean.pdb', 'test_data/rp17/5_restr1_
↳Michal3.pdb_clean.pdb']
```

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd.sort_nicely(l)`

Sort the given list in the way that humans expect.

<http://blog.codinghorror.com/sorting-for-humans-natural-sort-order/>

4.5.1.2 rna_calc_rmsd_multi_targets.py

rna_calc_rmsd_multi_targets.py - calculate RMSDs of structures to multiple targets:

```
$ rna_calc_rmsd_multi_targets.py --models multi-targets/rp21/*.pdb
↪      --targets multi-targets/rp21/solutions/*.pdb
↪      --target-selection A:1-27+29-41
↪selection A:1-27+29-41
```

CSV table produced:

```

                21_solution_0_ChainA.pdb  21_solution_0_ChainB.pdb  21_
↪solution_1_ChainA.pdb  21_solution_1_ChainB.pdb  21_solution_2.pdb  mean  min
↪max    sd
fn
21_3dRNA_1_rpr.pdb          12.17          12.11          12.11  12.13  12.11  12.17  0.03
↪      12.17          12.11          12.11  12.13  12.11  12.17  0.03
21_Adamiak_1_rpr.pdb        4.64          4.61          4.64  4.63  4.61  4.64  0.01
↪      4.64          4.61          4.64  4.63  4.61  4.64  0.01
21_ChenHighLig_1_rpr.pdb    4.01          3.97          4.07  4.01  3.97  4.07  0.04
↪      4.01          3.97          4.07  4.01  3.97  4.07  0.04
21_Das_1_rpr.pdb           5.71          5.60          5.61  5.65  5.60  5.71  0.05
↪      5.71          5.60          5.61  5.65  5.60  5.71  0.05
-----
Save rna_calc_rmsd_multi_targets_output.csv
```

```
usage: rna_calc_rmsd_multi_targets.py [-h] [-v] [--models MODELS [MODELS ...]]
                                     [--targets TARGETS [TARGETS ...]]
                                     [--output-csv OUTPUT_CSV]
                                     [--model-selection MODEL_SELECTION]
                                     [--target-selection TARGET_SELECTION]
```

-h, --help

show this help message and exit

-v, --verbose

be verbose

--models <models>

--targets <targets>

--output-csv <output_csv>

--model-selection <model_selection>

selection, e.g. A:10-16+20, where #16 residue is included

--target-selection <target_selection>

selection, e.g. A:10-16+20, where #16 residue is included

4.5.2 rna_calc_rmsd_trafl

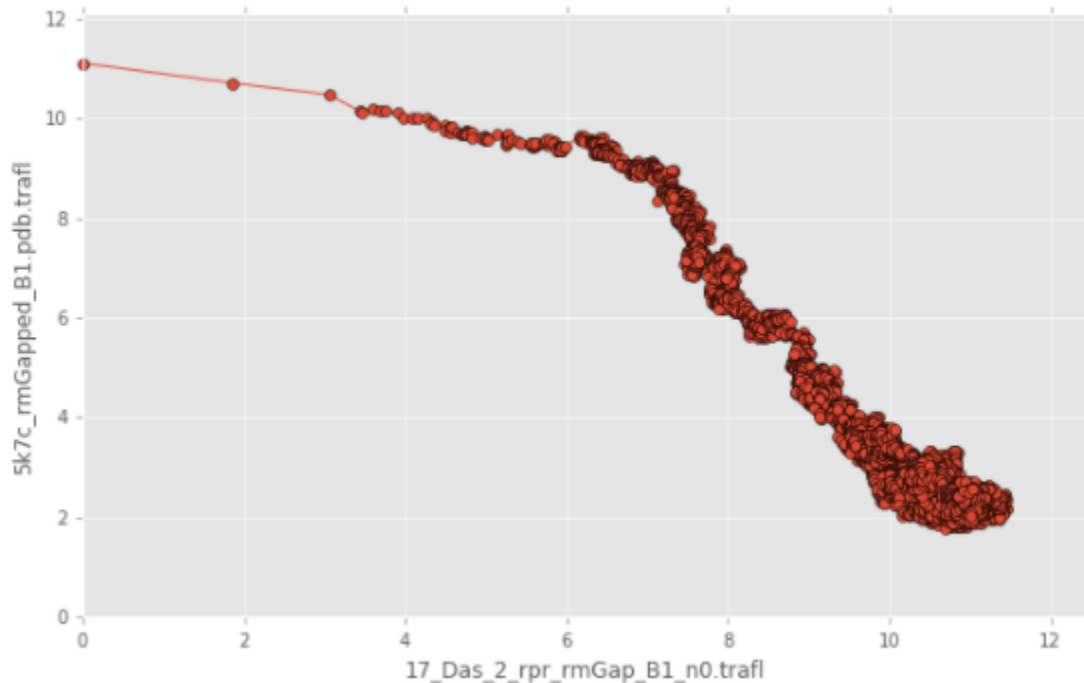
4.5.2.1 rna_calc_evo_rmsd

rmsd_calc_trafl - calculate RMSD of transition A->B based on a SimRNA trajectory

After this script, run:

```
rna_cal_rmsd_trafl_plot.py rmsd.txt
```

to get a plot like this:



Prepare structures:

```
$ SimRNA -p 17_Das_2_rpr.pdb -n 0 -o 17_Das_2_rpr_n0 # no trafl, trafl will be added
$ SimRNA -p 5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped.pdb -n 0 -o 5k7c_clean_
↪onechain_renumber_as_puzzle_rpr_rmGapped
#(struc must be (~CG~) nope. It has to be a trajectory!)
```

and run:

```
$ rmsd_calc_trafl.py 17_Das_2_rpr.pdb.trafl 17_Das_2_rpr_n0.trafl 5k7c_clean_onechain_
↪renumber_as_puzzle_rpr_rmGapped_n0.trafl rp17_rmsd.txt
> calc_rmsd_to_1st_frame
  /Users/magnus/work/opt/simrna/SimRNA_64bitIntel_MacOSX_staticLibs/calc_rmsd_to_1st_
↪frame 17_Das_2_rpr.pdb.trafl 17_Das_2_rpr.pdb_rmsd_e
< rmsd_out: 17_Das_2_rpr.pdb_rmsd_e
> struc: 17_Das_2_rpr_n0.trafl 2
> trafl: 17_Das_2_rpr.pdb.trafl 48
% saved: 17_Das_2_rpr.pdb.trafl_17_Das_2_rpr_n0.trafl
> calc_rmsd_to_1st_frame
```

(continues on next page)

(continued from previous page)

```

/Users/magnus/work/opt/simrna/SimRNA_64bitIntel_MacOSX_staticLibs/calc_rmsd_to_1st_
↪ frame 17_Das_2_rpr.pdb.trafl_17_Das_2_rpr_n0.trafl 17_Das_2_rpr.pdb_rmsd_e_17_Das_2_
↪ rpr_n0_rmsd_e
< rmsd_out: 17_Das_2_rpr.pdb_rmsd_e_17_Das_2_rpr_n0_rmsd_e
> struc: 5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped_n0.trafl 2
> trafl: 17_Das_2_rpr.pdb.trafl 48
% saved: 17_Das_2_rpr.pdb.trafl_5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped_n0.
↪ trafl
> calc_rmsd_to_1st_frame
  /Users/magnus/work/opt/simrna/SimRNA_64bitIntel_MacOSX_staticLibs/calc_rmsd_to_1st_
↪ frame 17_Das_2_rpr.pdb.trafl_5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped_n0.
↪ trafl 17_Das_2_rpr.pdb_rmsd_e_5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped_n0_
↪ rmsd_e
< rmsd_out: 17_Das_2_rpr.pdb_rmsd_e_5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped_
↪ n0_rmsd_e
  0.000 -695.634
  0.000 -551.093
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
< out: rp17_rmsd.txt

```

⚠ Warning

calc_rmsd_to_1st_frame (SimRNA) is required and the path to the binary file is defined in config_local.

```
usage: rna_calc_evo_rmsd [-h] trafl struc1 struc2 rmsds_fn
```

trafl

trafl

struc1

structure A

struc2

structure B

rmsds_fn

output file

-h, --help

show this help message and exit

4.5.2.2 rna_cal_rmsd_trafl_plot

rna_cal_rmsd_trafl_plot - generate a plot based of <rmsd.txt> of rna_calc_evo_rmsd.py.

```
usage: rna_cal_rmsd_trafl_plot [-h] file
```

file

rmsd.txt

-h, --help

show this help message and exit

4.5.3 rna_calc_rmsd_all_vs_all

4.5.3.1 rna_calc_rmsd_all_vs_all

rna_calc_rmsd_all_vs_all.py - calculate RMSDs all vs all and save it to a matrix

Examples:

```
rna_calc_rmsd_all_vs_all.py -i test_data -o test_output/rmsd_calc_dir.tsv
# of models: 4
... 1 test_data/struc1.pdb
... 2 test_data/struc2.pdb
... 3 test_data/struc3.pdb
... 4 test_data/struc4.pdb
```

The program is using (<https://github.com/charnley/rmsd>).

You can also use PyMOL to do align or fit:

```
python rna_calc_rmsd_all_vs_all.py -i test_data -o test_output/rmsd_calc_dir_align.mat -
-m align
# of models: 5
# test_data/2nd_triplex_FB_1AUA3_rpr.pdb test_data/struc1.pdb test_data/struc2.pdb test_
data/struc3.pdb test_data/struc4.pdb
0.0 4.13 4.922 4.358 4.368
4.13 0.0 11.092 4.707 3.46
4.922 11.092 0.0 11.609 11.785
4.358 4.707 11.609 0.0 2.759
4.368 3.46 11.785 2.759 0.0
matrix was created! test
```

```
usage: rna_calc_rmsd_all_vs_all [-h] [-i INPUT_DIR] [-o MATRIX_FN] [-m METHOD]
```

-h, --help

show this help message and exit

-i <input_dir>, **--input-dir** <input_dir>

input folder with structures

-o <matrix_fn>, **--matrix-fn** <matrix_fn>

output, matrix

-m <method>, **--method** <method>

all-atom, pymol: align, fit

rna_calc_rmsd_all_vs_all.py - calculate RMSDs all vs all and save it to a matrix

Examples:

```
rna_calc_rmsd_all_vs_all.py -i test_data -o test_output/rmsd_calc_dir.tsv
# of models: 4
... 1 test_data/struc1.pdb
... 2 test_data/struc2.pdb
... 3 test_data/struc3.pdb
... 4 test_data/struc4.pdb
```

The program is using (<https://github.com/charnley/rmsd>).

You can also use PyMOL to do align or fit:

```
python rna_calc_rmsd_all_vs_all.py -i test_data -o test_output/rmsd_calc_dir_align.mat -
↪m align
# of models: 5
# test_data/2nd_triplex_FB_1AUA3_rpr.pdb test_data/struc1.pdb test_data/struc2.pdb test_
↪data/struc3.pdb test_data/struc4.pdb
0.0 4.13 4.922 4.358 4.368
4.13 0.0 11.092 4.707 3.46
4.922 11.092 0.0 11.609 11.785
4.358 4.707 11.609 0.0 2.759
4.368 3.46 11.785 2.759 0.0
matrix was created! test
```

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all.calc_rmsd(a, b)`

Calc rmsd.

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all.get_parser()`

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all.get_rna_models_from_dir(directory)`

`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all.sort_nicely(l)`

Sort the given list in the way that humans expect. <http://blog.codinghorror.com/sorting-for-humans-natural-sort-order/>

4.6 Interaction Network Fidelity (INF)

4.6.1 rna_calc_inf.py

```
usage: rna_calc_inf.py [-h] [-t TARGET_FN] [-m NT]
                    [--ignore-files IGNORE_FILES] [-s SS] [--no-stacking]
                    [--debug] [--web] [-pr] [-sr] [--method METHOD]
                    [--target-selection TARGET_SELECTION]
                    [--model-selection MODEL_SELECTION]
                    [--renumber-residues] [--dont-remove-sel-files] [-f]
                    [-v] [-o OUT_FN]
                    files [files ...]
```

files

files, .e.g folder_with_pdbs/*pdbs

-h, --help

show this help message and exit

-t <target_fn>, **--target_fn** <target_fn>

pdb file

-m <nt>, **--number-of-threads** <nt>

number of threads used for multiprocessing, if 1 then mp is not used (useful for debugging)!

--ignore-files <ignore_files>
files to be ingored, .e.g, 'solution'

-s <ss>, **--ss** <ss>
A:([[]]), works only for single chain (the chain is A by default)

--no-stacking
default: use stacking, if this option on, don't take into account stacking,
WARNING/BUG: inf_all will be incorrectly calculated if stacking is off

--debug

--web

-pr, **--print-results**

-sr, **--sort-results**

--method <method>
you can use mcanotate* or clarna (right now only clarna is tested)

--target-selection <target_selection>
selection, e.g. A:10-16+20, where #16 residue is included

--model-selection <model_selection>
selection, e.g. A:10-16+20, where #16 residue is included

--renumber-residues
renumber residues from 1 to X for comparison with selection

--dont-remove-sel-files
don't remove temp files created based on target|model-selectionforce

-f, **--force**
force to run ClaRNA even if <pdb>.outCR file is there, for will be auto True when selection defined

-v, **--verbose**
be verbose, tell me more what're doing

-o <out_fn>, **--out_fn** <out_fn>
out csv file, be default *inf.csv*

A tool to calc inf_all, inf_stack, inf_WC, inf_nWC, SNS_WC, PPV_WC, SNS_nWC, PPV_nWC between two structures.

Mind, that ClaRNA is pretty slow, it takes even a few seconds to analyze a structure, so for, say, 1000 models you need a few hours.

How to make it faster?

First, you can use **--number-of-threads** to specify the number of cores used for multiprocessing.

Second, the procedure implemented in here is composed of two steps, first for each structure ClaRNA is used to generate an output with contacts, then these files are used for comparisons. So, if you want to re-run your analysis, you don't have to re-run ClaRNA itself. Thus, be default ClaRNA is not executed if <model>.outCR is found next to the analyzed files. To change this behavior, force (**--force**) rna_calc_inf.py to re-run ClaRNA.

`rna_tools.tools.rna_calc_inf.rna_calc_inf.do_job()`

Run ClaRNA & Compare, add 1 to the counter, write output to csv file (keeping it locked)

`rna_tools.tools.rna_calc_inf.rna_calc_inf.get_parser()`

4.7 RNA filter (DCA)

4.7.1 Calculate distances based on given restraints on PDB files or SimRNA trajectories

4.7.1.1 rna_filter.py

rna_filter.py - calculate distances based on given restraints on PDB files or SimRNA trajectories.

Changes: weight is always 1 (at least for now). .,>,<,>=<=<= .

[PREVIOUS DOCUMENTATION - TO BE REMOVED]

```
rna_filter.py -s 4gxy_rpr.pdb -r rp06_MohPairs.rfrestrs d:A5-A42 100.0 measured: 26.7465763417 [x]
d:A11-A26 100.0 measured: 19.2863696104 [x]
```

```
[mm] rp06$ git:(master) $ rna_filter.py -s 4gxy_rpr.pdb -r rp06_MohPairs.rfrestrs
d:A5-A42 100.0 measured: 26.7465763417 [x] d:A11-A26 100.0 measured: 19.2863696104 [x]
```

Traceback (most recent call last):

```
File "/home/magnus/work-src/rna-pdb-tools/bin/rna_filter.py", line 270, in <module>
    calc_scores_for_pdbstructures(args.structures, restraints, args.verbose)
```

```
File "/home/magnus/work-src/rna-pdb-tools/bin/rna_filter.py", line 221, in
calc_scores_for_pdbstructures
    dist = get_distance(residues[h[0]]['mb'], residues[h[1]]['mb'])
```

KeyError: 'A24'

correct, there is no A24 in this structure:

The format of restraints:

```
(d:A1-A2 < 10.0 1) = if distance between A1 and A2 lower than 10.0, score it with 1
```

Usage:

```
$ python rna_filter.py -r test_data/restraints.txt -s test_data/CG.pdb
d:A1-A2 10.0 measured: 6.58677550096 [x]
test_data/CG.pdb 1.0 1 out of 1

# $ python rna_filter.py -r test_data/restraints.txt -t test_data/CG.traf1
(d:A1-A2 < 10.0 1)|(d:A2-A1 <= 10 1)
restraints [('A1', 'A2', '<', '10.0', '1'), ('A2', 'A1', '<=', '10', '1')]

Frame #1 e:1252.26
mb for A1 [ 54.729 28.9375 41.421 ]
mb for A2 [ 55.3425 35.3605 42.7455]
d:A1-A2 6.58677550096
mb for A2 [ 55.3425 35.3605 42.7455]
mb for A1 [ 54.729 28.9375 41.421 ]
d:A2-A1 6.58677550096
# this ^ is off right now
```

```
usage: rna_filter.py [-h] -r RESTRAINTS_FN [-v]
                  [-s STRUCTURES [STRUCTURES ...]] [--offset OFFSET]
                  [-t TRAJECTORY]
```

-h, --help

show this help message and exit

-r <restraints_fn>, --restraints_fn <restraints_fn>

restraints_fn: Format: (d:A9-A41 < 10.0 1)|(d:A41-A9 <= 10 1)

-v, --verbose

be verbose

-s <structures>

structures

--offset <offset>

use offset to adjust your restraints to numbering in PDB files, ade (1y26)pdb starts with 13, so offset is -12)

-t <trajectory>

SimRNA trajectory

4.7.1.2 rna_dca_mapping.py

```
usage: rna_dca_mapping.py [-h] --seq SEQ --gseq GSEQ --dca DCA
                          [--offset OFFSET] [--noss] [--mss] [--verbose]
                          [--noshort]
```

-h, --help

show this help message and exit

--seq <seq>

seq fn in Fasta format

--gseq <gseq>

gapped sequence and secondary structure (like in the alignment used for DCA) in Fasta format

--dca <dca>

file with parsed interactions

--offset <offset>

offset

--noss

filter out ss from plot

--mss

ss every each line

--verbose

be verbose

--noshort

filter out short interactions, dist in seq < 6 nt

4.7.2 Show distances in PyMOL

show_dists - show distances in PyMOL

The screenshot shows the PyMOL application window. The command line at the top contains the following text:

```
exec(layer.com2+"\n",seif.pymol_names,seif.pymol_names)
File "<string>", line 1, in <module>
File "/Users/magnus/src/rna-pdb-tools/utis/rna_dca/pymol_dca_dists.py", line 5, in get_dists
cmd.distance('d'+i[0]+'-'+i[1],'(" + a + ")', (" + b + ")')
TypeError: cannot concatenate 'str' and 'int' objects
PyMOL>run /Users/magnus/src/rna-pdb-tools/utis/rna_dca/pymol_dca_dists.py
PyMOL>get_dists([[29, 41], [7, 66], [28, 42], [51, 63], [50, 64], [2, 71], [5, 68], [3, 70], [31, 39], [4, 69], [6, 67], [12, 23], [52, 62], [30, 40], [49, 65], [27, 43], [11, 24], [1, 72], [10, 25], [15, 48], [53, 61], [19, 56], [13, 22], [36, 37], [18, 19], [22, 46], [35, 73], [32, 38], [9, 13], [19, 20], [18, 20], [54, 60], [9, 23], [34, 35], [36, 38], [53, 54], [20, 56], [9, 12], [26, 44], [18, 55], [54, 61], [32, 36]])
```

The main window displays a 3D model of an RNA structure with various distances labeled. The distances are listed in a table on the right side of the window:

d1-72	A	S	H	L	C
d10-25	A	S	H	L	C
d15-48	A	S	H	L	C
d53-61	A	S	H	L	C
d19-56	A	S	H	L	C
d13-22	A	S	H	L	C
d36-37	A	S	H	L	C
d18-19	A	S	H	L	C
d22-46	A	S	H	L	C
d35-73	A	S	H	L	C
d32-38	A	S	H	L	C
d9-13	A	S	H	L	C
d19-20	A	S	H	L	C
d18-20	A	S	H	L	C
d54-60	A	S	H	L	C
d9-23	A	S	H	L	C
d34-35	A	S	H	L	C
d36-38	A	S	H	L	C
d53-54	A	S	H	L	C
d20-56	A	S	H	L	C

The bottom right corner of the window shows the Mouse Mode 3-Button Viewing controls:

```
Mouse Mode 3-Button Viewing
Buttons L M R Wheel
& Keys Rota Move MovZ Slab
Shift +Box -Box Clip MovS
Ctrl +/- PKAt PK1 MovZ
CtSh Sele Drig Clip MovZ
SnglClk +/- Cent Menu
Db1Clk Menu - PKAt
Selecting Residues
State 1/ 1
```

Usage:

```
PyMOL>show_dists([[1,2]])
1, 2, 3.41
```

Analyze an evolutionary coupling file. ~~~~~

4.7.2.1 rna_ex2x.py

rna_ex2x.py - analyze an evolutionary coupling file.

Files can be downloaded from https://marks.hms.harvard.edu/ev_rna/, e.g. RF00167.EC.interaction.csv

--pairs:

```
$ rna_ex2x.py RF00167.EC.interaction_LbyN.csv --pairs
[18, 78],[31, 39],[21, 75],[30, 40],[28, 42],[27, 43],[59, 67],[54, 72],[57, 69],[25,
↪45],[29, 41],[17, 79],[26, 44],[16, 80],[14, 82],[19, 77],[55, 71],[
    15, 81],[34, 63],[56, 70],[58, 68],[35, 63],[26, 45],[35, 64],[32, 39],[54, 73],[24,
↪74],[16, 82],[24, 45],[24, 43],[32, 36],[25, 48],[48, 82],[36, 48],
```

```
usage: rna_ec2x.py [-h] [--sep SEP] [--chain CHAIN] [--ec-pairs]
                 [--ss-pairs SS_PAIRS] [--pairs-delta]
                 interaction_fn
```

interaction_fn

interaction file

-h, --help

show this help message and exit

--sep <sep>

separator

--chain <chain>

chain

--ec-pairs

--ss-pairs <ss_pairs>

file with secondary structure base pairs

--pairs-delta

delta: ec-bp - ss-pairs

4.7.3 Convert pairs to SimRNA restraints

4.7.3.1 rna_pairs2SimRNArestrs.py

rna_pairs2SimRNArestrs.py - convert pairs to SimRNA restraints

Example:

```
$ rna_pairs2SimRNArestrs.py rp06_pairs_delta.txt -v
# of pairs: 42
SLOPE A/2/MB A/172/MB 0 6 1
SLOPE A/2/MB A/172/MB 0 7 -1
SLOPE A/3/MB A/169/MB 0 6 1
SLOPE A/3/MB A/169/MB 0 7 -1
SLOPE A/12/MB A/32/MB 0 6 1
```

```
usage: rna_pairs2SimRNArestrs.py [-h] [--offset OFFSET] [--weight WEIGHT]
                                   [--dist DIST] [--well] [-v]
                                   pairs
```

pairs

a file with [[2, 172], [3, 169], [12, 32], [13, 31]]

-h, --help

show this help message and exit

--offset <offset>

can be -10

--weight <weight>
weight

--dist <dist>
distances, for MOHCA use 25

--well
well instead of slope

-v, --verbose
be verbose

4.7.4 Get a list of base pairs for a given “fasta ss” file.

4.7.4.1 rna_ss_get_bps.py

rna_ss_get_bps.py - get a list of base pairs for a given “fasta ss” file.

Input file:

```
cat ade_ss.fa
>1y26
CGCUUCAUAUAAUCCUAAUGAUAUGGUUUGGGAGUUUCUACCAAGAGCCUAAAACUCUUGAUUAUGAAGUG
((((((((...((((((.....)))))).....((((((.....))))))..)))))))))%
```

Usage:

```
$ rna_ss_get_bps.py ade_ss.fa --offset 12
[[13, 83], [14, 82], [15, 81], [16, 80], [17, 79], [18, 78], [19, 77], [20, 76], [21, ↵
↵75], [25, 45], [26, 44], [27, 43], [28, 42], [29, 41], [30, 40], [54, 72], [55, 71], ↵
↵[56, 70], [57, 69], [58, 68], [59, 67]]
```

Now it also work with pseudoknots.

```
usage: rna_ss_get_bps.py [-h] [--offset OFFSET] [-v] file
```

file

file in the Fasta format

-h, --help
show this help message and exit

--offset <offset>
offset

-v, --verbose
be verbose

4.7.5 Get a diff of pairs

4.7.5.1 rna_pairs_diff.py

rna_pairs_diff.py - get a diff of pairs

Usage:

```
$ rna_pairs_diff.py pistol_dca_all.bp pistol.bp
# of ec_paris: 31
# of ssbps   : 18
delta#      : 13
[[4, 32], [6, 9], [6, 36], [6, 39], [9, 39], [13, 32], [16, 17], [17, 18], [22, 49], [29,
↪ 58]]
```

```
usage: rna_pairs_diff.py [-h] [-v] pairs1 pairs2
```

pairs1

a list of pairs, A

pairs2

a list of pairs to subtract, A-B, results in C(all pairs that are in A and are not in B)

-h, --help

show this help message and exit

-v, --verbose

be verbose

4.8 Contacts classification & secondary structure detection

See also PyMOL4RNA for ways to visualize edges <https://rna-tools.readthedocs.io/en/latest/pymol4rna.html>

4.8.1 3DNA (contacts classification & secondary structure detection)

Python parser to 3dna <<http://x3dna.org/>>.

Installation:

```
# install the code from http://forum.x3dna.org/downloads/3dna-download/
Create a copy of the rna_x3dna_config_local_sample.py (remove "_sample") present in rna-
↪tools/rna_tools/tools/rna_x3dna folder.
Edit this line :
BINARY_PATH = <path to your x3dna-dssr file>
matching the path with the path of your x3dna-dssr file.
e.g. in my case: BINARY_PATH = ~/bin/x3dna-dssr.bin
```

For one structure you can run this script as:

```
[mm] py3dna$ git:(master) ./rna_x3dna.py test_data/1xjr.pdb
test_data/1xjr.pdb
>1xjr nts=47 [1xjr] -- secondary structure derived by DSSR
```

(continues on next page)


```
nt,id,res,alpha,beta,gamma,delta,epsilon,zeta,e-z,chi,phase-angle,sugar-type,ssZp,Dp,splay,bpseq
1,g,A.GTP1,nan,nan,142.1,89.5,-131.0,-78.3,-53(BI),-178.2(anti),358.6(C2'-exo),~C3'-
endo,4.68,4.68,29.98,0      2,G,A,G2,-75.8,-167.0,57.2,79.5,-143.4,-69.7,-74(BI),-169.2(anti),5.8(C3'-
endo),~C3'-endo,4.68,4.76,25.61,0
```

```
run_x3dna(show_log=False, verbose=False)
```

exception rna_tools.tools.rna_x3dna.rna_x3dna.x3DNAMissingFile

4.8.2 ClaRNA (contacts classification)

If you want to calculate “Interaction Network Fidelity (INF) and not only” see also [rna_calc_inf](#)

To run ClaRNA, see the documentaton below: Usage:

```
$ rna_clarna_app.py ../../input/5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
→ ../../input/5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
((((([[[[[]]]])))).....((([[[[[]]]]]))...((((([[[[[]]]]))))...))
```

Example

```
from rna_tools.utils.clarna_app import clarna_app
if __name__ == '__main__':
    ss = '((((([[[[[]]]])))).....((([[[[[]]]]]))...((((([[[[[]]]]))))...))'
    fnCRref = clarna_app.get_ClaRNA_output_from_dot_bracket(ss)
    f = '../rna_calc_rmsd/test_data/pistol/5k7c_clean_onechain_renumber_as_puzzle_srr.pdb
→ '
    fnCR = clarna_app.clarna_run(f, force=False)
    results = clarna_app.clarna_compare(fnCRref, fnCR)
    print results #
#tmp_Z42i_..pdb.outCR      5k7c_clean_onechain_renumber_as_puzzle_srr.pdb.outCR
→0.706      NA      0.865      NA      0.842      0.889      NA      0.000
```

rna_tools.tools.clarna_app.rna_clarna_app.clarna_compare(target_cl_fn, i_cl_fn, verbose=False)

Run ClaRNA compare (from rna_tools/tools/clarna_play).

Returns

a list target, fn, scores

Scores:

```
inf_all      0.706
inf_stack -999.999 -> NA
inf_WC      0.865
inf_nWC     -999.999 -> NA
SNS_WC      0.842
PPV_WC      0.889
SNS_nWC     NA
PPV_nWC     0.000
```

Example of the list:

```
5k7c_clean_onechain_renumber_as_puzzle_srr.pdb      pistol_thrs0.50A_clust01-000001_
↪AA.pdb
0.642      NA      0.874      0.000      0.944      0.810      0.000      0.000s
```

use `results.split()[4]` to get `inf_WC`

```
rna_tools.tools.clarna_app.rna_clarna_app.clarna_run(fn, force=True, stacking=True, verbose=False)
Run ClaRNA run
```

Parameters

fn (*str*) – filename to analyze

Returns

a filename to ClaRNA output (fn + '.outCR')

Return type

str

```
rna_tools.tools.clarna_app.rna_clarna_app.get_ClaRNA_output_from_dot_bracket(ss, temp=True,
                                                                              verbose=False)
```

Get dummy ClaRNA output out of dat bracket secondary structure (ss)

Parameters

ss (*string*) – secondary structure

Returns

a filename to ClaRNA output

```
rna_tools.tools.clarna_app.rna_clarna_app.get_dot_bracket_from_ClaRNAoutput(inCR,
                                                                              verbose=False)
```

In inCR file

Warning

This function requires 'ClaRNAwd_to_vienaSS' contact marcin.magnus@icloud.com

```
rna_tools.tools.clarna_app.rna_clarna_app.get_parser()
```

4.9 RNA 3D model quality assessment

Wrappers behind the server, <http://genesilico.pl/mqapRNA> (in short, mq)

4.9.1 RASP

This module contains functions for computing RASP potential

```
class rna_tools.tools.mq.RASP.RASP.RASP(job_id=None)
```

Wrapper class for running RASP automatically.

```
colour_by_local_score(path_to_pdb, potential_type='all')
```

```
executable = ['rasp_fd', 'rasp_profile_fd']
```

```
program_name = ['rasp_fd', 'rasp_profile_fd']
```

```
run(path_to_pdb, global_energy_score=True, potentials=['c3', 'bb', 'bbr', 'all'], handler=True,
    verbose=False)
```

Compute RASP potential for a single file

rasp_fd generates:

output_all.txt:

```
-9869.61  66447  -0.148534  0  0  0
```

rasp_profile_fd generates:

profile_all.txt:

```
C      1      R      -775.038
C      2      R      -1164.22
U      3      R      -2054.17
G      4      R      -1601.13
[. .]
```

Input:

- path_to_pdb = path to PDB file
- potential_type = all, bbr or c3
- global_energy_score = True/False (See Output), default=True
- hander = True (if you use PYRO)/FALSE otherwise

Output:

- the output depends on global_energy_score value T/F You might get:
-9869.61
profile [('C', 1, 'R', -775.0380000000000001), ('C', 2, 'R', -1164.22) ..

4.9.2 Dfire

This module contains functions for computing Dfire potential

Installation:

```
git clone https://github.com/tcgriffith/dfire_rna.git make # add DFIRE_RNA_HOME to .bashrc
```

```
class rna_tools.tools.mq.Dfire.Dfire.Dfire
```

Wrapper class for Dfire.

```
run(path_to_pdb, verbose=False)
```

Run program with previously set flags. This method should work if program wants some command some options and nothing more, otherwise you should override it.

```
rna_tools.tools.mq.Dfire.Dfire.get_parser()
```

```
rna_tools.tools.mq.Dfire.Dfire.test(verbose)
```

4.9.3 RNAkb

This module contains functions for computing RNAkb potential

It seems that this is impossible to run RNAkb in full atom mode. So this works only in 5 pt (5 points/atom per residue) mode.

<https://gromacs.bioexcel.eu/t/fatal-error-an-input-file-contains-a-line-longer-than-4095-characters/1397/2>

```
class rna_tools.tools.mq.RNAkb.RNAkb.RNAkb(job_id="", sandbox=False)
```

Wrapper class for running RNAkb automatically.

```
executable = ['/usr/local/gromacs/bin/pdb2gmx', '/usr/local/gromacs/bin/make_ndx',
              '/usr/local/gromacs/bin/editconf', '/usr/local/gromacs/bin/grompp',
              '/usr/local/gromacs/bin/mdrun']
```

```
log_stdout_stderr()
```

```
run(name, potential_type, verbose=False)
```

Compute RNAkb potential for a single file

Parameters

- **path** (*str*) – name of a PDB file
- **potential_type** (*str*) – ‘5pt’ for 5 point or ‘aa’ for all atom aa is off

Returns

a list of energies (strings) ['2.57116e+05', '1.62131e+03', '7.82459e+02', '3.00789e-01', '0.00000e+00', '0.00000e+00', '-2.51238e-03', '0.00000e+00', '2.59520e+05', '2.54174e-09', '2.59520e+05']

Warning

‘aa’ does not work because of “a line longer than 4095 characters”

The function parses:

```
Statistics over 1 steps using 1 frames
Energies (kJ/mol)
Bond          Angle    Proper Dih.  Improper Dih.    LJ-14
2.44111e+05   1.76069e+03   8.12947e+02   1.82656e-01     0.00000e+00
Coulomb-14    LJ (SR)    Coulomb (SR)   Potential        Kinetic En.
0.00000e+00  -1.94624e-03  0.00000e+00   2.46685e+05     2.43227e-09
Total Energy   Temperature Pressure (bar)
2.46685e+05   6.67884e-10  -5.94232e+04
Total Virial

# ['Statistics', 'over', '1', 'steps', 'using', '1', 'frames', 'Energies', '(kJ/mol)',
  'Bond', 'Angle', 'Proper', 'Dih.', 'Improper', 'Dih.', 'LJ-14', '2.44111e+05',
  ↪ '1.76069e+03',
  '8.12947e+02', '1.82656e-01', '0.00000e+00', 'Coulomb-14', 'LJ', '(SR)',
  ↪ 'Coulomb', '(SR)',
  'Potential', 'Kinetic', 'En.', '0.00000e+00', '-1.94624e-03', '0.00000e+00', '2.
  ↪ 46685e+05',
  '2.43227e-09', 'Total', 'Energy', 'Temperature', 'Pressure', '(bar)', '2.
```

(continues on next page)

(continued from previous page)

```
↪46685e+05', '6.67884e-10',
'-5.94232e+04', 'Total', 'Virial']
result[6] is really the RNAkb
```

4.9.3.1 rna_mq_rnakb.py

Standalone tool to run the RNAkb class in the terminal.

All atom mode does not really work, see the documentation of the RNAkb class.

```
usage: rna_mq_rnakb.py [-h] [-v] [-p POTENTIAL] file [file ...]
```

file

a PDB file, one or more

-h, --help

show this help message and exit

-v, --verbose

be verbose

-p <potential>, --potential <potential>

5pt

4.9.4 QRNA

4.9.5 eSCORE

This module contains functions for computing eSCORE

Install:

```
pip install barnaba # tested with barnaba==0.1.7
```

Output:

```
$ baRNAbA ESCORE --pdb ../test/1a9n.pdb --ff /Users/magnus/work/opt/barnaba/barnaba_
↪201128/test/data/1S72.pdb
# your output will be written to files with prefix outfile.ESCORE
# KDE computed. Bandwidth= 0.25 using 10655 base-pairs# Loaded sample ../test/1a9n.pdb

#      Frame      ESCORE
#          0      4.1693e-01
```

The eSCORE could be also accessed via Python:

```
from barnaba import escore
Score = escore.Escore([path_to_pdb])
(..)
# see example_12_escore.ipynb of barnaba package https://github.com/srnas/barnaba
```

It does not work on M1 mac (problem to compile mdtraj).

```
class rna_tools.tools.mq.eSCORE.eSCORE.eSCORE
```

Wrapper class for eSCORE

```
run(path_to_pdb, verbose=False)
```

Run program with previously set flags. This method should work if program wants some command some options and nothing more, otherwise you should override it.

```
rna_tools.tools.mq.eSCORE.eSCORE.exe(cmd)
```

```
rna_tools.tools.mq.eSCORE.eSCORE.main()
```

4.9.6 3dRNAscore

This module contains functions for computing 3dRNAscore

Install:

```
make # make clean if you don't have clean
```

At Mac there is a problem (201128, 211103):

```
(py37) [mx] example$ ../bin/3dRNAscore -s:l score.in >score.txt
[1] 69818 segmentation fault ../bin/3dRNAscore -s:l score.in > score.txt
```

```
class rna_tools.tools.mq.RNAscore.RNAscore.RNAscore(job_id=None)
```

Wrapper class for running 3dRNAscore

```
program_name = '/bin/3dRNAscore'
```

```
run(path_to_pdb, verbose=1)
```

Run program with previously set flags. This method should work if program wants some command some options and nothing more, otherwise you should override it.

```
rna_tools.tools.mq.RNAscore.RNAscore.main()
```

4.9.7 RNA3DCNN

This module contains functions for computing RNA3DCNN potential

Output:

```
Trainable params: 4,282,801
Non-trainable params: 0
-----
Scores for each nucleotide in /var/folders/yc/ssr9692s5fzf7k165grnhpk80000gp/T/
↳ tmp6jy2ud/query.pdb:
[[ 0.02462262]
 [ 0.03271335]
 [ 0.06199259]
 [ 0.02006263]
 [ 0.05937254]
 [ 0.12025979]
 [ 0.20201728]
 [ 0.24463326]
```

(continues on next page)

(continued from previous page)

```

[ 0.43518737]
[ 0.7260638 ]
[ 0.6140108 ]
[ 0.6588027 ]
[ 0.7668936 ]
[ 0.4776191 ]
[ 0.39859247]
[ 0.572009 ]
[ 0.64892375]
[ 0.11587611]
[ 0.0560993 ]
[ 0.05285829]
[ 0.0167731 ]
[ 0.01759553]
[ 0.02143204]
[-0.01818037]
Total score for /var/folders/yc/ssr9692s5fzf7k165grnhpk80000gp/T/tmpg6jy2ud/query.pdb
↳ is 6.3262305

```

If missing atoms:

```

Total params: 4,282,801
Trainable params: 4,282,801
Non-trainable params: 0

-----
There is no atom O5' in residue 620A in chain A in PDB /var/folders/yc/
↳ssr9692s5fzf7k165grnhpk80000gp/T/tmpx87uus6x/query.pdb.
There is no atom O5' in residue 635A in chain B in PDB /var/folders/yc/
↳ssr9692s5fzf7k165grnhpk80000gp/T/tmpx87uus6x/query.pdb.
There is no atom O5' in residue 1750G in chain C in PDB /var/folders/yc/
↳ssr9692s5fzf7k165grnhpk80000gp/T/tmpx87uus6x/query.pdb.
Scores for each nucleotide in /var/folders/yc/ssr9692s5fzf7k165grnhpk80000gp/T/
↳tmpx87uus6x/query.pdb:
[]
Total score for /var/folders/yc/ssr9692s5fzf7k165grnhpk80000gp/T/tmpx87uus6x/query.pdb
↳ is 0.0

```

```
class rna_tools.tools.mq.RNA3DCNN.RNA3DCNN.RNA3DCNN
```

Wrapper class for RNA3DCNN.

```
max_seq_len = 100000
```

```
run(path_to_pdb, verbose=False)
```

Run program with previously set flags. This method should work if program wants some command some options and nothing more, otherwise you should override it.

```
rna_tools.tools.mq.RNA3DCNN.RNA3DCNN.main()
```

4.9.8 FARNA

Wrapper for ROSETTA software for structure prediction of small RNA sequences

```
class rna_tools.tools.mq.FARNA.FARNA.FARNA(sequence='test', seq_name='test', job_id=None)
```

Wrapper class for running ROSETTA scoring function automatically.

```
best_energy = ''
```

```
cleanup()
```

Run this method when you are done with this wrapper. It deletes sandbox, some of it's children may do some other cleanup here.

```
db_path = ''
```

```
executable = 'rna_minimize'
```

```
get_result()
```

Parse and get result from score file created during ROSETTA run

All results are kept in self.result, but only global score is returned

```
input_file = ''
```

```
input_fn = 'seq.fasta'
```

```
mqap(pdb)
```

Total weighted score:s+(?P<ROSETTA_SCORE>[-d.]*)

```
program_name = 'farna'
```

```
run(pdb_file, hires, verbose=False, system=False)
```

Compute FARNA potential for a single file

Parameters

- **file** (* *pdb_file* = path to *pdb*)
- **True/False** (* *global_energy_score* =)

Output:

- A list of energies, e.g:

```
['-21.721', '-0.899', '-20.961', '-84.498', '-16.574', '-180.939', '11.549', '7.475', '-17.257', '-306.324', '0.0', '0.0', '17.503', '0.0']
```

??? or a dictionary of lists of local scores, eg:

```
{
  'N_BS': [17.0, -0.70039, -0.720981, -0.685238, -0.734146, ... ],
  'atom_pair_constraint': [0.0, -0.764688, -0.773833, ... ],
  ...
}
```

```
sandbox()
```

Create a sandbox in temporary folder for running program.

```
src_bin = ''
```

4.9.9 ClashScore

class rna_tools.tools.mq.ClashScore.ClashScore.ClashScore

ClashScore: Wrapper class for running phenix.clashscore

cleanup()

run(fn, verbose=False)

Parameters

- **fn** (*string*) – path to a file
- **verbose** (*bool*) – be verbose

Returns

clashscore (float)

Example:

```

/Applications/phenix-1.18.2-3874/build/bin/phenix.clashscore test/1xjrA.pdb
Using electron cloud x-H distances and vdW radii

Adding H/D atoms with reduce...

Bad Clashes >= 0.4 Angstrom:
A 17 A N1 A 34 G N1 :0.430
clashscore = 0.66
test/1xjrA.pdb
0.66
/Applications/phenix-1.18.2-3874/build/bin/phenix.clashscore test/1xjrA_M1.pdb
Using electron cloud x-H distances and vdW radii

Adding H/D atoms with reduce...

Bad Clashes >= 0.4 Angstrom:
A 41 G H2' A 42 U OP1 :0.738
A 25 U H6 A 25 U HO2' :0.659
A 46 U H2' A 47 U OP2 :0.656
A 29 A H5'' A 30 U O2' :0.623
(...)

A 24 G O2' A 26 A C8 :0.410
A 42 U O2' A 43 G H8 :0.409
A 43 G H2' A 44 A O4' :0.408
A 28 G C8 A 28 G O2' :0.403
clashscore = 15.45
test/1xjrA_M1.pdb
15.45

# so the output is
clashscore = 15.45

```

rna_tools.tools.mq.ClashScore.ClashScore.test()

4.9.10 AnalyzeGeometry

This module contains functions for computing AnalyzeGeometry.

class `rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry(verbose=False)`

Wrapper class for running clashscore

Note

Sequence is required to get the length to calculate % of correct residues.

`cleanup()`

`run(name, verbose=False)`

Parameters

- **name** (*str*) – the path of the file to wrap
- **verbose** (*boolen*) – be verbose

Returns

score; % of Backbone torsion suites (# of them per seq)

Return type

float

Output:

```
-----Backbone torsion suites-----
Suite ID          suite  suiteness  triaged angle
  A A    3          !!    0.000    delta
  G A    4          !!    0.000  epsilon-1
  G A   11          !!    0.000     None
  C A   20          !!    0.000    gamma
  U A   25          !!    0.000    delta
  A A   26          !!    0.000  delta-1
  A A   29          !!    0.000     None
  U A   30          !!    0.000  epsilon-1
  G A   41          !!    0.000    delta
  U A   42          !!    0.000  delta-1
  A A   45          !!    0.000    delta
  U A   46          !!    0.000  epsilon-1
  U A   47          !!    0.000  delta-1
11 suites triaged and 0 incomplete leaving 35 suites
13/46 suite outliers present
Average suiteness: 0.490
13 # count lines after 'triaged angle' and minus 3 (# of last lines)
test/1xjrA_M1.pdb
34.7826
```

Output for a perfect structure:

```
/Applications/phenix-1.18.2-3874/build/bin/phenix.rna_validate /Users/magnus/
work/src/rna-tools/rna_tools/input/mq/5e3hBC.pdb
```

(continues on next page)

(continued from previous page)

```

CGACGCUAGCGUACGCUAGCGUCG
AnalyzeGeomtery:: -----Backbone bond lenth-----

All bonds within 4.0 sigma of ideal values.

-----Backbone bond angles-----

All angles within 4.0 sigma of ideal values.

-----Sugar pucker-----

All puckers have reasonable geometry.

-----Backbone torsion suites-----

0 suites triaged and 0 incomplete leaving 24 suites
All RNA torsion suites are reasonable.
Average suiteness: 0.766

```

```
rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry.main()
```

4.10 RNA 3D structure prediction

4.10.1 ROSETTA

A set of wrappers around Rosetta (<https://www.rosettacommons.org/>), mostly based on C. Y. Cheng, F. C. Chou, and R. Das, Modeling complex RNA tertiary folds with Rosetta, 1st ed., vol. 553. Elsevier Inc., 2015. <http://www.sciencedirect.com/science/article/pii/S0076687914000524>

4.10.1.1 Run (modeling)

rna_rosetta_run.py

rna_rosetta_run.py - prepare & run ROSETTA simulations

Based on C. Y. Cheng, F. C. Chou, and R. Das, Modeling complex RNA tertiary folds with Rosetta, 1st ed., vol. 553. Elsevier Inc., 2015. <http://www.sciencedirect.com/science/article/pii/S0076687914000524>

The script makes(1) a folder for you job, with seq.fa, ss.fa, input file is copied as input.fa to the folder(2) make helices(3) prepare rosetta input files(4) sends jobs to the cluster.

The header is take from the fast file(`> /header /`) not from the filename of your Fasta file.

I discovered this:

```
qstat -xml | tr '\n' ' ' | sed 's#<job_list[^\>]*>#\n#g' \
```

```
> | sed 's#<[^\>]*>###g' | grep "" | column -t
```

(<https://stackoverflow.com/questions/26104116/qstat-and-long-job-names>) so there is now need to shorted my job ids.

Helix

Run:

```
rna_rosetta_run.py -i -e -r -g -c 200 cp20.fa
```

-i:

```
# prepare a folder for a run
>cp20
AUUAUCAAGAAUCUCAAGAGAGAUAGCAACCUAGCAUAACGAGCAAGGUGCUGCAAAAUAGAUUAGCCAAAUUCAAUUGGAAAAAUGUUA
.((((....((((.....))))(((.((((..[[[.])..))..)).....))))).(((.....))).....
↪]]]].

[peyote2] ~ rna_rosetta_run.py -i cp20.fa
run rosetta for:
cp20
AUUAUCAAGAAUCUCAAGAGAGAUAGCAACCUAGCAUAACGAGCAAGGUGCUGCAAAAUAGAUUAGCCAAAUUCAAUUGGAAAAAUGUUA
.((((....((((.....))))(((.((((..[[[.])..))..)).....))))).(((.....))).....
↪]]]].
/home / magnus // cp20 / created
Seq & ss created
```

Troubleshooting.

If one of the helices is missing you will get:

```
IOError: [Errno 2] No such file or directory: 'helix1.out'
rosetta_submit.py README_FARFAR o 500 100 taf
Could not find: README_FARFAR
```

and the problem was a1 and g8 pairing:

```
outputting command line to: helix0.RUN # previous helix #0
Sequence: AUGG CCGG
Secstruc: (((()))))
Not complementary at positions a1 and g8!
Sequence: GUGGG CCCAU
Secstruc: (((((()))))

Writing to fasta file: helix2.fasta # next helix #2
```

My case with a modeling of rp12

Sequence: cc gc Secstruc: (()) Not complementary at positions 1 and 4!

edit the secondary structure, run the program with -i(init, to overwrite seq.fa, ss.fa) and then it works.

Notes:

```
rp17hc 6 characters
```

```
usage: rna_rosetta_run.py [-h] [-i] [-e] [-r] [-g] [-m MOTIF] [-n NSTRUC]
                        [-c CPUS] [--sandbox SANDBOX]
                        file
```

file

file: > a04 UAUAACAUAUAUUUUGACAAUAUGGGUCAUAAGUUUCUACCGGAAUACCGUAAAUAU-
UCUGACUAUGUAUA ((((((((((((.....)))))).).....((((.....))))).).....))))))

-h, --help

show this help message and exit

-i, --init

prepare _folder with seq and ss

-e, --helices

produce h(E)lices

-r, --rosetta

prepare rosetta files (still you need *go* to send jobs to a cluster)

-g, --go

send jobs to a cluster(run qsubMINI)

-m <motif>, --motif <motif>

include a motif file, e.g. -s E-loop_1q9a_mutated_no_flanks_renumber.pdb

-n <nstruc>, --nstruc <nstruc>

of structures you want to get

-c <cpus>, --cpus <cpus>

of cpus to be used

--sandbox <sandbox>

where to run it (default: RNA_ROSETTA_RUN_ROOT_DIR_MODELING)

rna_rosetta_run.py - prepare & run ROSETTA simulations

Based on C. Y. Cheng, F. C. Chou, and R. Das, Modeling complex RNA tertiary folds with Rosetta, 1st ed., vol. 553. Elsevier Inc., 2015. <http://www.sciencedirect.com/science/article/pii/S0076687914000524>

The script makes(1) a folder for you job, with seq.fa, ss.fa, input file is copied as input.fa to the folder(2) make helices(3) prepare rosetta input files(4) sends jobs to the cluster.

The header is take from the fast file(`> /header /`) not from the filename of your Fasta file.

I discovered this:

```
qstat -xml | tr '\n' ' ' | sed 's#<job_list[^>]*>#\n#g' \
```

```
> | sed 's#<[^>]*>##g' | grep " " | column -t
```

(<https://stackoverflow.com/questions/26104116/qstat-and-long-job-names>) so there is now need to shorted my job ids.

Helix

Run:

```
rna_rosetta_run.py -i -e -r -g -c 200 cp20.fa
```

-i:

```
# prepare a folder for a run
>cp20
AUUAUCAAGAAUCUCAAGAGAGAUAGCAACCUUGCAAUAACGAGCAAGGUGCUAAAAUAGUAAGCCAAAUCAAUUGGAAAAAUGUUA
.((((....((((.....))))(((((....[[[...)).)))).)).....))))).((((.....)))).....
↪]]]].

[peyote2] ~ rna_rosetta_run.py -i cp20.fa
run rosetta for:
cp20
AUUAUCAAGAAUCUCAAGAGAGAUAGCAACCUUGCAAUAACGAGCAAGGUGCUAAAAUAGUAAGCCAAAUCAAUUGGAAAAAUGUUA
.((((....((((.....))))(((((....[[[...)).)))).)).....))))).((((.....)))).....
↪]]]].
/home / magnus // cp20 / created
Seq & ss created
```

Troubleshooting.

If one of the helices is missing you will get:

```
IOError: [Errno 2] No such file or directory: 'helix1.out'
rosetta_submit.py README_FARFAR o 500 100 taf
Could not find: README_FARFAR
```

and the problem was a1 and g8 pairing:

```
outputting command line to: helix0.RUN # previous helix #0
Sequence: AUGG CCGG
Secstruc: (((()))
Not complementary at positions a1 and g8!
Sequence: GUGGG CCCAU
Secstruc: (((((()))))
Writing to fasta file: helix2.fasta # next helix #2
```

My case with a modeling of rp12

Sequence: cc gc Secstruc: (()) Not complementary at positions 1 and 4!

edit the secondary structure, run the program with -i(init, to overwrite seq.fa, ss.fa) and then it works.

Notes:

```
rp17hc 6 characters
```

```
class rna_tools.tools.rna_rosetta.rna_rosetta_run.CustomFormatter(prog, indent_increment=2,
                                                                    max_help_position=24,
                                                                    width=None)
```

```
rna_tools.tools.rna_rosetta.rna_rosetta_run.get_parser()
```

```
rna_tools.tools.rna_rosetta.rna_rosetta_run.go()
```

send jobs to a cluster(run qsubMINI)

```
rna_tools.tools.rna_rosetta.rna_rosetta_run.main()
```

Pipeline for modeling RNA

`rna_tools.tools.rna_rosetta.rna_rosetta_run.prepare_folder(args, header, seq, ss, path)`

Make folder for you job, with seq.fa, ss.fa, input file is copied as input.fa to the folder.

For ss lowercase is needed when used with motifs, otherwise:

```
[peyote2] aa20$ rna_rosetta_run.py -r -m E-loop_1q9a_mutated_no_flanks_renumber_for_
↳acy20.pdb ~/aa20.fa
2019-05-03 21:31:30,842 rpt_config_local.py::<module>::rpt_config_local loading...
run rosetta for:
aa20
UACGUUCAUCAUCCGUUUGGAUGACGGAAGUAAGCGAAAGCUGAAGGAACGCAUG
..(((((((.....))))))..[.....(((.....)).....)))]...
rna_denovo_setup.py -fasta seq.fa -secstruct_file ss.fa -cycles 20000 -no_minimize -
↳nstruct 50 -s E-loop_1q9a_mutated_no_flanks_renumber_for_acy20.pdb -silent_
↳helix0.out helix1.out helix2.out -input_silent_res 3-7 47-51 9-14 19-24 33-35 40-
↳42

Sequence: UACGUUCAUCAUCCGUUUGGAUGACGGAAGUAAGCGAAAGCUGAAGGAACGCAUG
Secstruc: ..(((((((.....))))))..[.....(((.....)).....)))]...
aaguagaag
AAGUAGAAG
Traceback (most recent call last):
  File "/home/magnus/opt/rosetta_src_2016.13.58602_bundle/tools/rna_tools/bin//rna_
↳denovo_setup.py", line 170, in <module>
    raise ValueError('The sequence in %s does not match input sequence!!' % pdb)
ValueError: The sequence in E-loop_1q9a_mutated_no_flanks_renumber_for_acy20.pdb_
↳does not match input sequence!!
rosetta_submit.py README_FARFAR o 200 100 aa20_
Could not find: README_FARFAR
```

`rna_tools.tools.rna_rosetta.rna_rosetta_run.prepare_helices()`

Make helices(wrapper around 'helix_preassemble_setup.py')

Warning

I think multiprocessing of helixX.run does not work.

`rna_tools.tools.rna_rosetta.rna_rosetta_run.prepare_rosetta(header, cpus, motif, nstruc)`

Prepare ROSETTA using `rna_denovo_setup.py`

`cpus` is used to calc `nstruc` per job to get 10k structures per full run:

```
:param nstruc: how many structures you want to obtain
:type nstruc: int
:param nstruct = int:
:type nstruct = int: math.floor(20000 / cpus)
:param motif: motif file; e.g., -s E-loop_1q9a_mutated_no_flanks_renumber.pdb
:param 50:
:type 50: nstruc) = 10k / 200 (cpus
```

4.10.1.2 Get a number of structures

rna_rosetta_n.py

rna_rosetta_n.py - show me # of structure in a silent file

Example:

```
$ rna_rosetta_n.py ade.out
21594

$ rna_rosetta_n.py *out
default.out 100
default1.out 200
default2.out 200
default3.out 100
```

```
usage: rna_rosetta_n.py [-h] [-v] [-0] file [file ...]
```

file

ade.out

-h, --help

show this help message and exit

-v, --verbose

-0, --zero

rna_rosetta_n.py - show me # of structure in a silent file

Example:

```
$ rna_rosetta_n.py ade.out
21594

$ rna_rosetta_n.py *out
default.out 100
default1.out 200
default2.out 200
default3.out 100
```

`rna_tools.tools.rna_rosetta.rna_rosetta_n.get_no_structures(file)`

`rna_tools.tools.rna_rosetta.rna_rosetta_n.get_parser()`

`rna_tools.tools.rna_rosetta.rna_rosetta_n.run(f)`

Pipeline for modeling RNA.

4.10.1.3 Get a head of a Rosetta silent file

rna_rosetta_n.py

rna_rosetta_head.py - get a head of a Rosetta silent file.

Example:

```
$ rna_rosetta_head.py -n 10000 thf.out
# a new file will be created, thf_10000.out
```

Silent file:

```
[peyote2] thf head -n 100 thf.out
SEQUENCE:
→ggagaguagaugaucgcuuaagugugugugaugggaugucgucacacaacgaagcgagagcgcgggugaaucaugcauccgcucca
SCORE:   score   rna_data_backbone   rna_vdw   rna_base_backbone   rna_backbone_
→backbone   rna_repulsive   rna_base_pair   rna_base_axis   rna_base_stagger   rna_
→base_stack   rna_base_stack_axis   rna_rg   atom_pair_constraint   linear_
→chainbreak   N_WC   N_NWC   N_BS description
REMARK BINARY SILENTFILE
SCORE:   -601.975           0.000   31.113           -16.960           -
→3.888           20.501           -302.742           -38.531           -158.004           _
→-80.764           -110.053           23.750           0.000           33.601_
→           32           6           86   S_000001_000
FOLD_TREE EDGE 1 4 -1 JEDGE 4 85 1 C4 C2 END EDGE 4 5 -1 EDGE 85 80 -1 EDGE 85_
→89 -1 JEDGE 80 40 5 C4 C2 END EDGE 80 78 -1 EDGE 40 43 -1 EDGE 40 33 -1 _
→JEDGE 33 45 4 C4 C2 END EDGE 45 54 -1 EDGE 45 44 -1 EDGE 33 20 -1 JEDGE 20 65_
→3 C2 C4 END EDGE 65 67 -1 EDGE 20 17 -1 EDGE 65 63 -1 JEDGE 17 69 2 C4 C2 _
→END JEDGE 63 58 6 C4 C2 END EDGE 17 6 -1 EDGE 58 59 -1 EDGE 63 60 -1 EDGE_
→69 77 -1 EDGE 58 55 -1 EDGE 69 68 -1 S_000001_000
RT -0.987743 0.139354 0.0703103 0.135963 0.989404 -0.0509304 -0.0766626 -0.0407466 -0.
→996224 6.25631 0.103544 0.0647696 S_000001_000
RT -0.98312 0.1587 -0.091045 0.166923 0.981743 -0.0912024 0.074909 -0.10486 -0.991662 5.
→89962 -1.95819 -0.1075 S_000001_000
RT -0.987645 0.154078 0.0285994 0.153854 0.988044 -0.00989514 -0.0297821 -0.00537275 -0.
→999542 6.13138 1.047 0.115722 S_000001_000
RT -0.989884 0.140722 0.0180554 0.140036 0.989532 -0.0348618 -0.0227723 -0.0319807 -0.
→999229 6.21787 0.201588 -0.0264223 S_000001_000
RT -0.988455 0.134013 0.0706914 0.134924 0.990822 0.00825457 -0.0689364 0.0176973 -0.
→997464 6.19447 0.189237 0.125791 S_000001_000
RT -0.990412 0.138036 0.00546261 0.137927 0.990299 -0.0168644 -0.00773751 -0.0159492 -0.
→999843 6.25456 0.0842849 -0.0135807 S_000001_000
ANNOTATED_SEQUENCE: g[RGU:LowerRNA:Virtual_Phosphate]gaga[RAD:rna_cutpoint_
→lower]g[RGU:rna_cutpoint_upper]uagaugaucgcuuaagugugugugaugggauguc[RCY:rna_cutpoint_
→lower]g[RGU:rna_cutpoint_upper]ucacacaacg[RGU:rna_cutpoint_lower]a[RAD:rna_cutpoint_
→upper]agcg[RGU:rna_cutpoint_lower]a[RAD:rna_cutpoint_upper]gagcgcg[RGU:rna_cutpoint_
→lower]g[RGU:rna_cutpoint_upper]ugaaucau[URA:rna_cutpoint_lower]g[RGU:rna_cutpoint_
→upper]cauccgcucca[RAD:UpperRNA] S_000001_000
Le3nY9smsa4zEMGdVAA+z+e
```

It seems to work:

```
-rw-rw-r-- 1 magnus users 474M 2017-08-06 05:25 thf_10000.out
-rw-rw-r-- 1 magnus users 947M 2017-08-06 04:54 thf.out

[peyote2] thf rna_rosetta_n.py thf_10000.out
10000
```

```
usage: rna_rosetta_n.py [-h] [-v] [-n NSTRUC] file
```

file

ade.out

-h, --help

show this help message and exit

-v, --verbose**-n <nstruc>, --nstruc <nstruc>**

rna_roseta_head.py - get a head of a Rosetta silent file.

Example:

```
$ rna_rosetta_head.py -n 10000 thf.out
# a new file will be created, thf_10000.out
```

Silent file:

```
[peyote2] thf head -n 100 thf.out
SEQUENCE:
→ggagaguagaugauucgcuuaagugugugugaauugggaugcugucacacaacgaagcgagagcgcgguagaucaugcauccgcucca
SCORE:   score   rna_data_backbone   rna_vdw   rna_base_backbone   rna_backbone_
→backbone   rna_repulsive   rna_base_pair   rna_base_axis   rna_base_stagger   rna_
→base_stack   rna_base_stack_axis   rna_rg   atom_pair_constraint   linear_
→chainbreak   N_WC   N_NWC   N_BS description
REMARK BINARY SILENTFILE
SCORE:  -601.975           0.000    31.113           -16.960           -
→3.888           20.501           -302.742           -38.531           -158.004           1
→-80.764           -110.053           23.750           0.000           33.6011
→           32           6           86   S_0000001_000
FOLD_TREE EDGE 1 4 -1 JEDGE 4 85 1 C4 C2 END EDGE 4 5 -1 EDGE 85 80 -1 EDGE 85 1
→89 -1 JEDGE 80 40 5 C4 C2 END EDGE 80 78 -1 EDGE 40 43 -1 EDGE 40 33 -1 1
→JEDGE 33 45 4 C4 C2 END EDGE 45 54 -1 EDGE 45 44 -1 EDGE 33 20 -1 JEDGE 20 65 1
→3 C2 C4 END EDGE 65 67 -1 EDGE 20 17 -1 EDGE 65 63 -1 JEDGE 17 69 2 C4 C2 1
→END JEDGE 63 58 6 C4 C2 END EDGE 17 6 -1 EDGE 58 59 -1 EDGE 63 60 -1 EDGE 1
→69 77 -1 EDGE 58 55 -1 EDGE 69 68 -1 S_0000001_000
RT -0.987743 0.139354 0.0703103 0.135963 0.989404 -0.0509304 -0.0766626 -0.0407466 -0.
→996224 6.25631 0.103544 0.0647696 S_0000001_000
RT -0.98312 0.1587 -0.091045 0.166923 0.981743 -0.0912024 0.074909 -0.10486 -0.991662 5.
→89962 -1.95819 -0.1075 S_0000001_000
RT -0.987645 0.154078 0.0285994 0.153854 0.988044 -0.00989514 -0.0297821 -0.00537275 -0.
→999542 6.13138 1.047 0.115722 S_0000001_000
RT -0.989884 0.140722 0.0180554 0.140036 0.989532 -0.0348618 -0.0227723 -0.0319807 -0.
→999229 6.21787 0.201588 -0.0264223 S_0000001_000
RT -0.988455 0.134013 0.0706914 0.134924 0.990822 0.00825457 -0.0689364 0.0176973 -0.
```

(continues on next page)

(continued from previous page)

```

↪997464 6.19447 0.189237 0.125791 S_0000001_000
RT -0.990412 0.138036 0.00546261 0.137927 0.990299 -0.0168644 -0.00773751 -0.0159492 -0.
↪999843 6.25456 0.0842849 -0.0135807 S_0000001_000
ANNOTATED_SEQUENCE: g[RGU:LowerRNA:Virtual_Phosphate]gaga[RAD:rna_cutpoint_
↪lower]g[RGU:rna_cutpoint_upper]uagaugauucgcguuaaguguguguaugggauguc[RCY:rna_cutpoint_
↪lower]g[RGU:rna_cutpoint_upper]ucacacaacg[RGU:rna_cutpoint_lower]a[RAD:rna_cutpoint_
↪upper]agcg[RGU:rna_cutpoint_lower]a[RAD:rna_cutpoint_upper]gagcgcg[RGU:rna_cutpoint_
↪lower]g[RGU:rna_cutpoint_upper]ugaaucauu[URA:rna_cutpoint_lower]g[RGU:rna_cutpoint_
↪upper]cauccgcucca[RAD:UpperRNA] S_0000001_000
Le3nY9smsa4zEMGdvAA+z+e

```

It seems to work:

```

-rw-rw-r-- 1 magnus users 474M 2017-08-06 05:25 thf_10000.out
-rw-rw-r-- 1 magnus users 947M 2017-08-06 04:54 thf.out

[peyote2] thf rna_rosetta_n.py thf_10000.out
10000

```

```
rna_tools.tools.rna_rosetta.rna_rosetta_head.get_parser()
```

```
rna_tools.tools.rna_rosetta.rna_rosetta_head.run()
```

Pipeline for modeling RNA.

4.10.1.4 Cluster

rna_rosetta_cluster.py

rna_rosetta_cluster.py - cluster silent Rosetta files

A wrapper to ROSETTA tools for RNA modeling

Based on C. Y. Cheng, F. C. Chou, and R. Das, Modeling complex RNA tertiary folds with Rosetta, 1st ed., vol. 553. Elsevier Inc., 2015. <http://www.sciencedirect.com/science/article/pii/S0076687914000524>

```
rna_rosetta_cluster.py ade_min.out 20000
```

Take $n * 0.005$ (.5%) of all frames and put them into *selected.out*. Then the tool clusters this *selected.out*.

```

usage: rna_rosetta_cluster.py [-h] [--no_select]
                               [--radius-inc-step RADIUS_INC_STEP]
                               [--limit-clusters LIMIT_CLUSTERS]
                               file n

```

file

ade.out

n

of total structures

-h, --help

show this help message and exit

--no_select

Don't run selection once again. Use *selected.out* in the current folder

--radius-inc-step <radius_inc_step>

radius incremental step, default 0.5

--limit-clusters <limit_clusters>

of clusters

rna_rosetta_cluster.py - cluster silent Rosetta files

A wrapper to ROSETTA tools for RNA modeling

Based on C. Y. Cheng, F. C. Chou, and R. Das, Modeling complex RNA tertiary folds with Rosetta, 1st ed., vol. 553. Elsevier Inc., 2015. <http://www.sciencedirect.com/science/article/pii/S0076687914000524>

```
rna_rosetta_cluster.py ade_min.out 20000
```

Take $n * 0.005$ (.5%) of all frames and put them into *selected.out*. Then the tool clusters this *selected.out*.

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.cluster(radius, limit_clusters)
```

Internal function of cluster_loop: It removes cluster.out first.

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.cluster_loop(ns, radius_inc_step, limit_clusters)
```

Go from radius 1 to get 1/6 of structures of ns (# of selected structures) in the first cluster, then it stops.

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.extract()
```

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.get_no_structures(file)
```

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.get_no_structures_in_first_cluster(fn)
```

Get # of structures in a silent file.

Parameters

fn (*string*) – a filename to a silent file

Returns

int: # of structures in a silent file

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.get_parser()
```

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.get_selected(file, nc)
```

Get selected for clustering

```
rna_tools.tools.rna_rosetta.rna_rosetta_cluster.run()
```

Pipeline for modeling RNA

4.10.1.5 Minimize**rna_rosetta_min.py**

rna_rosetta_min.py - a script to do minimization

The script takes the number of structures and the analyzed silence file and does the maths.

Job names will be as your silent file preceding with ~, .e.g ~tha.

<http://www.sciencedirect.com/science/article/pii/S0076687914000524>

```
ade$ rna_rosetta_cluster.py ade.out
```

The first number states how many processors to use for the run, while the second number is 1/6 the total number of previously generated FARNA models. If you are running on a supercomputer that only allows specific multiples of processors, use an appropriate number for the first input.:

```
rosetta_submit.py min_cmdline min_out 1 24
```

rosetta_submit.py min_cmdline min_out [1] [16] The first number states how many processors to use for each line in min_cmdline. Here, enter 1 for the first input so that the total number of processors used will be equal to the number of processors entered with the “-proc” flag in command line [12], above. The second number states the maximum time each job will be allowed to run (walltime). Start the run with the appropriate command listed by the out- put above (e.g., source qsubMPI for the Stampede cluster).

E.g. for 20k silet file, 1/6 will be minimized = 3.3k:

```
parallel_min_setup.py -silent rp21cr62.out -tag rp21cr62_min -proc 200 -nstruct 3200 -
↳out_folder mo -out_script MINIMIZE " -ignore_zero_occupancy false "
rosetta_submit.py MINIMIZE mo 1 100 m

[peyote2] rp21 easy_cat.py mo
Cutting into: rp21_min.out ... from 200 primary files. Found 3200 decoys.

# on 200 cpus it took around ~30min
```

```
usage: rna_rosetta_min.py [-h] [-g] [-c CPUS] file
```

file

ade.out

-h, --help

show this help message and exit

-g, --go

-c <cpus>, --cpus <cpus>

default: 200

rna_rosetta_min.py - a script to do minimization

The script takes the number of structures and the analyzed silence file and does the maths.

Job names will be as your silent file preceding with ~, .e.g ~tha.

<http://www.sciencedirect.com/science/article/pii/S0076687914000524>

```
ade$ rna_rosetta_cluster.py ade.out
```

The first number states how many processors to use for the run, while the second number is 1/6 the total number of previously generated FARNA models. If you are running on a supercomputer that only allows specific multiples of processors, use an appropriate number for the first input.:

```
rosetta_submit.py min_cmdline min_out 1 24
```

rosetta_submit.py min_cmdline min_out [1] [16] The first number states how many processors to use for each line in min_cmdline. Here, enter 1 for the first input so that the total number of processors used will be equal to the number of processors entered with the “-proc” flag in command line [12], above. The second number states the maximum time

each job will be allowed to run (walltime). Start the run with the appropriate command listed by the out- put above (e.g., source qsubMPI for the Stampede cluster).

E.g. for 20k silent file, 1/6 will be minimized = 3.3k:

```
parallel_min_setup.py -silent rp21cr62.out -tag rp21cr62_min -proc 200 -nstruct 3200 -
→out_folder mo -out_script MINIMIZE " -ignore_zero_occupancy false "
rosetta_submit.py MINIMIZE mo 1 100 m

[peyote2] rp21 easy_cat.py mo
Cattig into: rp21_min.out ... from 200 primary files. Found 3200 decoys.

# on 200 cpus it took around ~30min
```

`rna_tools.tools.rna_rosetta.rna_rosetta_min.get_no_structures(file)`

Get a number of structures in a silent file

`rna_tools.tools.rna_rosetta.rna_rosetta_min.get_parser()`

`rna_tools.tools.rna_rosetta.rna_rosetta_min.min(silent_file, take_n, cpus, go)`

Run parallel_min_setup (to MINIMIZE file), rosetta_submit.py, and qsubMINI.

Fix on the way, qsub files:

```
-out:file:silent mo/0/mo/123/tha_min.out -> -out:file:silent mo/123/tha_min.out
```

I don't know why mo/0/ is there. I might be because of my changes in rosetta_submit.py (?).

`rna_tools.tools.rna_rosetta.rna_rosetta_min.run()`

4.10.1.6 Extract lowscore decoy

`rna_rosetta_extract_lowscore_decoys.py`

`rna_rosetta_extract_lowscore_decoys.py` - a simple wrapper to `extract_lowscore_decoys.py`

To be used in Jupyter notebooks and other scripts.

```
usage: rna_rosetta_extract_lowscore_decoys.py [-h] [-v] nstruc file
```

nstruc

of low score structures to obtained

file

silent file

-h, --help

show this help message and exit

-v, --verbose

be verbose

`rna_rosetta_extract_lowscore_decoys.py` - a simple wrapper to `extract_lowscore_decoys.py`

To be used in Jupyter notebooks and other scripts.

`rna_tools.tools.rna_rosetta.rna_rosetta_extract_lowscore_decoys.get_parser()`

`rna_tools.tools.rna_rosetta.rna_rosetta_extract_lowscore_decoys.rosetta_extract_lowscore_decoys`(*file*,
nstruc,
log=True)

4.10.1.7 Check progress

`rna_rosetta_cluster.py`

`rna_rosetta_check_progress.py` - check progress for many simulations of Rosetta

Example:

```
[peyote2] rosetta_jobs rna_rosetta_check_progress.py .
      jobs #curr #todo #decoys done
0 ./rp17s223    200     0    407 [ ]
1 ./rp17hcf     0     0     0 [ ]
# curr 232 #todo 0
```

```
usage: rna_rosetta_cluster.py [-h] [-v] [-m] [-s SELECT] [-k] dir
```

dir

directory with rosetta runs, define by `RNA_ROSETTA_RUN_ROOT_DIR_MODELING` right now:

-h, --help

show this help message and exit

-v, --verbose

be verbose

-m, --min-only

check only for mo folder

-s <select>, --select <select>

select for analysis only jobs with this phrase, .e.g., **evoseq_**

-k, --kill

kill (qdel) jobs if your reach limit (nstruc) of structure that you want, right now is 10000 structures

`rna_rosetta_check_progress.py` - check progress for many simulations of Rosetta

Example:

```
[peyote2] rosetta_jobs rna_rosetta_check_progress.py .
      jobs #curr #todo #decoys done
0 ./rp17s223    200     0    407 [ ]
1 ./rp17hcf     0     0     0 [ ]
# curr 232 #todo 0
```

`rna_tools.tools.rna_rosetta.rna_rosetta_check_progress.get_parser()`

`rna_tools.tools.rna_rosetta.rna_rosetta_check_progress.run_cmd(cmd)`

4.10.2 SimRNA

4.10.2.1 Select low energy frames

`rna_simrna_lowest.py`

`rna_simrna_lowest.py` - get lowest energy frames out of a SimRNA trajectory file

This code uses heavily the `SimRNATrajectory` class. Be default 100 lowest energy frames is exported.

```
usage: rna_simrna_lowest.py [-h] [-n NSTRUC] trafl
```

trafl

SimRNA trafl file

-h, --help

show this help message and exit

-n <nstruc>, --nstruc <nstruc>

SimRNA trafl file

4.10.2.2 Extract

`rna_simrna_extract.py`

`rna_simrna_extra.py` - extract full atom structures from a SimRNA trajectory file

Options:

SIMRNA_DATA_PATH has to be properly defined in `rpt_config_local`.

```
usage: rna_simrna_extract.py [-h] -t TEMPLATE -f TRAFI [-c]
                        [-n NUMBER_OF_STRUCTURES]
```

-h, --help

show this help message and exit

-t <template>, --template <template>

template PDB file used for reconstruction to full atom models

-f <trafl>, --trafl <trafl>

SimRNA trafl file

-c, --cleanup

Keep only `*_AA.pdb` files, move `*.ss_detected` and `*.pdbto` `_<traj name folder>`

-n <number_of_structures>, --number_of_structures <number_of_structures>

4.10.3 SimRNAweb

4.10.3.1 Download files of a SimRNAweb run

rna_simrnaweb_download_job.py

rna_simrnaweb_download_job.py - download model files, trajectory for a given SimRNAweb job.

Usage:

```
rp17pk$ rna_pdb_download_simrna_job.py 27b5093d -m -t -x
# download more clusters, trajectory, extract100

cp771_pk$ rna_pdb_download_simrna_job.py -t -x -m cf8f8bb2 -p cp771_pk
# download with a trajectory, and cluster #4 and #5, add to all pdb files
# prefix: cp771_pk

$ rna_simrnaweb_download_job.py --web-models rp17_well_d10_e1-a43d3ab5 --prefix tar
# prefix added will be tar_XXXX
```

Example:

```
rna_pdb_download_simrna_job.py -t -x -m 20569fa1 -p zmp_pk

[mm] zmp_pk ls
20569fa1_ALL_100low.trafl
_20569fa1-thrs7.10A_clust04
_20569fa1-thrs7.10A_clust05
_20569fa1_ALL_100low
data
rna_simrna_extract.log
subset.png
zmp_pk_20569fa1-thrs7.10A_clust01X.pdb
zmp_pk_20569fa1-thrs7.10A_clust02X.pdb
zmp_pk_20569fa1-thrs7.10A_clust03X.pdb
zmp_pk_20569fa1-thrs7.10A_clust04X.pdb
zmp_pk_20569fa1-thrs7.10A_clust05X.pdb
```

```
usage: rna_simrnaweb_download_job.py [-h] [-p PREFIX] [-n NSTRUC] [-e] [-m]
                                     [-r] [-c] [-d] [--top100] [--top200]
                                     [--web-models]
                                     job_id
```

job_id

job_id

-h, --help

show this help message and exit

-p <prefix>, --prefix <prefix>

prefix to the name, without `_`, be careful with this. If you have already some files with the given folder, their names might be changed.

-n <nstruc>, --nstruc <nstruc>

extract nstruc the lowest energy, this option must go with `-web`

- e, --extract**
extract nstruc the lowest energy, this option must go with `-web`
- m, --more_clusters**
download also cluster 4 and 5
- r, --remove-trajectory**
remove trajectory after analysis
- c, --cluster**
get trajectory from cluster OR local on your computer (mdfind for macOS)
- d, --download-trajectory**
web
- top100**
download top100 trajectory
- top200**
download top200 trajectory
- web-models**
web models download

4.10.4 SimRNATrajectory

SimRNATrajectory module.

SimRNATrajectory / Frame / Residue / Atom

```
class rna_tools.tools.simrna_trajectory.simrna_trajectory.Atom(name, x, y, z)
    x y z coord
```

```
    get_coord()
        Return coords (np.array).
```

```
class rna_tools.tools.simrna_trajectory.simrna_trajectory.Frame(id, header, coords,
                                                                top_level=False)
```

Syntax of header:

- write_number
- replica_id
- total_energy
- energy_without_restraints
- temperature

Warning

If there is an invalid frame, please use `repair_trafl.py` to fix the trajectory first.

```
rmsd_to(frame, verbose=False)
```

class `rna_tools.tools.simrna_trajectory.simrna_trajectory.Residue`(*id, p, c4p, n1n9, b1, b2*)

Create Residue object.

Each residue in SimRNA coarse-grained representation consists only 5 coarse-grained atoms:

- backbone: p = phosphate group, c4p = sugar moiety
- nucleotide: n1n9 = N1 for pyrimidines, N9 for purines, b1 = C2 for purines and pyrimidines, b2 = C4 for pyrimidines, C6 for purines

get_atoms()

Return all atoms

get_center()

Return MB for residue $((self.n1n9 + self.b2) / 2)$

class `rna_tools.tools.simrna_trajectory.simrna_trajectory.SimRNATrajectory`

load_from_file(*fn, debug_break=False, top_level=False, only_first_frame=False*)

Create a trajectory based on give filename.

Parameters

top_level – top_level = True, don't make a huge tree of objects (Residues/Atoms) == amazing speed up! Useful if you need only frames, energies and coords as text. You only get the info that is in header of each frame.

top_level = False, makes huge tree of objects (Residues/Atoms) == very slow for a huge trajectories

 **Warning**

Loads up whole trafil file into memory, and get stuck. Use this if you want to compute e.g. distances between atoms, get the positions of specified atoms etc. If you can not process your trajectory

use top_level=True or look at load_from_string() to load a frame by frame from a file.

h(eader), l(line), f(ile)

load_from_list(*frames*)

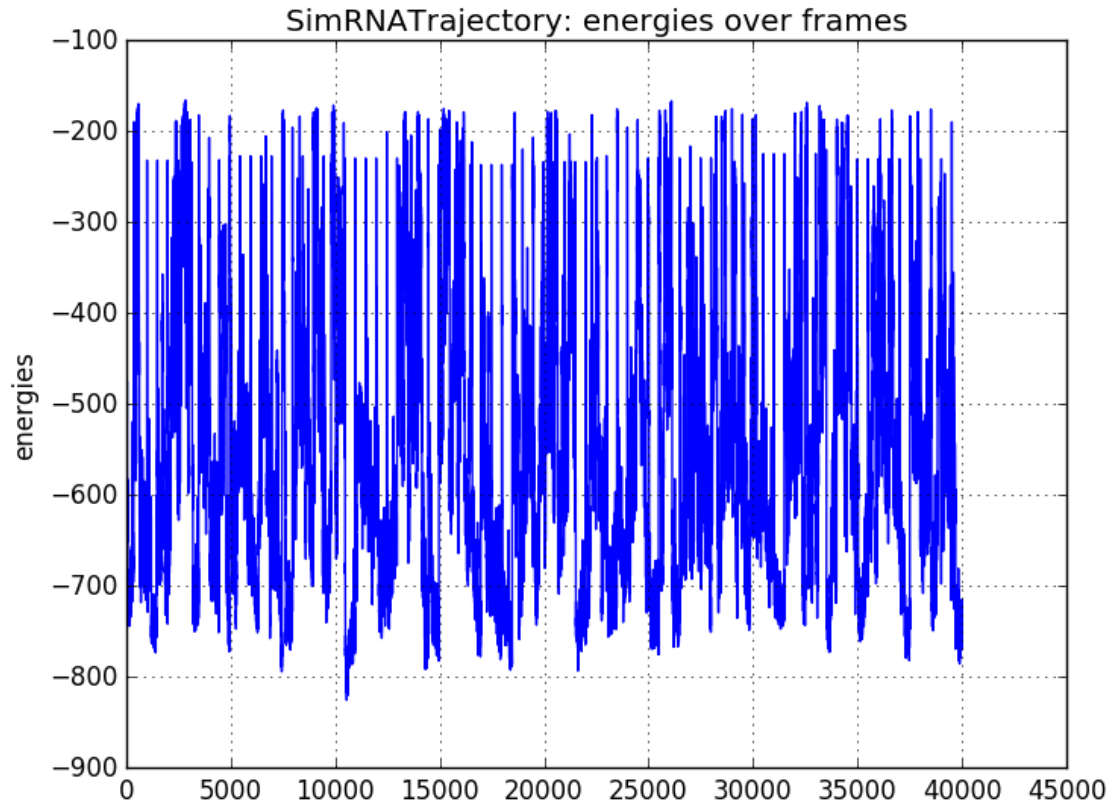
load_from_string(*c, txt*)

Create a trajectory based on given string (txt) with id given by c.

Faster method, loads only one frame at a time to memory, and after computations loads the next frame (memory efficient).

plot_energy(*plotfn='plot.png'*)

Plots the SimRNA energy of the trajectory.



`save(fn, verbose=True)`

Save the trajectory to file.

`sort(inplace=True)`

Sort frames within the trajectory according to energy.

4.11 RNA Refinement (QRNAS)

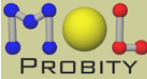
4.11.1 rna_refinement.py

rna_refinement - RNA refinement with QRNAS.


Models of RNA 3D structures obtained by modeling methods often suffer from local inaccuracies such as clashes or physically improbable bond lengths, backbone conformations, or sugar puckers. To ensure high quality of models, a procedure of refinement should be applied as a final step in the modeling pipeline. The software tool QRNAS was developed in our laboratory to perform local refinement of nucleic acid structures based on an extended version of the AMBER force field. The extensions consist of energy terms associated with introduction of explicit hydrogen bonds, idealization of base pair planarity and regularization of backbone conformation.

Read more: Piatkowski, P., Kasprzak, J. M., Kumar, D., Magnus, M., Chojnowski, G., & Bujnicki, J. M. (2016). RNA 3D Structure Modeling by Combination of Template-Based Method ModeRNA, Template-Free Folding with SimRNA, and Refinement with QRNAS. *Methods in Molecular Biology* (Clifton, N.J.), 1490(Suppl), 217-235. http://doi.org/10.1007/978-1-4939-6433-8_14

Right now, there is 20k steps of refinement.



Analysis output: all-atom contacts and geometry for 179c48aa-c0d3-4bd6-8e06-12081da22998_ALL_thrs6.20A_clust01-000001_AA.pdb



Duke Biochemistry
Duke University School of Medicine

Summary statistics

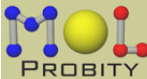
All-Atom Contacts	Clashscore, all atoms:	176.47	0 th percentile* (N=1784, all resolutions)	
	Clashscore is the number of serious steric overlaps (> 0.4 Å) per 1000 atoms.			
Nucleic Acid Geometry	Probably wrong sugar puckers:	4	6.45%	Goal: 0
	Bad backbone conformations [‡] :	21	33.87%	Goal: <= 5%
	Bad bonds:	60 / 1481	4.05%	Goal: 0%
	Bad angles:	161 / 2306	6.98%	Goal: <0.1%

In the two column results, the left column gives the raw count, right column gives the percentage.


* 100th percentile is the best among structures of comparable resolution; 0th percentile is the worst. For clashscore the comparative set of structures was selected in 2004, for MolProbity score in 2006.

[‡] RNA backbone was recently shown to be rotameric. Outliers are RNA suites that don't fall into recognized rotamers.

The initial structure, 179c48aa-c0d3-4bd6-8e06-12081da22998_ALL_thrs6.20A_clust01-000001_AA.pdb.



Analysis output: all-atom contacts and geometry for 179c48aa-c0d3-4bd6-8e06-12081da22998_ALL_thrs6.20A_clust01-000001_AA.refi1_trimmed.pdb



Duke Biochemistry
Duke University School of Medicine

Summary statistics

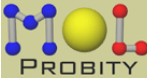
All-Atom Contacts	Clashscore, all atoms:	66.7	1 st percentile* (N=1784, all resolutions)	
	Clashscore is the number of serious steric overlaps (> 0.4 Å) per 1000 atoms.			
Nucleic Acid Geometry	Probably wrong sugar puckers:	7	11.29%	Goal: 0
	Bad backbone conformations [‡] :	20	32.79%	Goal: <= 5%
	Bad bonds:	24 / 1458	1.65%	Goal: 0%
	Bad angles:	235 / 2270	10.35%	Goal: <0.1%

In the two column results, the left column gives the raw count, right column gives the percentage.


* 100th percentile is the best among structures of comparable resolution; 0th percentile is the worst. For clashscore the comparative set of structures was selected in 2004, for MolProbity score in 2006.

[‡] RNA backbone was recently shown to be rotameric. Outliers are RNA suites that don't fall into recognized rotamers.

after 3k, ~10min



Analysis output: all-atom contacts and geometry for 179c48aa-c0d3-4bd6-8e06-12081da22998_ALL_thrs6.20A_clust01-000001_AA.refi_10k1_trimmed.pdb



Duke Biochemistry
Duke University School of Medicine

Summary statistics

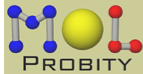
All-Atom Contacts	Clashscore, all atoms:	2.02	99 th percentile* (N=1784, all resolutions)	
	Clashscore is the number of serious steric overlaps (> 0.4 Å) per 1000 atoms.			
Nucleic Acid Geometry	Probably wrong sugar puckers:	7	11.29%	Goal: 0
	Bad backbone conformations [‡] :	15	24.59%	Goal: <= 5%
	Bad bonds:	2 / 1458	0.14%	Goal: 0%
	Bad angles:	35 / 2270	1.54%	Goal: <0.1%

In the two column results, the left column gives the raw count, right column gives the percentage.

* 100th percentile is the best among structures of comparable resolution; 0th percentile is the worst. For clashscore the comparative set of structures was selected in 2004, for MolProbity score in 2006.

[‡] RNA backbone was recently shown to be rotameric. Outliers are RNA suites that don't fall into recognized rotamers.

after 10k steps, around 30min



Analysis output: all-atom contacts and geometry for 179c48aa-c0d3-4bd6-8e06-12081da22998_ALL_thrs6.20A_clust01-000001_AA._refx.pdb



Duke Biochemistry
Duke University School of Medicine

Summary statistics

All-Atom Contacts	Clashscore, all atoms:	4.02	96 th percentile* (N=1784, all resolutions)	
	Clashscore is the number of serious steric overlaps (> 0.4 Å) per 1000 atoms.			
Nucleic Acid Geometry	Probably wrong sugar puckers:	6	9.68%	Goal: 0
	Bad backbone conformations [‡] :	12	19.67%	Goal: <= 5%
	Bad bonds:	1 / 1458	0.07%	Goal: 0%
	Bad angles:	17 / 2270	0.75%	Goal: <0.1%

In the two column results, the left column gives the raw count, right column gives the percentage.

* 100th percentile is the best among structures of comparable resolution; 0th percentile is the worst. For clashscore the comparative set of structures was selected in 2004, for MolProbity score in 2006.

[‡] RNA backbone was recently shown to be rotameric. Outliers are RNA suites that don't fall into recognized rotamers.

after 20k steps, around 1h.

Installation of QRNAS

Download the QRNAS package from <http://genesilico.pl/qrnas/>, unzip the archive, and compile it with the following command:

```
./qrnamake sequential
```

This should create an executable version of QRNAS.

Warning

Please, change the name of the binary file from QRNA to QRNAS!

By default the script searches QRNAS in <rna-tools>/opt/qrnas/.

Usage of QRNA:

```
QRNA - Quick Refinement of Nucleic Acids (0.2 alpha)
by Juliusz Stasiewicz (jstasiewicz@genesilico.pl)
```

To use **type**:

```
QRNA -i <input PDBfile> [-o <output PDBfile>] [-c <configfile>] [-p] [-m
↪<restraintsfile>]
```

OR specify <input PDBfile>, <output PDBfile> **and** <restraintsfile> **in** <configfile> **and** ↪**type** just:

```
QRNA -c <configfile>
```

Installation of this util

Set up in your bashrc:

```
export QRNAS_PATH=<your path to qrnas> # e.g. /home/magnus/src/rna-tools/opt/qrnas
```

but default rna-tools searches for qrnas in <rna-tools>/opt/qrnas.

QRNAS at Peyote2

There is no problem to run QRNAS at our Genesilico cluster, *peyote2*. Tested by mmagnus -170822. Copy files of QRNAS to peyote and run `./qrnamake sequential`.

To run it at a cluster with the Sun Grid Engine queuing system (this one with qsub ;-)):

```
for p in *.pdb; do echo "rna_refinement.py $p >& ${p}.log" | qsub -cwd -V -pe mpi 1 -N  
↪ "r_$p" ; done
```

DONE:

- [x] clean up the output structure
- [x] configuration should not be hardcoded

```
usage: rna_refinement.py [-h] [-s STEPS] [-o OUTPUT_FILE] [-i] [-v] fn
```

fn

input pdb file

-h, --help

show this help message and exit

-s <steps>, --steps <steps>

of steps, default: 20k

-o <output_file>, --output_file <output_file>

output pdb file

-i, --interactive

-v, --verbose

4.12 RNA Molecular Dynammmics (MD)

4.13 diffpdb

diffpdb - a simple tool to compare text-content of PDB files

The method is quick-and-dirty, but works!

The script takes first 31 characters of lines (or only atom names and residue names) starting with HETATM or ATOM and save these lines to a <filename>.out file.

One file is created per pdb. In the final step DIFF_TOOL is executed on these two output files. You get a diff output. That's it! Enjoy!

Configuration:

- DIFF_TOOL="open -a diffmerge" or DIFF_TOOL="kompare" to set up what tool would you like to use to diff files in the file rna-pdb-tools/tools/diffpdb/diffpdb_conf.py (create it if needed)

File: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/1/1kxkA.pdb.out -- file: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/1/1kxkA_M1.pdb.out -- Kompare

Navigation

Source Folder: /home/magnus/Dropbox/WORK/_workspace/rnastruc/... Destination Folder: /home/magnus/Dropbox/WORK/_workspace/rnastruc/...

Source File: 1kxkA.pdb.out Destination File: 1kxkA_M1.pdb.out

Source Line: 5, 28, 48, 68 Destination Line: 5, 28, 48, 68 Difference: Changed 16 lin, Changed 13 lin, Changed 13 lin

Line	Atom	X	Y	Z	Occupancy	Element	Chain	Residue
1	ATOM	1	05'	G	A	1		
2	ATOM	2	C5'	G	A	1		
3	ATOM	3	C4'	G	A	1		
4	ATOM	4	O4'	G	A	1		
5	ATOM	5	C3'	G	A	1		
6	ATOM	6	O3'	G	A	1		
7	ATOM	7	C2'	G	A	1		
8	ATOM	8	O2'	G	A	1		
9	ATOM	9	C1'	G	A	1		
10	ATOM	10	N9	G	A	1		
11	ATOM	11	C8	G	A	1		
12	ATOM	12	N7	G	A	1		
13	ATOM	13	C5	G	A	1		
14	ATOM	14	C6	G	A	1		
15	ATOM	15	O6	G	A	1		
16	ATOM	16	N1	G	A	1		
17	ATOM	17	C2	G	A	1		
18	ATOM	18	N2	G	A	1		
19	ATOM	19	N3	G	A	1		
20	ATOM	20	C4	G	A	1		
21	ATOM	21	P	U	A	2		
22	ATOM	22	OP1	U	A	2		
23	ATOM	23	OP2	U	A	2		
24	ATOM	24	O5'	U	A	2		
25	ATOM	25	C5'	U	A	2		
26	ATOM	26	C4'	U	A	2		
27	ATOM	27	O4'	U	A	2		
28	ATOM	28	C3'	U	A	2		
29	ATOM	29	O3'	U	A	2		
30	ATOM	30	C2'	U	A	2		
31	ATOM	31	O2'	U	A	2		
32	ATOM	32	C1'	U	A	2		
33	ATOM	33	N1	U	A	2		
34	ATOM	34	C2	U	A	2		
35	ATOM	35	O2	U	A	2		
36	ATOM	36	N3	U	A	2		
37	ATOM	37	C4	U	A	2		
38	ATOM	38	O4	U	A	2		
39	ATOM	39	C5	U	A	2		
40	ATOM	40	C6	U	A	2		
41	ATOM	41	P	C	A	3		

Comparing File: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/1/1kxkA.pdb.out -- file: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/1/1kxkA_M1.pdb.out 1 of 87 differences, 0 applied 1 of 1 file

```
./diffpdb.py --names test_data/4/1duq.pdb test_data/4/1duq_decoy0171_amb_clx.pdb
```

File: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/4/1duq.pdb.out -- file: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/4/1duq_decoy0171_amb_clx.pdb.out -- Kompare

Navigation

Source Folder: /home/magnus/Dropbox/WORK/_workspace/rnastruc/... Destination Folder: /home/magnus/Dropbox/WORK/_workspace/rnastruc/...

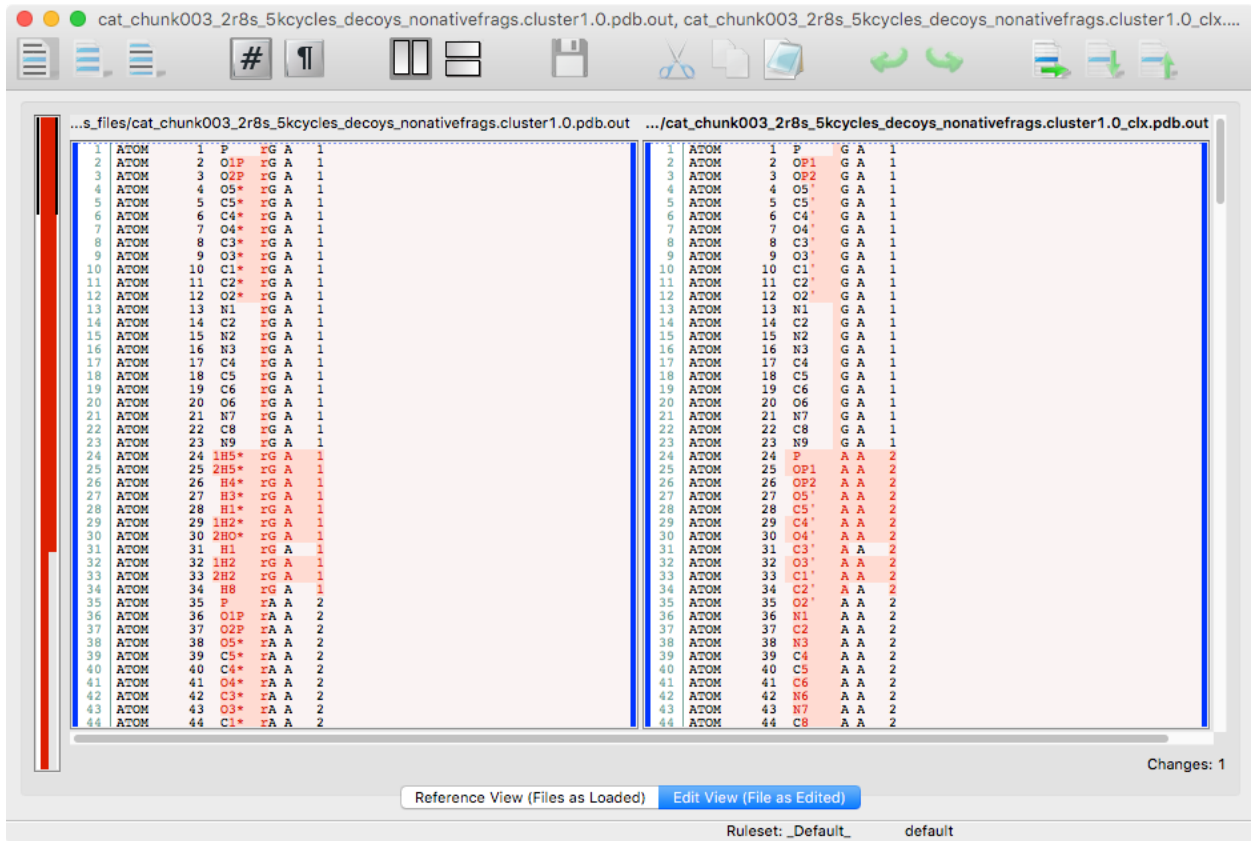
Source File: 1duq.pdb.out Destination File: 1duq_decoy0171_amb_clx.p...

Source Line: 1, 3, 6, 10 Destination Line: 1, 5, 9, 12 Difference: Changed 1, Inserted 1, Deleted 1, Changed 1

Line	Atom	X	Y	Z	Occupancy	Element	Chain	Residue
1	C5'	G						
2	O5'	G						
3	C4'	G						
4	O4'	G						
5	C3'	G						
6	O3'	G						
7	C2'	G						
8	O2'	G						
9	C1'	G						
10	N1	G						
11	C2	G						
12	N2	G						
13	N3	G						
14	C4	G						
15	C5	G						
16	C6	G						
17	O6	G						
18	N7	G						
19	C8	G						
20	N9	G						
21	P	C						
22	C5'	C						
23	O5'	C						
24	C4'	C						
25	O4'	C						
26	C3'	C						
27	O3'	C						
28	C2'	C						
29	O2'	C						
30	C1'	C						
31	N1	C						

Comparing File: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/4/1duq.pdb.out -- file: //home/magnus/Dropbox/WORK/_workspace/rnastruc/lib/diffpdb/test_data/4/1duq_decoy0171_amb_clx.pdb.out 1 of 133 differences, 0 applied 1 of 1 file

and on the Mac (using diffmerge):

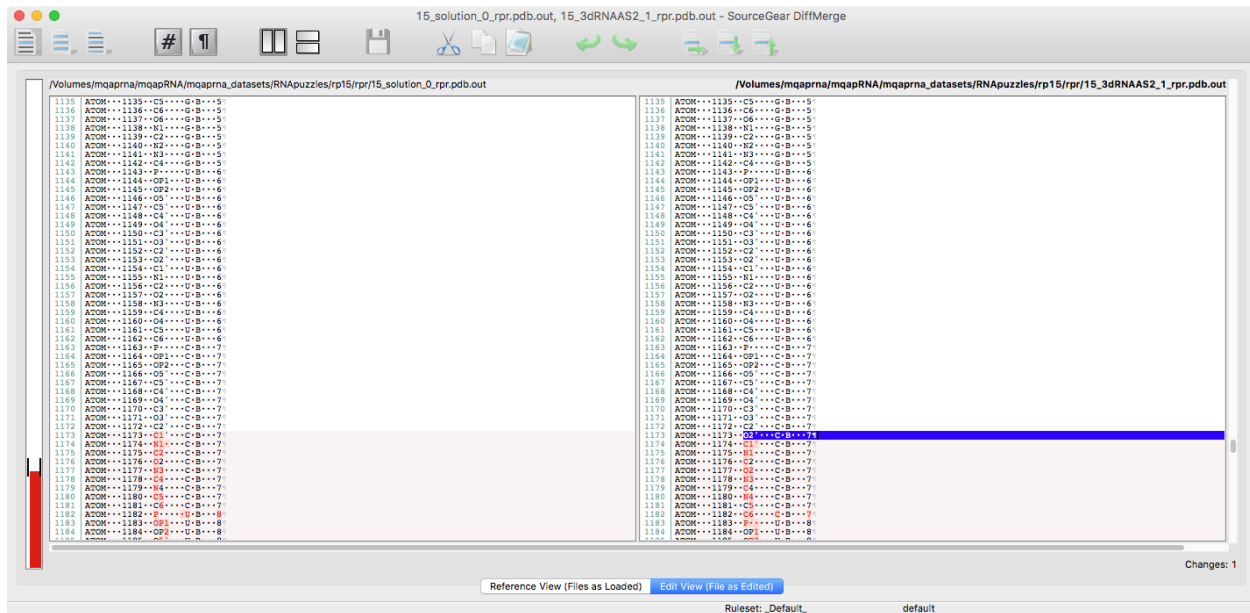


One of the difference that can be detected with the script is variants of atoms.

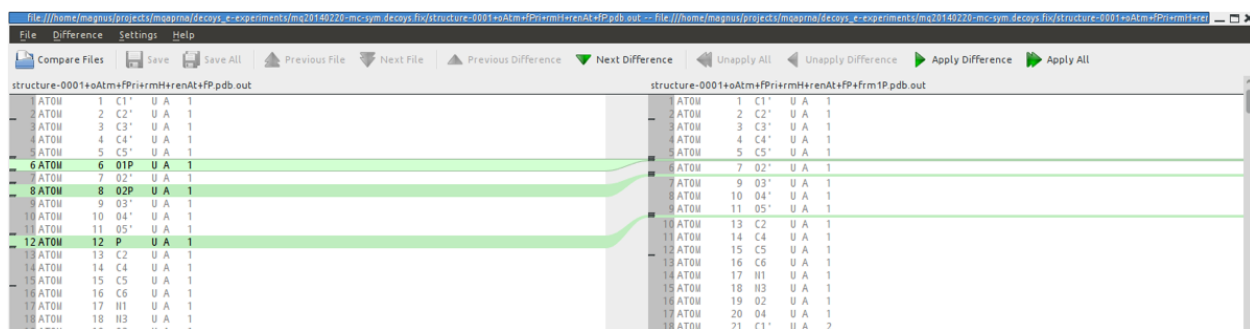
```

320 C4...A.A..16%
321 P..A..A.A..18%
322 P..B..A.A..18%
323 OP1A..A.A..18%
324 OP1B..A.A..18%
325 OP2A..A.A..18%
326 OP2B..A.A..18%
327 O5'A..A.A..18%
328 O5'B..A.A..18%
    
```

or a detection of missing atom.



or a detection of missing OP3 at the beginning.



4.14 RNA clustering with CLANS (clanstix)

rna_clanstix - a tool for visualizing RNA 3D structures based on pairwise structural similarity with Clans.

We hacked Clans thus instead of BLAST-based distances between sequences, you can analyze distances between structures described as p-values of rmsd (based on the method from the Dokholyan lab.)

Quickref:

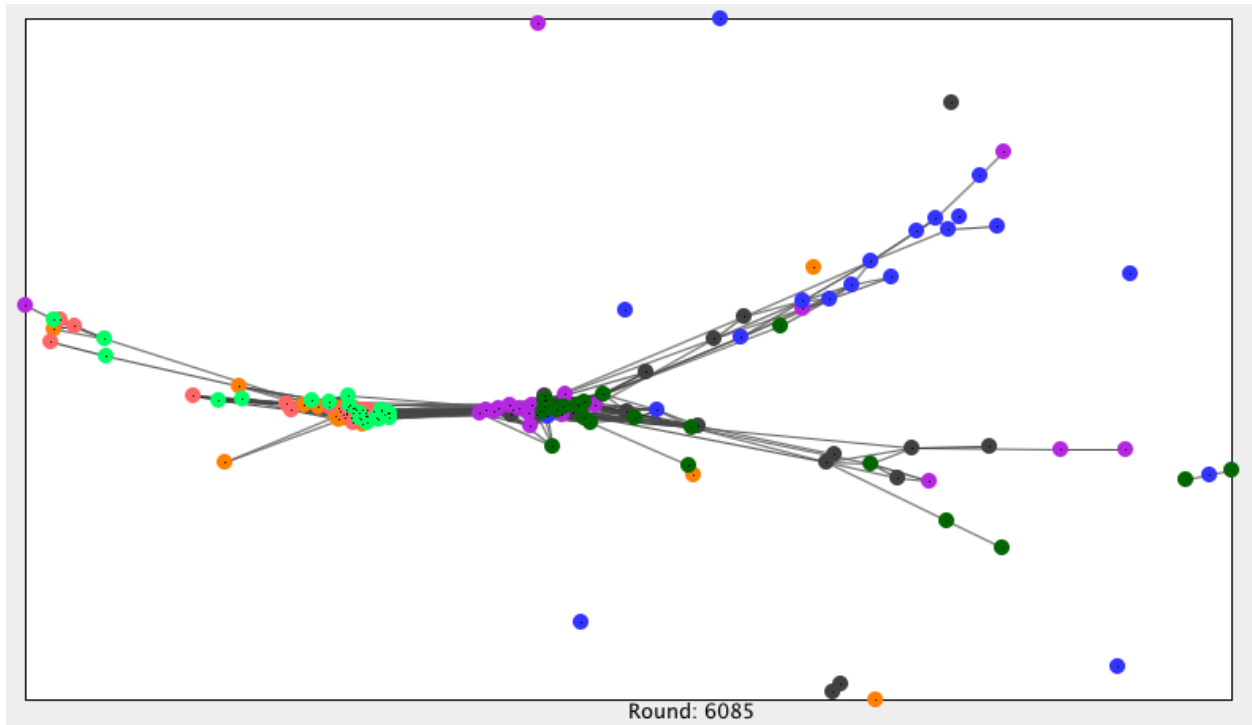
```
rna_clanstix.py --groups-auto 10 --color-by-homolog --shape-by-source thf_ref_mapping_
↳pk_refX.txt input2.clans
```

Running Clans: To run CLANS you need to have Java 1.4 or better installed (java can be downloaded [HERE](#)). For full functionality you will also need the NCBI BLAST,PSI-BLAST and formatdb executables (NCBI). For command line parameters and basic help please refer to the README file. (source: <http://www.eb.tuebingen.mpg.de/research/departments/protein-evolution/software/clans.html>)

The RMSDs between structures are converted into p-values based on the method from the Dokholyan lab or some hacky way developed by mmagnus .

4.14.1 Color groups

You can color your groups:



To get colors, run a cmd like this:

```
rna_clastix.py rnapz17_matrix_farfar_HelSeedCst.txt --groups_
↔20:seq1+20+20+20+20+20+20:seq10
```

where with the + sign you separate groups. Each group has to have a number of structures. Optionally it can have a name, e.g., 20:seq1, use : as a separator. If a provided name is native then this group will be shown as starts.

Get inspiration for more colors (http://www.rapidtables.com/web/color/RGB_Color.htm)

4.14.2 How to use ClanstixRNA?

1. Get a matrix of distances, save it as e.g. matrix.txt (see Comment below)
2. run ClanstixRNA on this matrix to get an input file to Clans (e.g. clans_rna.txt):

```
rna_clanstix.py test_data/matrix.txt # clans.input will be created by default
```

3. open CLANS and click File -> Load run and load clans_run.txt
4. You're done! :-)

Comment: To get this matrix you can use for example another tool from the rna-pdb-tools packages:

```
rna_calc_rmsd_all_vs_all.py -i rp18 -o rp18_rmsd.csv
rna_clastix.py --groups 1:native+5:3dRNA+
5:Chen+3:Dokh+5:Feng+5:LeeASModel+
5:Lee+5:RNAComposer+10:RW3D+5:Rhiju+
```

(continues on next page)

(continued from previous page)

```
1:YagoubAli+3:SimRNA rp18_rmsd.csv clans.in
rna_clastix.py --groups 100+100+100+100+100+100+100+100+100+100+1:native rp18_rmsd.csv
```

where rp18 is a folder with structure and rp18_rmsd.csv is a matrix of all-vs-all rmsds.



Hajdin, C. E., Ding, F., Dokholyan, N. V., & Weeks, K. M. (2010). On the significance of an RNA tertiary structure prediction. *RNA (New York, N.Y.)*, 16(7), 1340–9. doi:10.1261/rna.1837410

An output of this tool can be viewed using CLANS.

Frickey, T., & Lupas, A. (2004). CLANS: a Java application for visualizing protein families based on pairwise similarity. *Bioinformatics (Oxford, England)*, 20(18), 3702–4. doi:10.1093/bioinformatics/bth444

class rna_tools.tools.clanstix.rna_clanstix.RNAStructClans(*n=10, dotsize=10*)

Clans run.

Usage:

```
>>> f = open('matrix.txt')
>>> ids = f.readline().replace('#', '').split()
>>> c = RNAStructClans(n=len(ids)) # 200?
>>> c.add_ids(ids)
>>> c.dist_from_matrix(f)
>>> print(c.txt)
```

add_ids(*ids*)

dist_from_matrix(*rmsds, matrix=0, use_pv=False, use_input_values=False, dont_calc=False, debug=False*)

dist_from_matrix_mp(*output_pmatrix_fn, max, min, lines, pmat=False, use_pv=False, use_input_values=False, debug=False*)

```
rna_tools.tools.clanstix.rna_clanstix.check_symmetric(a, rtol=1e-05, atol=1e-08)
https://stackoverflow.com/questions/42908334/checking-if-a-matrix-is-symmetric-in-numpy
rna_tools.tools.clanstix.rna_clanstix.get_parser()
```

4.15 Misc

4.15.1 Plotting

4.15.1.1 rna_plot_hist.py

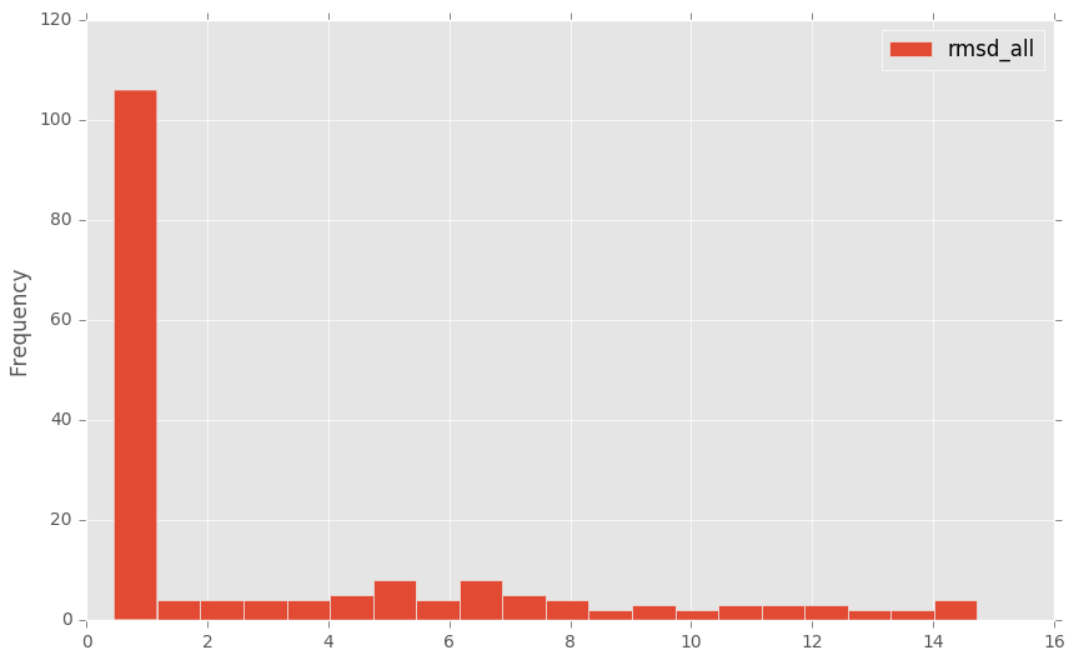
rna_plot_hist.py - generate a histogram

Don't open Excel, Jupyter. Simple plot a histogram of one column and save it to a file.

Example:

```
# file
                                fn  rmsd_all
0  19_Bujnicki_Human_4_rpr_n0-000001.pdb-000001_A...  14.73
1  19_Bujnicki_Human_4_rpr_n0-000001.pdb.trafl_19...   0.46
2  19_Bujnicki_Human_4_rpr_n0-000001.pdb.trafl_19...  14.73
3  19_Bujnicki_Human_4_rpr_n0-000001.pdb_thrs0.50...  0.73
4  19_Bujnicki_Human_4_rpr_n0-000001.pdb_thrs0.50...  0.83

$ rna_plot_hist.py rmsds.csv --column rmsd_all
```



```
usage: rna_plot_hist.py [-h] [--column COLUMN] [--sep SEP] [-o OUTPUT]
                        [--bins BINS]
                        file
```

file

rmsd.txt

-h, --help

show this help message and exit

--column <column>

column of file to plot

--sep <sep>

separator, be default

-o <output>, --output <output>**--bins <bins>****4.15.1.2 rna_plot_density.py**

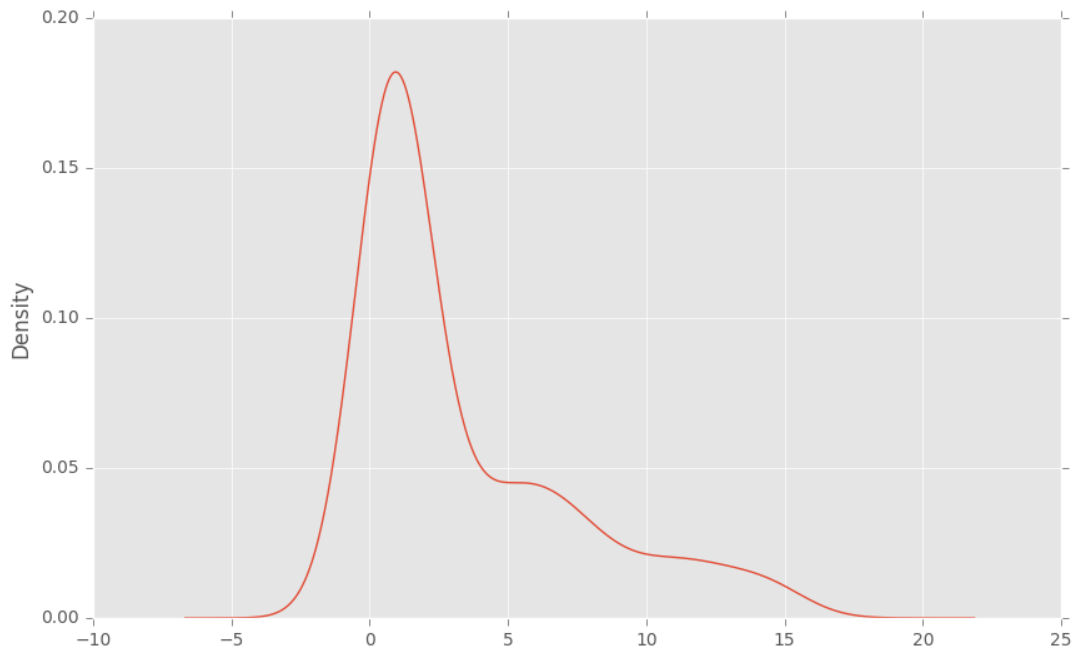
rna_plot_density.py - generate a density plot

Don't open Excel, Jupyter. Simple plot a density of one column and save it to a file.

Example:

```
# file
                                fn  rmsd_all
0  19_Bujnicki_Human_4_rpr_n0-000001.pdb-000001_A...  14.73
1  19_Bujnicki_Human_4_rpr_n0-000001.pdb.trafl_19...   0.46
2  19_Bujnicki_Human_4_rpr_n0-000001.pdb.trafl_19...  14.73
3  19_Bujnicki_Human_4_rpr_n0-000001.pdb_thrs0.50...  0.73
4  19_Bujnicki_Human_4_rpr_n0-000001.pdb_thrs0.50...  0.83

$ rna_plot_hist.py rmsds.csv --column rmsd_all
```



```
usage: rna_plot_density.py [-h] [--column COLUMN] [--sep SEP] [-o OUTPUT] file
```

file

rmsd.txt

-h, --help

show this help message and exit

--column <column>

column of file to plot

--sep <sep>

separator, be default

-o <output>, --output <output>

4.15.2 rna_sali2dotbracket

4.15.2.1 rna_sali2dotbracket

```
usage: rna_sali2dotbracket [-h] filename
```

filename

file in the Sali format

-h, --help

show this help message and exit

This beauty here will go to sali notation and convert it to dotbracket notation. The file name should be xxxx.sali

Author: Catarina Almeida

`rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket.convert_sali2dotbracket(fn)`

The function needs a filename in the Sali format. This function will get the secondary structure of the sequence, then get its identifier and then the sequence itself.

To get the ss

The line with the secondary structure is a list and will look like this:

```
['', '', '', '', '', '', '', '', '', '', '--...<<[[...]]..>>>', '', '', '\n']
```

In this case, the ss is in the 11th position. But in some files it may be in the 12th, 13th, 10th, etc..

If the longest element from the list is extracted, then this problem is overcome.

The ss will some times have patterns of repeated gaps, which will come in the form of:

- a. x
- b. xnt
- c. (x)

With x being any number, from 1 to 1000. These must be converted to the correspondent number of gaps (-) in the converted ss. This conversion is done by:

- 1 - Identifying the pattern with regex
- 2 - Replacing it with repl function.

As such, the following expressions will replace the previously mentioned patterns:

- a. `re.sub(r'\d*\d', repl, temp)`
- b. `re.sub(r'\d*\dnt', repl, temp)`
- c. `re.sub(r'(?P<smthBeautiful>\(\d+\))', repl, temp)`

To get the sequence

The sequence, much like the ss, can sometimes be in a different position in the list. Like in the ss, the longest element will be selected. Also, like in the ss, patterns for repeated gaps appear. So these must also be removed.

`rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket.get_parser()`

`rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket.repl(m)`

This function will replace the length of a given string by the correspondent number of dashes. The expression `qwerty` will be replaced by `-----`.

4.15.3 Cluster load

A very simple tool to see your cluster load per user:

```
MAX_JOBS: 1000
#jobs cluster 917 load: 0.917 to use: 83
#jobs you 749 load: 0.749 to use: 251
{'deepak': 160, 'azyla': 8, 'magnus': 749}
1 azyla r 8
20 magnus r 10
16 deepak r 10
```

(continues on next page)

(continued from previous page)

```
329 magnus      r  1
22  magnus      qw 10
```

A super simple script to get some statistics of who is running at a cluster

Set MAX_JOBS to calc % of usage, it's an approximation of max number of jobs, e.g. peyote ~1k (rather 700, e.g. FARNA runs.).

```
rna_tools.tools.cluster_load.cluster_load.get_parser()
```

```
rna_tools.tools.cluster_load.cluster_load.per_user()
```

get stats (#cpus) per user

```
rna_tools.tools.cluster_load.cluster_load.stats_for_cluster()
```

get stats (#jobs) per cluster

```
rna_tools.tools.cluster_load.cluster_load.stats_for_user()
```

get stats (#jobs) per user

4.15.4 RNAkb

RNA-TOOLS & PYMOL [PYMOL4RNA]

See also https://github.com/mmagnus/rna-tools/tree/master/rna_tools/tools/PyMOL4RNA

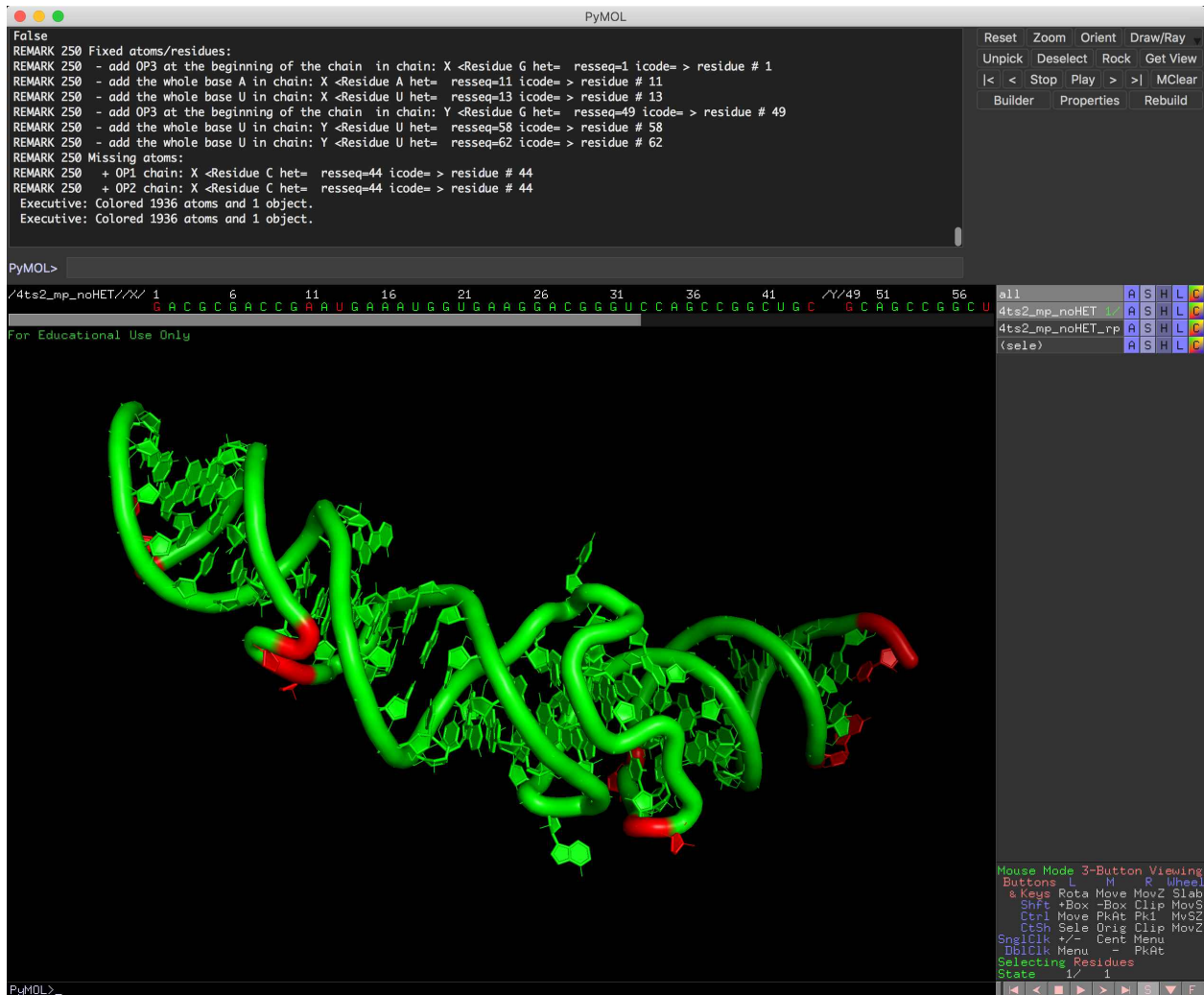
5.1 Inspect structure

There is new `--inspect` function for `rna_pdb_tools.py` that tells you what are missing atoms in your structure file.

You can run it also from PyMOL, so in here you can see missing atoms directly in PyMOL (the object comes green and the residues missing atoms comes in red).

PyMOL:

```
inspect <object name>
```



and here you can compare the input file vs file after --rpr (of rna_pdb_tools.py) to see what was rebuilt (in pink).

PyMOL

```

False
REMARK 250 Fixed atoms/residues:
REMARK 250 - add OP3 at the beginning of the chain in chain: X -Residue G het= resseq=1 icode= > residue # 1
REMARK 250 - add the whole base A in chain: X -Residue A het= resseq=11 icode= > residue # 11
REMARK 250 - add the whole base U in chain: X -Residue U het= resseq=13 icode= > residue # 13
REMARK 250 - add OP3 at the beginning of the chain in chain: Y -Residue G het= resseq=49 icode= > residue # 49
REMARK 250 - add the whole base U in chain: Y -Residue U het= resseq=58 icode= > residue # 58
REMARK 250 - add the whole base U in chain: Y -Residue U het= resseq=62 icode= > residue # 62
REMARK 250 Missing atoms:
REMARK 250 + OP1 chain: X -Residue C het= resseq=44 icode= > residue # 44
REMARK 250 + OP2 chain: X -Residue C het= resseq=44 icode= > residue # 44
Executive: Colored 1936 atoms and 1 object.
Executive: Colored 1936 atoms and 1 object.

```

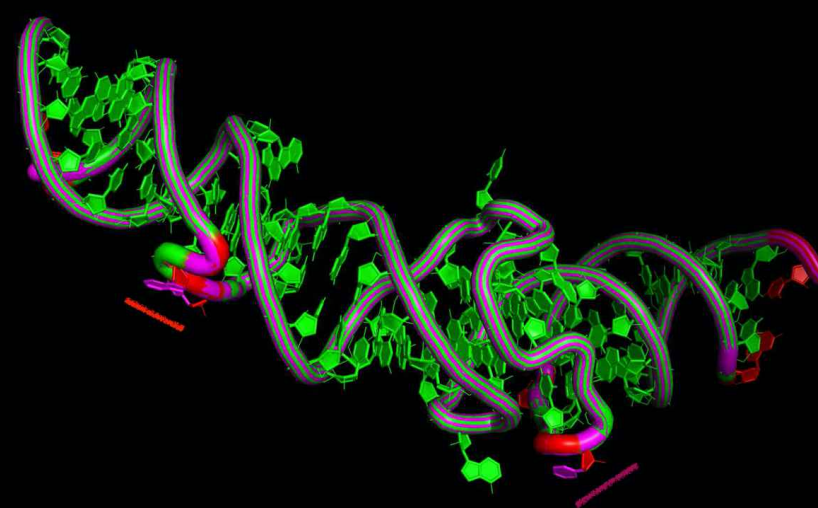
PyMOL>

```

/4ts2_mp_noHET// 1 6 11 16 21 26 31 36 41 /Y/49 51 all A S H L C
G A C G C G A C C G A A U G A A A U G G U G A A G G A C C G G U C C A G C C G G C U G C G C A G C C G G
/4ts2_mp_noHET_rpr// 1 6 11 16 21 26 31 36 41 /Y/49 51 4ts2_mp_noHET_rpr A S H L C
G A C G C G A C C G A A U G A A A U G G U G A A G G A C C G G U C C A G C C G G C U G C G C A G C C G G

```

For Educational Use Only



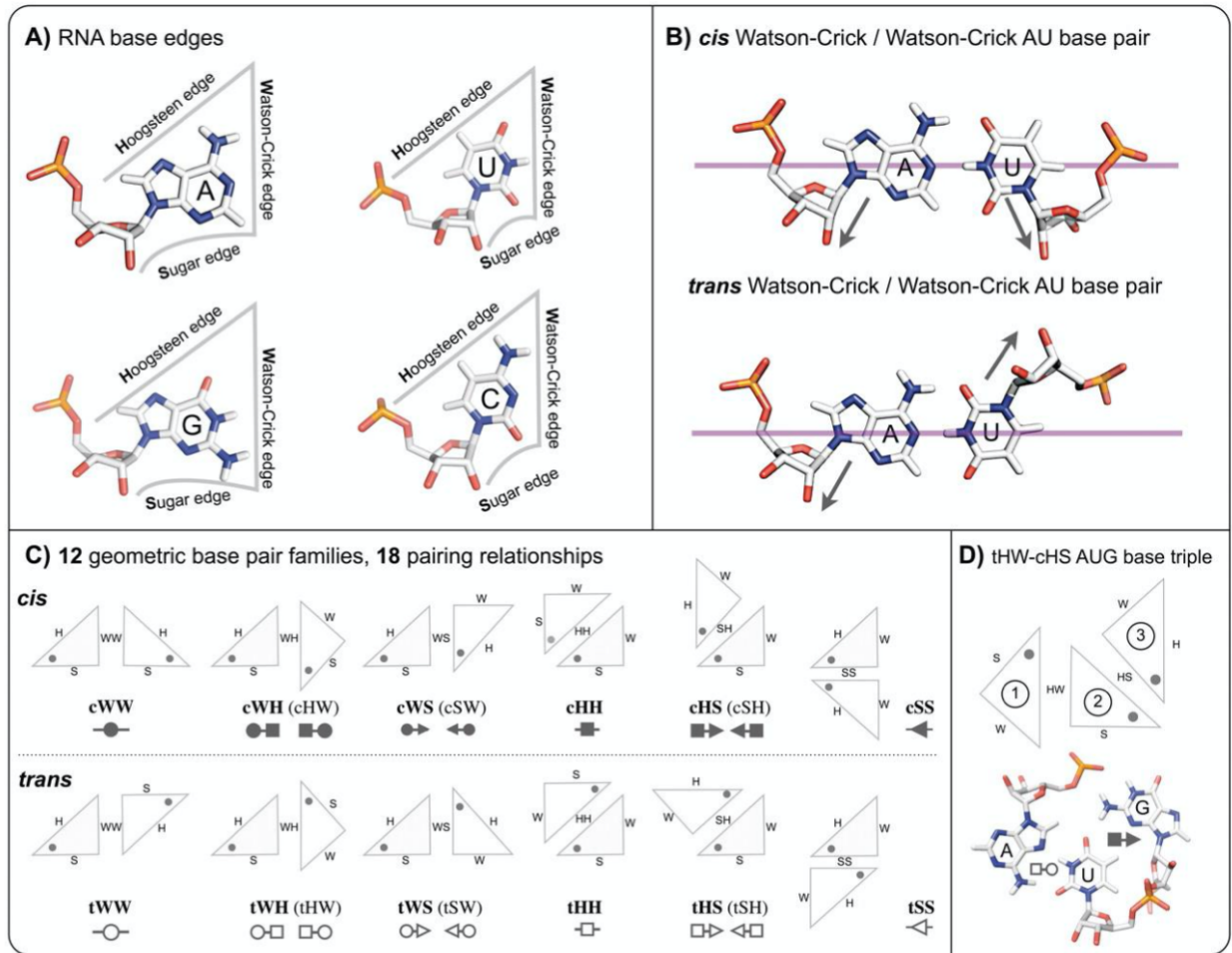
PyMOL>

```

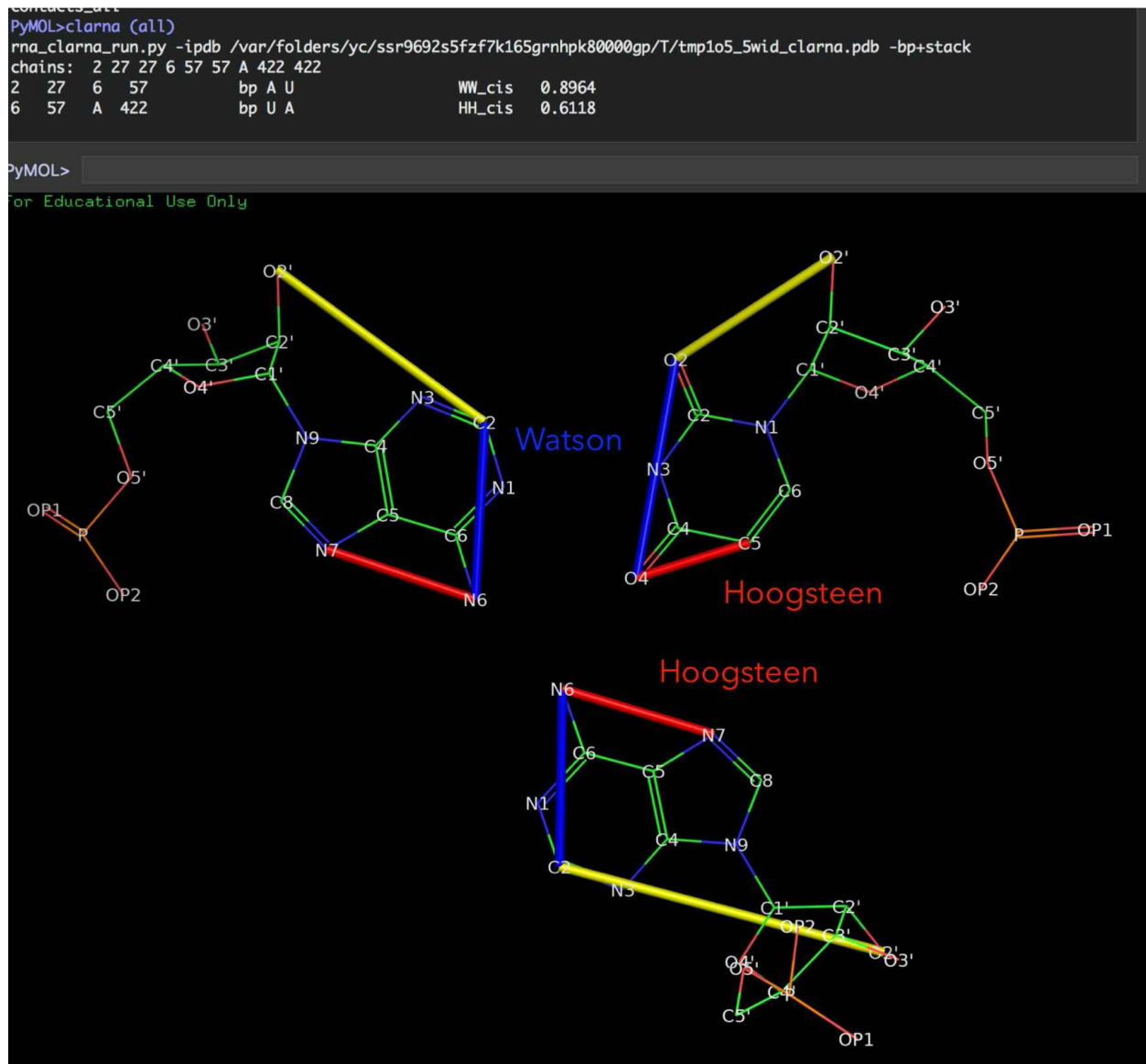
Mouse Mode 3-Button Viewing
Buttons L M R Wheel
% Keys Rota Move Mov2 Slab
Shift +Box -Box Clip MovS
Ctrl Move PkAt Pk1 MovSZ
EtsH Sele Orig Clip MovZ
SnglClk +/- Cent Menu
DblClk Menu - PkAt
Selecting Residues
State 1/ 1

```

5.2 Show base pair edges



Leontis/Westhof classification of base pairings. (A) RNA bases - adenine (A), cytosine (C), guanine (G) and uracil (U) - involve one of three distinct edges: the Watson–Crick (W) edge, the Hoogsteen (H) edge, and the Sugar (S) edge. (B) Each pair of can interact in either *cis* or *trans* orientations with respect to the glycosidic bonds. (C) For these reasons, all base pairs can be grouped into twelve geometric base pair families and eighteen pairing relationships (bases are represented as triangles). Each pair is represented by a symbol that can be used in a secondary structure and a tertiary structure diagrams. Filled symbols mean *cis* base pair configuration, and open symbols, *trans* base pair. (D) Interestingly, bases can form triples and they have own classification devised by Leontis and coworkers (Creative Commons License) (A. S. Abu Almakarem, A. I. Petrov, J. Stombaugh, C. L. Zirbel, and N. B. Leontis, “Comprehensive survey and geometric classification of base triples in RNA structures.,” *Nucleic Acids Research*, vol. 40, no. 4, pp. 1407–1423, Feb. 2012.)



⚠ Warning

This PyMOL function is very simple and works on standard atom naming, so to make sure that everything will work, get your structure in the RNA Puzzle ready format, See <https://rna-tools.readthedocs.io/en/latest/main.html#get-rnapuzzle-ready>

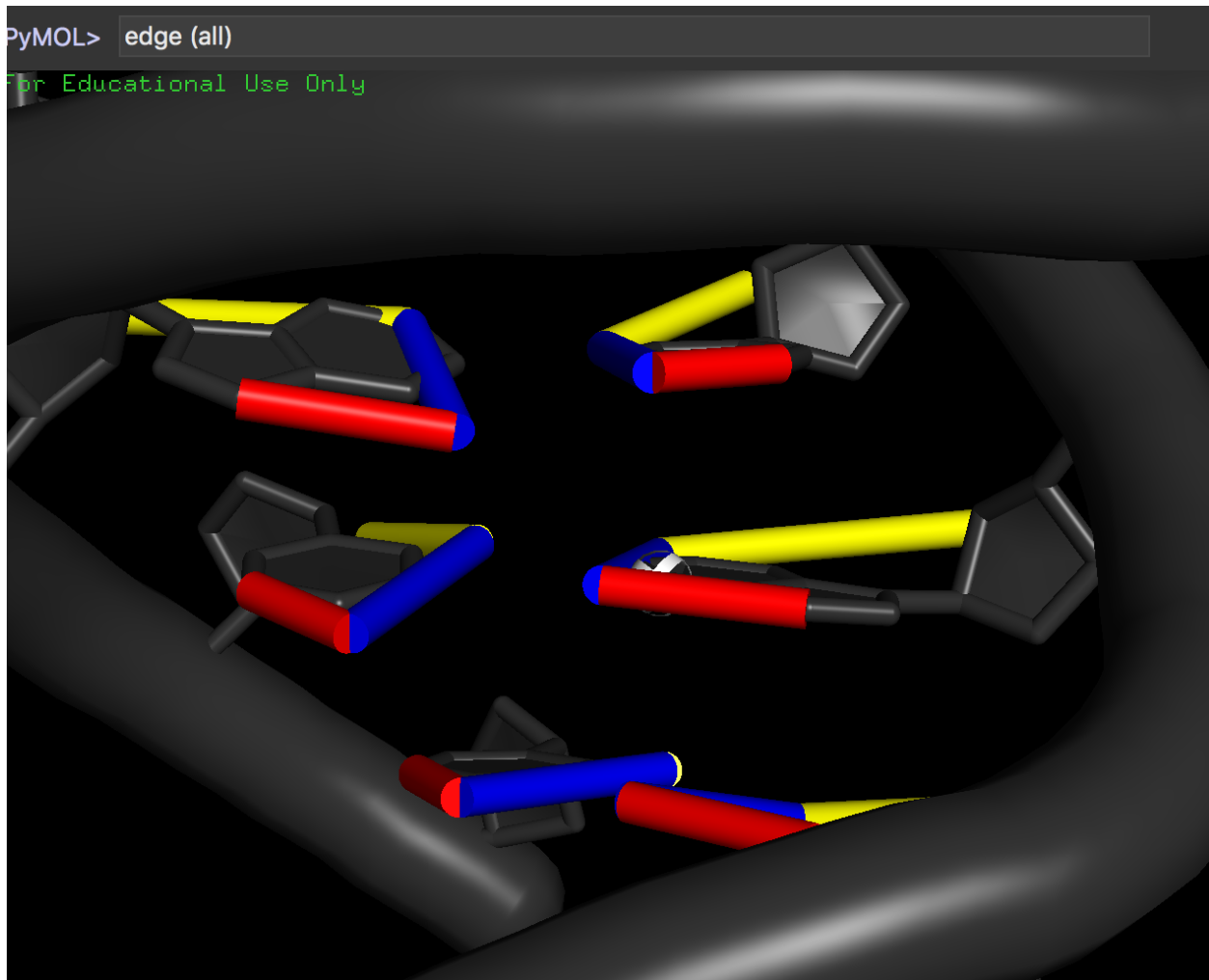
Type:

edges (<selection>)

e.g.,:

edges (all)

to show the edges drawn on bases.



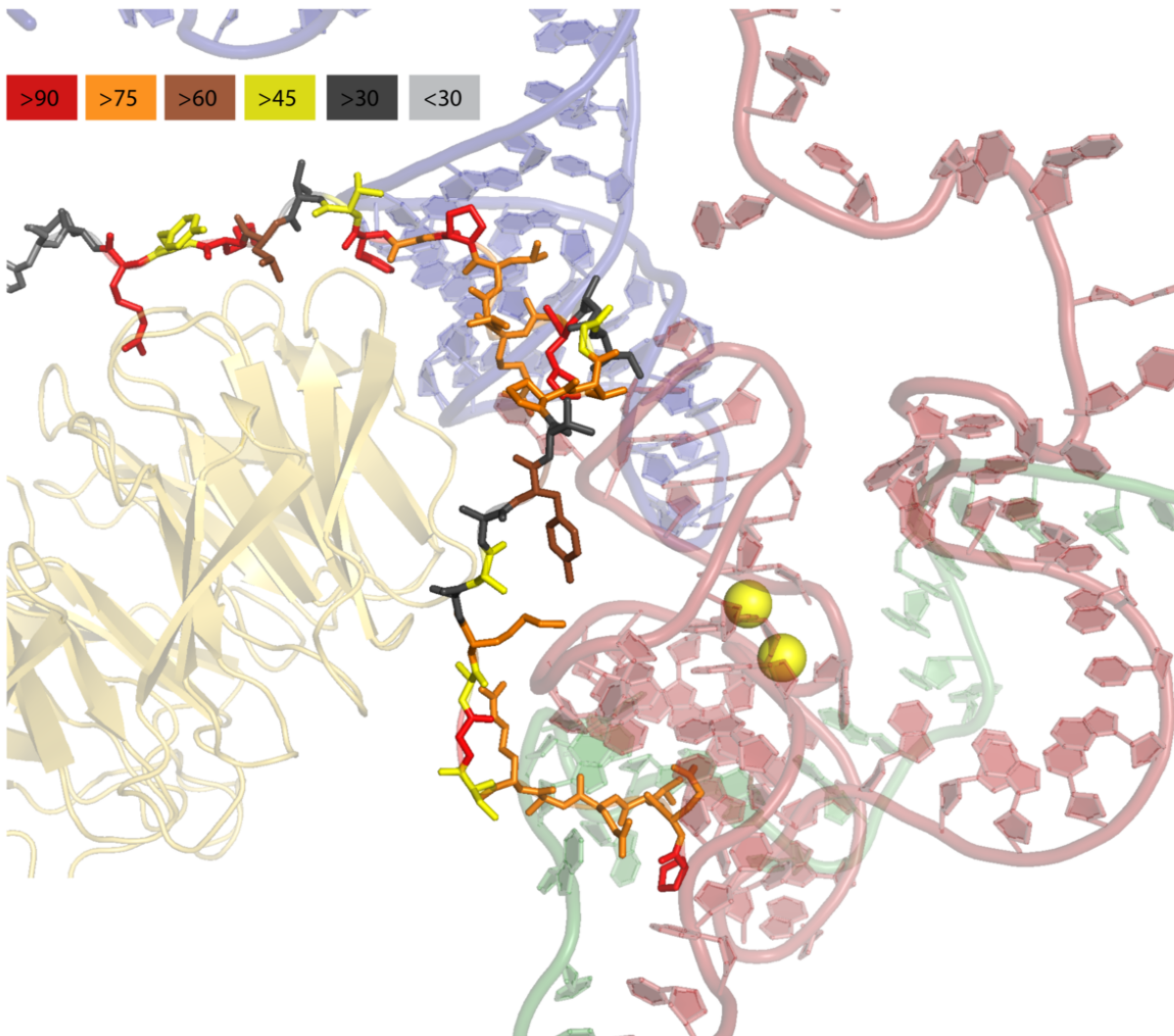
5.3 PyMOL4Spliceosome: all spliceosome structures in one PyMOL session

Download one PyMOL session with structures for each step.

See own project <https://github.com/mmagnus/PyMOL4Spliceosome>

5.4 PyMOL: Color by conservation

Show conserved regions of proteins in PyMOL.



See more: https://github.com/mmagnus/rna-tools/tree/master/rna_tools/tools/pymol_color_by_conserv

5.5 PyMOL Drawing

```
rna_tools.tools.pymol_drawing.pymol_drawing.draw_circle(x, y, z, r=8.0, cr=1.0, cg=0.4, cb=0.8, w=2.0)
```

Create a CGO circle

PARAMS

x, y, z
X, Y and Z coordinates of the origin

r
Radius of the circle

cr, cg, cb

Color triplet, [r,g,b] where r,g,b are all [0.0,1.0].

w

Line width of the circle

RETURNS

the CGO object (it also loads it into PyMOL, too).

`rna_tools.tools.pymol_drawing.pymol_drawing.draw_circle_selection(selName, r=None, cr=1.0, cg=0.4, cb=0.8, w=2.0)`

circleSelection – draws a cgo circle around a given selection or object

PARAMS

selName

Name of the thing to encircle.

r

Radius of circle. DEFAULT: This script automatically defines the radius for you. If you select one atom and the resultant circle is too small, then you can override the script's calculation of r and specify your own.

cr, cg, cb

red, green and blue coloring, each a value in the range [0.0, 1.0]

RETURNS

The circle object.

`rna_tools.tools.pymol_drawing.pymol_drawing.draw_dist(54.729, 28.9375, 41.421, 55.342, 35.3605, 42.745)`

<https://sourceforge.net/p/pymol/mailman/message/25795427/>

`rna_tools.tools.pymol_drawing.pymol_drawing.draw_dists(interactions)`

`rna_tools.tools.pymol_drawing.pymol_drawing.draw_vector(x1, y1, z1, x2, y2, z2)`

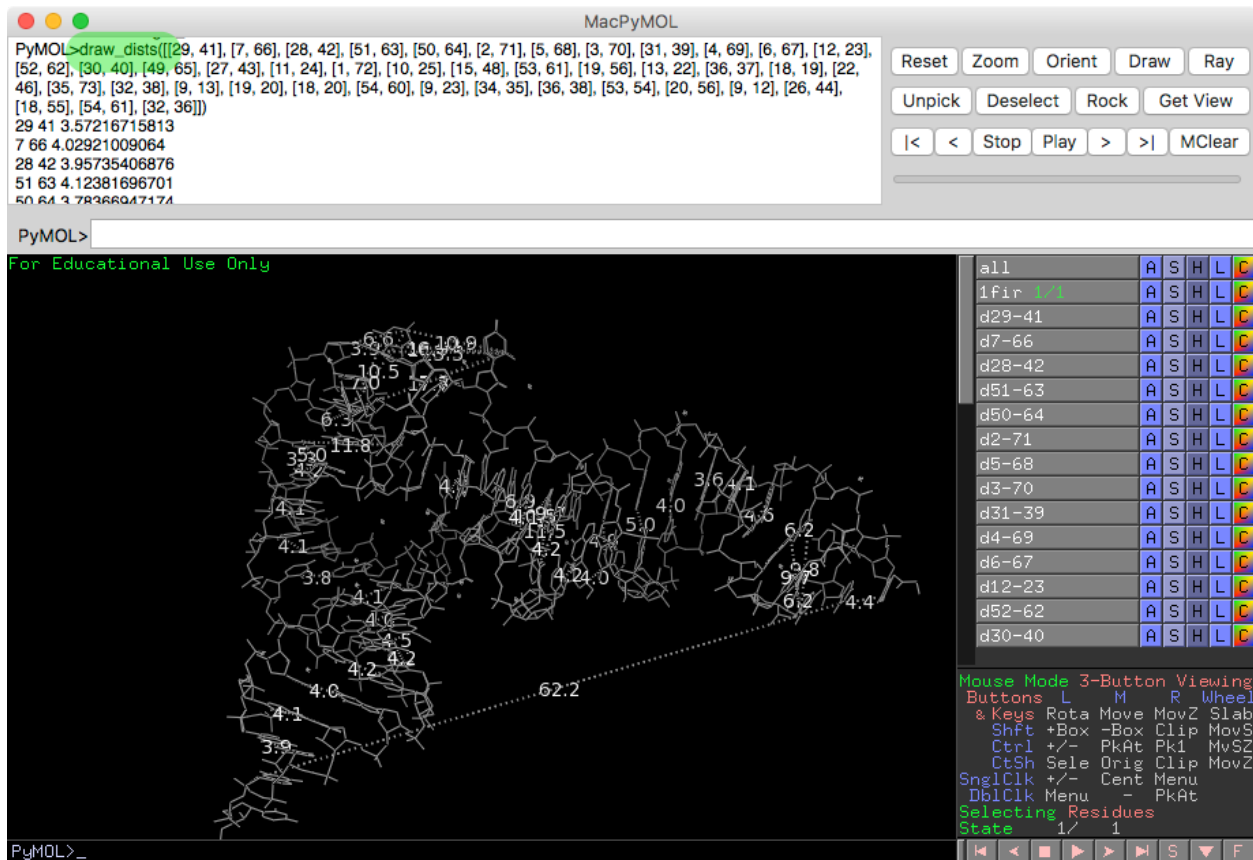
<https://pymolwiki.org/index.php/CGOCylinder>

Install PyMOL plugin to view the interactions with PyMOL:

```
run <path>rna-tools/tools/pymol_drawing/pymol_dists.py
```

and type:

```
draw_dists([[29, 41], [7, 66], [28, 42], [51, 63], [50, 64], [2, 71], [5, 68], [3, 70], ↵
↪ [31, 39], [4, 69], [6, 67], [12, 23], [52, 62], [30, 40], [49, 65], [27, 43], [11, 24],
↪ [1, 72], [10, 25], [15, 48], [53, 61], [19, 56], [13, 22], [36, 37], [18, 19], [22, ↵
↪ 46], [35, 73], [32, 38], [9, 13], [19, 20], [18, 20], [54, 60], [9, 23], [34, 35], [36,
↪ 38], [53, 54], [20, 56], [9, 12], [26, 44], [18, 55], [54, 61], [32, 36]])
```



5.6 Install

After you install rna-tools, run these two lines your terminal:

```
$ echo "sys.path.append(`rna_tools_which.py --site`)" >> ~/.pymolrc
$ echo "PyMOL4RNA_PATH=`rna_tools_which.py`/tools/PyMOL4RNA/PyMOL4RNA.py"; cmd.
↳run(PyMOL4RNA_PATH)" >> ~/.pymolrc
```

so something like this will be added to your ~/.pymolrc:

```
sys.path.append('/Users/magnus/work/src/rna-tools')
PyMOL4RNA_PATH='/Users/magnus/work/src/rna-tools/rna_tools/tools/PyMOL4RNA/PyMOL4RNA.py';
↳ cmd.run(PyMOL4RNA_PATH)
```

5.7 Libs: Pucker analysis

pucker.py is a PyMol script that returns the sugar pucker information (phase, amplitude, pucker) for a given selection. This script uses its own dihedral calculation scheme rather than the `get_dihedral` command. Thus, it is lightning fast! If a selection does not contain any ribose sugars then an error message is returned.

Author: Sean Law (Institute: University of Michigan) adapted for Python 3 by magnus <mmagnus@fas.harvard.edu>.

Install, add `run <path>/rna-tools/rna_tools/tools/PyMOLRNA/libs/pucker.py` to your `~/pymolrc`

Source: <<https://pymolwiki.org/index.php/Pucker>>

```
PyMOL>pucker (sele)
```

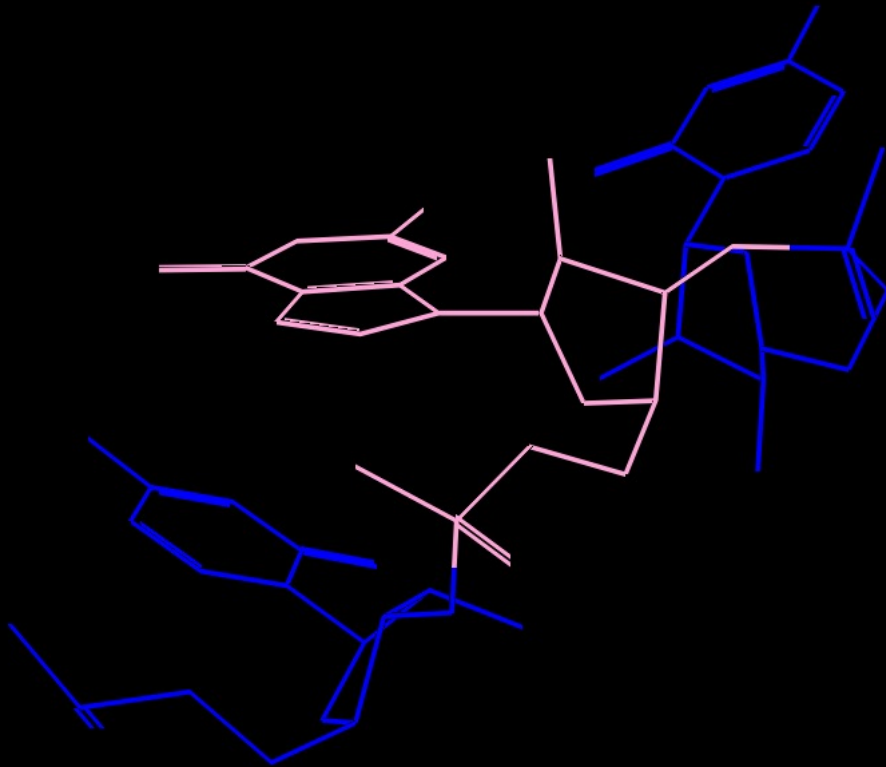
Phase	Amp	Pucker	Residue
17.63	40.64	C3'-endo	A 18
170.88	41.82	C2'-endo	A 19
30.37	50.65	C3'-endo	A 20

```
PyMOL>
```

```
/obj02//A/ 18
```

```
 C G C
```

```
For Educational Use Only
```



RNA-TOOLS & MAC/LINUX

6.1 PyMOL preview generator

https://github.com/mmagnus/rna-tools/tree/master/rna_tools/tools/pymol_preview_generator

Youtube video: <https://www.youtube.com/watch?v=mNgW7sekgug>

RNA-TOOLS & JUPYTER NOTEBOOKS / COLAB

The tools collected here as rna-tools, ideally could be also used in IPython/Jupyter (<https://ipython.org/notebook.html>) Notebooks. We believe it would be valuable if we could, as scientists share our notebooks used for RNA structure analysis, e.g, protocols of modeling used in the RNA Puzzle challenge.

```
In [2]: seq = Seq.Seq("GGGUCAGGCCGGCGAAAGUCGCCACAGUUUGGGGAAAGCUGUGCAGCCUGUAACCCCCCAGAAAGUGGG")
```

```
In [3]: seq
```

```
Out[3]: GGGUCAGGCCGGCGAAAGUCGCCACAGUUUGGGGAAAGCUGUGCAGCCUGUAACCCCCCAGAAAGUGGG
```

Secondary structure prediction

```
In [12]: print seq.predict_ss()
```

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

```
In [13]: print seq.predict_ss(method='RNAsubopt')
```

```
GGGUCAGGCCGGCGAAAGUCGCCACAGUUUGGGGAAAGCUGUGCAGCCUGUAACCCCCCAGAAAGUGGG -3310 100  
(((((((((.((((.....))))(((((.....))))))..))))..))))..(((.....)))) -32.40  
(((((((((.((((.....))))(((((.....))))))..))))..))))..(((.....)))) -33.10  
(((((((((.((((.....))))..(((.....))))))..))))..))))..(((.....)))) -32.30
```

```
In [14]: print seq.predict_ss(method='ipknot')
```

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

```
In [15]: print seq.predict_ss(method='centroid_fold')
```

```
(((((((((((.....))))..(((.....))))))..))))..(((.....)))) (g=1,th=0.5,e=-27.26)
```

```
In [16]: print seq.predict_ss(method='contextfold')
```

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

See more at <https://github.com/mmagnus/rna-tools/blob/master/rp18.ipynb> <https://github.com/mmagnus/rna-tools/blob/master/rp19.ipynb>

7.1 Share your notebooks

You can share Jupyter notebooks from your Google Drive using the new Jupyter Drive. This allows you share Jupyter notebooks like NBViewr with all the access control that Google Drive provides. You can also push your notebooks to a Github repository, so then can be rendered for easy viewing.

E.g., <https://github.com/mmagnus/rna-tools/blob/master/rp18.ipynb>

7.2 Use Colab

Some of the functions of rna-tools will work also on Colab.

E.g., ClaRNA <https://colab.research.google.com/drive/1rcPz7ZLWpn-hbfJo8hpz6ZuMGiGGmKeW>

7.3 Learn Jupyter

This is a quick introduction to jupyter which is the IPython version 3. It covers some of the new and interesting features about Jupyter.

<https://www.youtube.com/watch?v=Rc4JQWowG5I>

RNA-TOOLS & EMACS

Warning

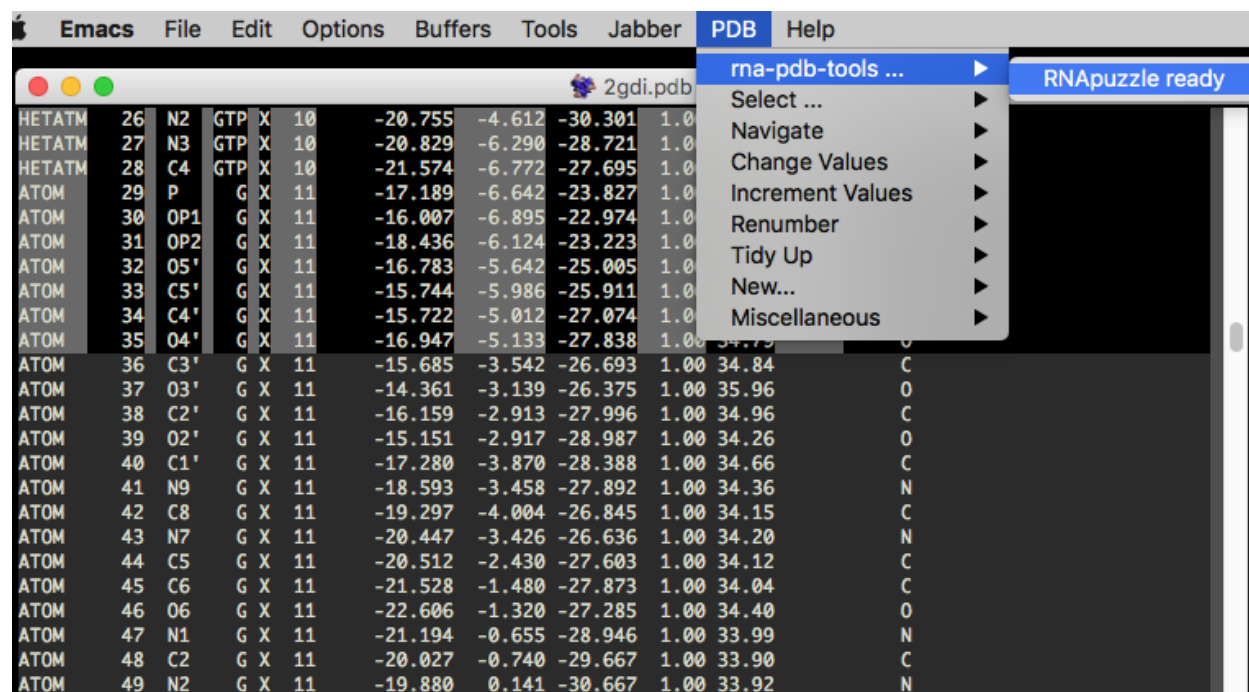
Proof of concept

rna-tools can be used side-by-side <https://github.com/mmagnus/emacs-pdb-mode> to edit files structural files in the PDB format.

Emacs /imæks/ and its derivatives are a family of text editors that are characterized by their extensibility.[2] The manual for the most widely used variant, GNU Emacs, describes it as “the extensible, customizable, self-documenting, real-time display editor”. Development of the first Emacs began in the mid-1970s and continues actively as of 2017. (<https://en.wikipedia.org/wiki/Emacs>)

pdb-mode (<https://github.com/mmagnus/emacs-pdb-mode> not authored by me, I’m a maintainer, and a beginner developer) is an emacs-lisp minor mode for Emacs to perform a number of useful editing functions on Protein DataBank (PDB) formatted files. XEmacs and/or GNU Emacs are available for most computing platforms.

Youtube video: <https://www.youtube.com/embed/o99YFbLSVRw>



RNA-TOOLS & GEEKBOOK

 **Warning**

Proof of concept

G33KB00K - eXtreme eXtensible note taking system for nerds/geeks (including scientists!) (= beautiful html generator of your markdown-based notes) docs: <http://geekbook.rtfid.io>

Marcin Magnus (mmagnus) & Pietro Boccaletto (akaped)

The code of the project can be found at GitHub (<https://github.com/mmagnus/geekbook>).

A neat way how to combine Emacs/Atom/Sublime editor + Markdown Syntax + Git + Html engine (bootstrap/python) to get the best notes-taking experience ever. Highly customizable with plugins written in Python. What's the most important, under the hood it's just a set of Markdown files.. you can do with them whatever you want, e.g. you can Pandoc (<http://pandoc.org/epub.html>) them to epub (that's origin of "book" part of the name).

9.1 Draw VARNA-based image of RNA secondary structure

Type:

```
<pre>[ss:rna]
UUUCUGUAUAAUAGCCGAUAAUAAGGUUCGGCAGUUUCUACCAAACAGCCGUAAACUGUUUGACUACAGUAA
((.((((((...((((((.....)))))))).).....((((((.....)))))))).).....))
</pre>
```

 **Warning**

Keep exactly the same syntax as in the example above and below.

The syntax:

```
<pre>
[ss:/name of your seq/]
/seq/
/ss/
</pre>
# ^ not <pre/> nor <pre>. Keep a new line after this syntax. So don't do:
```

(continues on next page)

(continued from previous page)

```
</pre>
this could be fixed at some point
```

 **Warning**

This plugin will change your Markdown file, so make sure that your editor will detect this change and ask you to reload the file!

to get a VARNA-drawn image of secondary structure.

RNA PUZZLE SUBMISSION

The RNA Puzzle organizers required ONE file with your submissions in the NMR-style multiple model PDB format. First, prepare your structures in the folder and run to get them RNApuzzle ready (*_rpr*):

```
$ for i in *.pdb; do rna_pdb_tools.py --get-rnapuzzle-ready $i > ${i}/.pdb/_rpr.pdb; done
```

merge them as one file in the order as you like (or use *):

```
$ rna_pdb_merge_into_one.py 02_19pz_v1_SimRNA3.22_thrs6.60A_clust02-000001_AA_out_rpr.
→pdb \
09_19pz_v2_SimRNA3.22_thrs6.60A_clust03-000001_AA_out_rpr.pdb \
d311d821-a075-4df0-bd7d-1dcf7669dad9_ALL_thrs6.20A_clust01-000001_AA_out_rpr.pdb \
d311d821-a075-4df0-bd7d-1dcf7669dad9_ALL_thrs6.20A_clust03-000001_AA_out_rpr.pdb \
05_19pz_v1_SimRNA4.xx_thrs6.60A_clust02-000001_AA_out_rpr.pdb > rp19_bujnicki.pdb
```

and verify your file with the template provided by the organizers (if provided):

```
$ diffpdb --method diff Reference_19.pdb rp19_bujnicki.pdb
#<empty = no difference but xyz columns, OK!>
```

diffpdb is a part of the rna-tools package.

For [Educational Use Only](#)

In terminal:

```

$ head -n 5 Reference_19.pdb rp19_bujnicki.pdb
==> Reference_19.pdb <==
MODEL      1
ATOM       1  P   G A   1      0.000  0.000  0.000  1.00  0.00      P
ATOM       2  OP1 G A   1      0.000  0.000  0.000  1.00  0.00      O
ATOM       3  OP2 G A   1      0.000  0.000  0.000  1.00  0.00      O
ATOM       4  O5'  G A   1      0.000  0.000  0.000  1.00  0.00      O
==> rp19_bujnicki.pdb <==
MODEL      1
ATOM       1  P   G A   1     31.463  14.180  -0.676  1.00  0.00      P
ATOM       2  OP1 G A   1     31.412  12.806  -1.223  1.00  0.00      O
ATOM       3  OP2 G A   1     30.646  15.083  -1.517  1.00  0.00      O
ATOM       4  O5'  G A   1     30.955  14.212   0.842  1.00  0.00      O

$ tail -n 5 Reference_19.pdb rp19_bujnicki.pdb
==> Reference_19.pdb <==
ATOM     1325  C5   C B  22      0.000  0.000  0.000  1.00  0.00      C
ATOM     1326  C6   C B  22      0.000  0.000  0.000  1.00  0.00      C
TER      1327          C B  22
ENDMDL
END
==> rp19_bujnicki.pdb <==
ATOM     1325  C5   C B  22     29.927  21.506  -6.542  1.00  0.00      C
ATOM     1326  C6   C B  22     29.822  22.338  -5.500  1.00  0.00      C
TER      1327          C B  22
ENDMDL
END

```

The Bujnicki lab is using this script to process the results and send the results, you can hack it for your own case:

```

[mm] rnapuzzle_sender$ git:(master) ./rnapuzzle_sender.py
rnapuzzle_sender

Usage: rnapuzzle_sender.py

Options:
  -h, --help          show this help message and exit
  -d DIR, --dir=DIR
  -s EMAIL_SUBJECT, --email_subject=EMAIL_SUBJECT
                       email subject

```

File: *rna-tools/rna_tools/tools/rnapuzzle_sender*.

SELECTION

These functions are intended to be imported to tools to give a unified framework for making selections.

```
rna_tools.tools.extra_functions.select_fragment.select_pdb_fragment(txt, separator='-',  
                                                                    splitting='[:\\+]',  
                                                                    verbose=False)
```

Take txt such as A:1-31+B:1-11 and parse into:

```
OrderedDict([('A', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,  
15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]),  
('B', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])])
```

 **Warning**

e.g. for A:1-31, resi 31 is included

```
rna_tools.tools.extra_functions.select_fragment.select_pdb_fragment_pymol_style(txt)
```

Take txt such as A/10-15/P and parse into:

```
A/57/02' -> ['A', ['57'], "02"]
```

If you want to combine a few subselections, please use ,:

```
--model_ignore_selection "A/57/02',A/58/02'"
```

 **Warning**

e.g. for A:1-31, resi 31 is included

12.1 Run in batch

You can easily run a single tool in batch and rename new files:

```
$ for i in *.pdb; do rna_pdb_tools.py --get_rnapuzzle_ready $i > ${i}/.pdb/_rpr.pdb; done
```

or write new files in a different folder (*out*):

```
$ for i in *.pdb; do rna_pdb_tools.py --get_rnapuzzle_ready $i > ../out/$i; done
```

You can also easily run a single tool parallel using **parallel_**:

```
$ parallel "rna_add_chain.py -c A {} > ../nchain/{}" ::: *.pdb  
# ex2  
$ parallel "rna_clashscore.py {} > {}.csv" ::: *.pdb
```

12.2 Using sed

sed (stream editor) is a Unix utility that parses and transforms text, using a simple, compact programming language.

You can use sed to find & replace parts of text files:

```
$ head 1msy_rnakbmd_decoy1661_clx.pdb.outCR  
Classifier: Clarna  
chains: 1 27  
  2      26      bp G U      WW_cis  0.8500  
  3      25      bp C G      WW_cis  0.8114  
  4      24      bp U A      WW_cis  0.9222  
  5      23      bp C G      WW_cis  0.9038  
  6      22      bp C G      WW_cis  0.8913  
  9      10      bp G U      SH_cis  0.8563  
 10      19      bp U A      WH_tran 0.7826  
 11      18      bp A G      HS_tran 0.7620  
  
$ sed 's/chains: /chains: A/' 1msy_rnakbmd_decoy1661_clx.pdb.outCR  
Classifier: Clarna  
chains: A 1 27  
  2      26      bp G U      WW_cis  0.8500
```

(continues on next page)

(continued from previous page)

3	25	bp C G	WW_cis	0.8114
4	24	bp U A	WW_cis	0.9222
5	23	bp C G	WW_cis	0.9038
6	22	bp C G	WW_cis	0.8913
9	10	bp G U	SH_cis	0.8563
10	19	bp U A	WH_tran	0.7826
11	18	bp A G	HS_tran	0.7620
12	17	bp C G	WW_cis	0.7242

Read more about [sed](#).

12.3 In PyMOL

Quickref:

```
set ignore_case, off
```

Rename a chain:

```
PyMOL>alter (sele), chain="B"  
Alter: modified 708 atoms.  
PyMOL>sort
```

don't forget about *sort*.

To select all, use *PyMOL>alter all, resv -= 12*.

To renumber a fragment starting with 24 to 29, select the fragment and:

```
PyMOL>alter (sele), resv += 5  
Alter: modified 109 atoms.
```

To renumber residues:

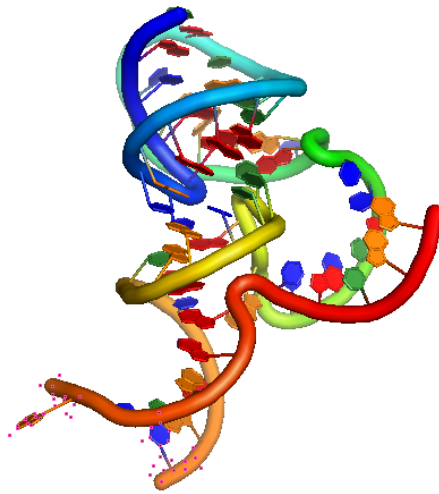
```
PyMOL>alter (chain B), resv -= 44  
Alter: modified 708 atoms.  
PyMOL>sort
```

Renumber again, your chain starts at 123, renumber it, so it starts at 1.

```
PyMOL>alter (sele), resv -= 122 Alter: modified 2921 atoms.
```

Read [more](#).

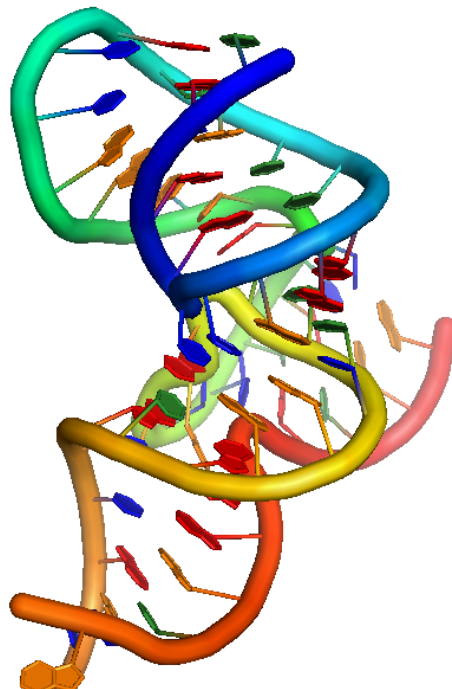
The example of the pistol ribozyme editing.



```
/5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped 16      21      26      31      36      41      46 52      56      61
C G U G G U U A G G G C C A C G U U A A A U A G U U G C U U A G C C C U A R G C G U U G A U U C A G G U G C A A
```

Run:

```
PyMOL>alter (sele), chain="B"
Alter: modified 236 atoms.
PyMOL>alter (chain B), resv == 51
Alter: modified 236 atoms.
PyMOL>sort
```



```
/5k7c_clean_onechain_renumber_as_puzzle_rpr_rmGapped 21      26      31      36      41      46 /B/1      6
C G U G G U U A G G G C C A C G U U A A A U A G U U G C U U A G C C C U A R G C G U U G A U U C A G G U G C A A
```

12.4 In Python

To get residue index use:

```
resi = int(l[22:26].strip())
```

Example:

```
ATOM      1  P      C A      1      -16.936  -3.789  68.770  1.00 11.89      P
```

Qucikref:

COLUMNS	PYTHON	DATA	TYPE	FIELD	DEFINITION
1 - 6	[0:6]	Record name		"ATOM "	
7 - 11	[6:11]	Integer		serial	Atom serial number.
13 - 16	[12:16]	Atom		name	Atom name.
17	[16]	Character		altLoc	Alternate location indicator.
18 - 20	[17:20]	Residue name		resName	Residue name.
22	[21]	Character		chainID	Chain identifier.
23 - 26	[22:26]	Integer		resSeq	Residue sequence number.
27	[26]	AChar		iCode	Code for insertion of residues.
31 - 38	[30:38]	Real(8.3)		x	Orthogonal coordinates for X in Angstroms.
39 - 46	[38:46]	Real(8.3)		y	Orthogonal coordinates for Y in Angstroms.
47 - 54	[46:54]	Real(8.3)		z	Orthogonal coordinates for Z in Angstroms.
55 - 60	[54:60]	Real(6.2)		occupancy	Occupancy.
61 - 66	[60:66]	Real(6.2)		tempFactor	Temperature factor.
77 - 78	[76:78]	LString(2)		element	Element symbol, right-justified. # 1[76:78]
79 - 80	[78:80]	LString(2)		charge	Charge on the atom.

ATOM lines in PDB files

column number	1	2	3	4	5	6	7	8			
ATOM	1	N	GLY	A	3	17.119	0.186	36.320	1.00	64.10	N
ATOM	2	CA	GLY	A	3	16.944	-0.800	35.208	1.00	63.46	C
ATOM	3	C	GLY	A	3	16.818	-0.087	33.851	1.00	61.22	C
ATOM	4	O	GLY	A	3	15.721	0.337	33.463	1.00	62.81	O
ATOM	5	N	PRO	A	4	17.944	0.077	33.129	1.00	57.39	N
ATOM	6	CA	PRO	A	4	17.950	0.742	31.815	1.00	53.27	C
ATOM	7	C	PRO	A	4	18.005	-0.247	30.629	1.00	49.78	C
ATOM	8	O	PRO	A	4	19.086	-0.678	30.218	1.00	48.17	O
ATOM	9	CB	PRO	A	4	19.191	1.613	31.898	1.00	54.33	C
ATOM	10	CG	PRO	A	4	20.161	0.686	32.625	1.00	55.45	C
ATOM	11	CD	PRO	A	4	19.305	0.019	33.701	1.00	55.83	C

field id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
----------	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

field id	definition	length	format	range	string slicing (Python)
1	"ATOM " or "HETATM"	6	%-6s	01-06	[0:6]
2	atom serial number	5	%5d	07-11	[6:11]
3	atom name	4	%4s	13-16	[12:16]
4	alternate location indicator	1	%1s	17	[16:17]
5	residue name	3	%3s	18-20	[17:20]
6	chain identifier	1	%1s	22	[21:22]
7	residue sequence number	4	%4d	23-26	[22:26]
8	code for insertion of residues	1	%1s	27	[26:27]
9	orthogonal coordinates for X (in Angstroms)	8	%8.3f	31-38	[30:38]
10	orthogonal coordinates for Y (in Angstroms)	8	%8.3f	39-46	[38:46]
11	orthogonal coordinates for Z (in Angstroms)	8	%8.3f	47-54	[46:54]
12	occupancy	6	%6.3f	55-60	[54:60]
13	temperature factor	6	%6.3f	61-66	[60:66]
14	element symbol	2	%2s	77-78	[76:78]
15	charge on the atom	2	%2s	79-80	[78:80]

(source: <http://cupnet.net/pdb-file-atom-line-memo/>)

12.5 Working with cluster

Tips:

```
# get your pdb files
[mm] ade rsync -v peyote2:'~/ade/*.pdb' . # ' is required!
```

See long name with qstat:

```
magnus@peyote2:~$ qstat -xml | tr '\n' ' ' | sed 's#<job_list[^>]*>#\n#g' \
> | sed 's#<[^>]*>##g' | grep " " | column -t
4752204 5.54737 r_6bd26658_run_04          magnus dr 2017-02-20T22:09:04 all.
↳q@c6.cluster3.genesilico.pl 10
4752201 5.54737 r_6bd26658_run_01          magnus dr 2017-02-20T22:09:04 all.
↳q@c6.cluster3.genesilico.pl 10
4752203 5.54737 r_6bd26658_run_03          magnus dr 2017-02-20T22:09:04 all.
↳q@c6.cluster3.genesilico.pl 10
4752202 5.54737 r_6bd26658_run_02          magnus dr 2017-02-20T22:09:04 all.
↳q@c6.cluster3.genesilico.pl 10
4805710 5.54737 r_hTERC_251-451-85d4ac69_run_01 magnus r 2017-08-20T17:04:15 all.
↳q@c11.cluster3.genesilico.pl 10
4805711 5.54737 r_hTERC_251-451-85d4ac69_run_02 magnus r 2017-08-20T17:04:15 all.
↳q@c11.cluster3.genesilico.pl 10
4805712 5.54737 r_hTERC_251-451-85d4ac69_run_03 magnus r 2017-08-20T17:04:15 all.
↳q@c8.cluster3.genesilico.pl 10
4805713 5.54737 r_hTERC_251-451-85d4ac69_run_04 magnus r 2017-08-20T17:04:15 all.
↳q@c8.cluster3.genesilico.pl 10
4805714 5.54737 r_hTERC_251-451-85d4ac69_run_05 magnus r 2017-08-20T17:04:15 all.
↳q@c8.cluster3.genesilico.pl 10
4805715 5.54737 r_hTERC_251-451-85d4ac69_run_06 magnus r 2017-08-20T17:04:15 all.
↳q@c8.cluster3.genesilico.pl 10
4805716 5.54737 r_hTERC_251-451-85d4ac69_run_07 magnus r 2017-08-20T17:04:15 all.
↳q@c8.cluster3.genesilico.pl 10
4805728 5.54737 r_mCherry_sub3-3c970489_run_03 magnus r 2017-08-20T17:21:15 all.
↳q@c15.cluster3.genesilico.pl 10
```

<https://stackoverflow.com/questions/26104116/qstat-and-long-job-names>

12.6 Numbering line used in my flat-file notes

Numbering:

```
|1.....|10.....|20.....|30.....|40.....|50.....|60.....|70.....|80.....
↳|90.....
123456789112345678921234567893123456789412345678951234567896123456789712345678981234567899123456789
```

12.7 TER format

Example of pro TER:

ATOM	72307	C4	U x	9	304.768	147.960	320.897	1.00218.84	C
ATOM	72308	O4	U x	9	304.171	146.902	321.104	1.00225.09	O
ATOM	72309	C5	U x	9	304.190	149.269	320.912	1.00211.91	C
ATOM	72310	C6	U x	9	304.960	150.336	320.668	1.00205.76	C
TER	72311		U x	9					

12.8 Add missing atoms

Add missing atoms etc.:

```
(py37) [mx] cwc46$ pdbfixer prp46.pdb --add-atoms all --add-residues
```

Read more:

- <https://github.com/openmm/pdbfixer>
- <http://htmlpreview.github.io/?https://raw.githubusercontent.com/pandegroup/pdbfixer/master/Manual.html>

12.9 Test for Cuda

Run a test for CUDA:

```
rna_test_cuda.py
> rna_test_cuda.py:9 in <module>- torch.cuda.current_device(): 0
> rna_test_cuda.py:10 in <module>
  torch.cuda.device(0): <torch.cuda.device object at 0x146bf9bf7b80>
> rna_test_cuda.py:11 in <module>- torch.cuda.is_available(): True
Using device: cuda

NVIDIA A40
Memory Usage:
Allocated: 0.0 GB
Reserved: 0.0 GB

tensor([[ 0.9374,  1.1526, -0.5648,  0.9870]], device='cuda:0')
```


EXAMPLES OF WORKFLOWS

13.1 Example #1

The native:

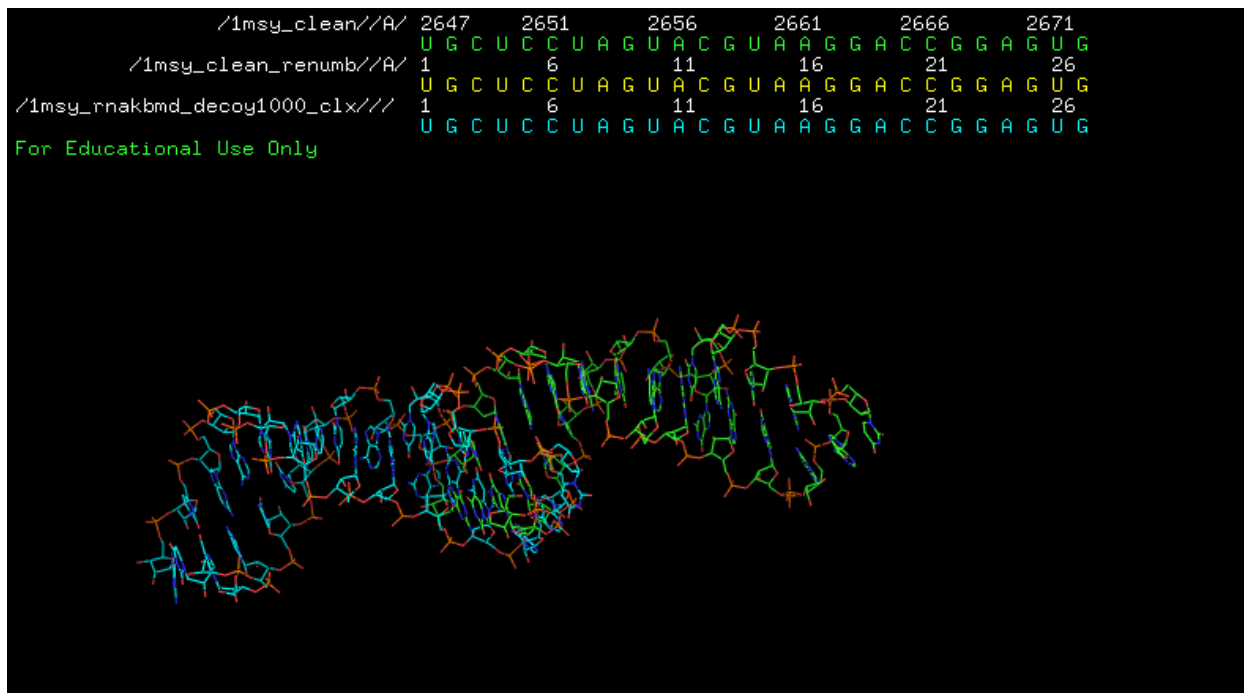
```
[mq] md_lmsy_clx$ cat lmsy_clean.pdb.outCR
Classifier: Clarna
chains: A 2647 2673
A 2648 A 2672 bp G U WW_cis 0.8732
A 2649 A 2671 bp C G WW_cis 0.9160
A 2650 A 2670 bp U A WW_cis 0.9289
A 2651 A 2669 bp C G WW_cis 0.9439
A 2652 A 2668 bp C G WW_cis 0.9281
A 2655 A 2656 bp G U SH_cis 0.9227
A 2656 A 2665 bp U A WH_tran 0.8526
A 2657 A 2664 bp A G HS_tran 0.8513
A 2658 A 2663 bp C G WW_cis 0.9421
A 2659 A 2662 bp G A SH_tran 0.7619
```

but analyzed structures are like:

```
[mq] md_lmsy_clx$ cat struc/lmsy_rnakbmd_decoy1478_clx.pdb.outCR
Classifier: Clarna
chains: A 1 27
2 26 bp G U WW_cis 0.7196
3 25 bp C G WW_cis 0.6702
4 24 bp U A WW_cis 0.8911
5 23 bp C G WW_cis 0.8925
6 22 bp C G WW_cis 0.9026
9 10 bp G U SH_cis 0.8714
10 19 bp U A WH_tran 0.7279
11 18 bp A G HS_tran 0.8810
12 17 bp C G WW_cis 0.9115
```

You have to renumber lmsy_clean.pdb to 1:27:

```
$ rna_pdb_tools.py --edit 'A:2647-2673>A:1:17' lmsy_clean.pdb > lmsy_clean_renumb.pdb
```



13.2 Example #2

Listing:

```

$ rna_pdb_tools.py --get-seq 1nuj_rnakbmd_decoy1000_clx.pdb
> 1nuj_rnakbmd_decoy1000_clx.pdb A:1-13
CGGACCGAGCCAG
> 1nuj_rnakbmd_decoy1000_clx.pdb B:14-24
GCUGGGAGUCC

$ rna_pdb_tools.py --get-seq 1nuj_clean.pdb
> 1nuj_clean.pdb A:18-30
CGGACCGAGCCAG
> 1nuj_clean.pdb B:39-49
GCUGGGAGUCC

$ rna_pdb_tools.py --edit 'A:18-30>A:1-13,B:39-49>B:14-24' 1nuj_clean.pdb > 1nuj_clean_
↪renumber.pdb

$ rna_pdb_tools.py --get-seq 1nuj_clean_renumber.pdb
> 1nuj_clean_renumber.pdb A:1-13
CGGACCGAGCCAG
> 1nuj_clean_renumber.pdb B:14-24
GCUGGGAGUCC

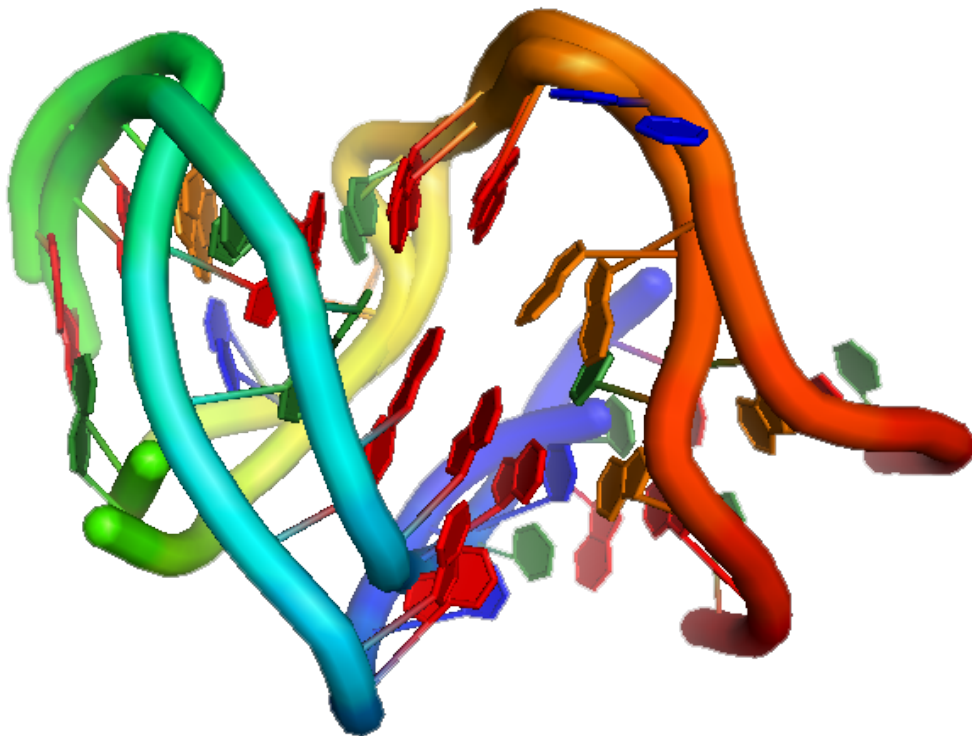
```

13.3 Example #3

Starting structure doesn't have chain id:

```
# add chain A
$ parallel "rna_add_chain.py -c A {} > ../struc_with_chain/{}" ::: *.pdb
# edit the second part of the new chain A as B
$ parallel "rna_pdb_gtools.py --edit 'A:14-27>B:14-27' {} > out/{}" ::: *.pdb
```

```
/1duq_rnakbnm_decoy0473_amb_c1x//A/ 1      6      11  14  16      21      26
G C U G G G C G C A G G C C U G A C G G U A C A G C
/1duq_rnakbnm_decoy0008_amb_c1x//A/ 1      6      11  /B/14  16      21      26
G C U G G G C G C A G G      C C U G A C G G U A C A G C
```



13.4 Example #4 Calculate RMSDs of unstandardized structures (RNA Puzzle #1)

You try to calculate RMSDs for RNA Puzzles #1:

```
$ rna_calc_rmsd.py -t 1_solution_0_rpr.pdb *.pdb
method: all-atom-built-in
# of models: 15
1_bujnicki_1_rpr.pdb 5.71 978
1_bujnicki_2_rpr.pdb 6.16 978
1_bujnicki_3_rpr.pdb 5.3 978
1_bujnicki_4_rpr.pdb 4.95 978
1_bujnicki_5_rpr.pdb 5.1 978
```

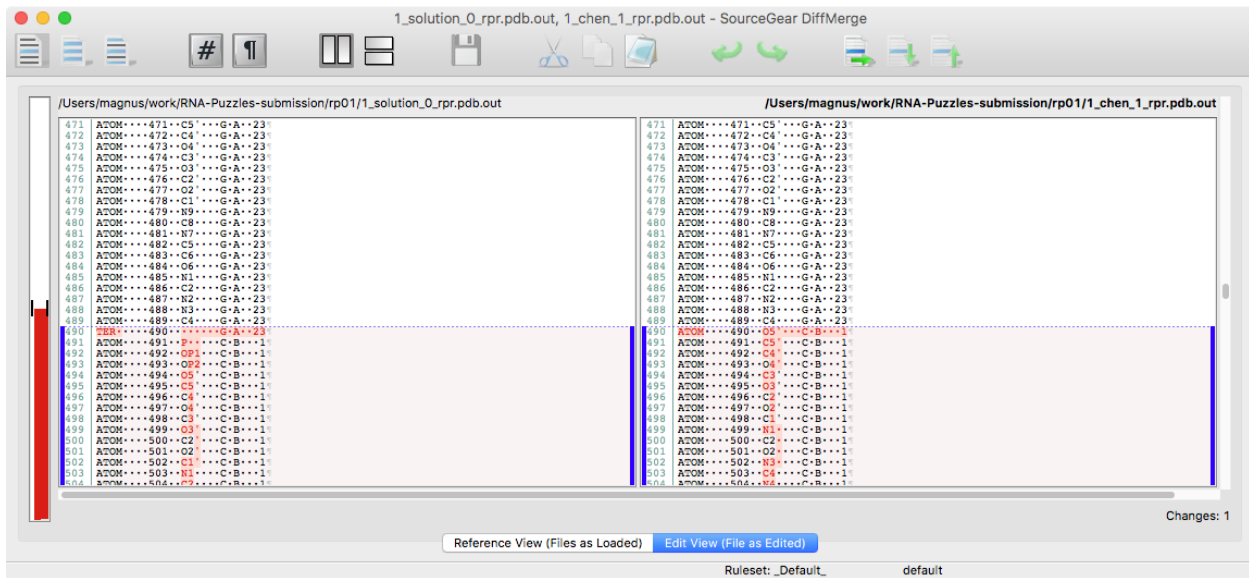
(continues on next page)

(continued from previous page)

```
Error: # of atoms is not equal target (1_solution_0_rpr.pdb):978 vs model (1_chen_1_rpr.
↳pdb):975
```

you can see that there is a different number of atoms in *1_solution_0_rpr.pdb* and *1_chen_1_rpr.pdb*.

To see more you can run *diffpdb*.



you see that something is wrong. To fix it, run:

```
$ rna_pdb_tools.py --rpr --inplace *.pdb
93% (15 of 16) |#####
↳##### | Elapsed Time: 0:00:03 ETA:↳
↳0:00:00
```

you can tail the files:

```
$ tail *.pdb
==> 1_bujnicki_1_rpr.pdb <==
ATOM 971 N7 G B 23 -16.558 -3.375 78.345 1.00 0.00 N
ATOM 972 C5 G B 23 -17.169 -2.575 77.384 1.00 0.00 C
ATOM 973 C6 G B 23 -17.589 -2.874 76.053 1.00 0.00 C
ATOM 974 O6 G B 23 -17.497 -3.930 75.430 1.00 0.00 O
ATOM 975 N1 G B 23 -18.234 -1.800 75.459 1.00 0.00 N
ATOM 976 C2 G B 23 -18.441 -0.576 76.049 1.00 0.00 C
ATOM 977 N2 G B 23 -19.127 0.345 75.382 1.00 0.00 N
ATOM 978 N3 G B 23 -18.053 -0.282 77.292 1.00 0.00 N
ATOM 979 C4 G B 23 -17.419 -1.324 77.898 1.00 0.00 C
...
==> 1_chen_1_rpr.pdb <==
ATOM 971 N7 G B 23 -14.462 -1.101 79.998 1.00 0.00 N
ATOM 972 C5 G B 23 -14.952 -0.485 78.839 1.00 0.00 C
ATOM 973 C6 G B 23 -15.577 -1.020 77.655 1.00 0.00 C
ATOM 974 O6 G B 23 -15.822 -2.189 77.351 1.00 0.00 O
```

(continues on next page)

(continued from previous page)

```

ATOM   975  N1   G B  23   -15.972  -0.051  76.763  1.00  0.00      N
ATOM   976  C2   G B  23   -15.787   1.274  76.944  1.00  0.00      C
ATOM   977  N2   G B  23   -16.269   2.059  76.021  1.00  0.00      N
ATOM   978  N3   G B  23   -15.224   1.822  78.022  1.00  0.00      N
ATOM   979  C4   G B  23   -14.818   0.884  78.935  1.00  0.00      C
TER    980           G B  23

==> 1_solution_0_rpr.pdb <==
ATOM   971  N7   G B  23    22.256  -1.292  27.403  1.00 34.10      N
ATOM   972  C5   G B  23    22.625  -0.176  28.135  1.00 31.12      C
ATOM   973  C6   G B  23    23.470  -0.096  29.260  1.00 28.80      C
ATOM   974  O6   G B  23    24.062  -1.036  29.804  1.00 28.26      O
ATOM   975  N1   G B  23    23.616   1.224  29.705  1.00 27.28      N
ATOM   976  C2   G B  23    22.971   2.318  29.112  1.00 28.31      C
ATOM   977  N2   G B  23    23.179   3.538  29.655  1.00 27.03      N
ATOM   978  N3   G B  23    22.170   2.245  28.047  1.00 28.85      N
ATOM   979  C4   G B  23    22.041   0.961  27.632  1.00 28.58      C
TER    980           G B  23%
```

so now you can see that the files look the same. Let's try to calculate RMSDs again:

```

$ rna_calc_rmsd.py -t 1_solution_0_rpr.pdb *.pdb
method: all-atom-built-in
# of models: 16
1_bujnicki_1_rpr.pdb 5.71 978
1_bujnicki_2_rpr.pdb 6.16 978
1_bujnicki_3_rpr.pdb 5.3 978
1_bujnicki_4_rpr.pdb 4.95 978
1_bujnicki_5_rpr.pdb 5.1 978
1_chen_1_rpr.pdb 4.35 978
1_chen_1_rpr_v2.pdb 4.35 978
1_das_1_rpr.pdb 3.97 978
1_das_2_rpr.pdb 4.48 978
1_das_3_rpr.pdb 3.43 978
1_das_4_rpr.pdb 3.92 978
1_das_5_rpr.pdb 4.57 978
1_dokholyan_1_rpr.pdb 7.25 978
1_major_1_rpr.pdb 4.34 978
1_santalucia_1_rpr.pdb 5.76 978
1_solution_0_rpr.pdb 0.0 978
# of atoms used: 978
csv was created! rmsds.csv
```

worked! :-)

This is a real-life case, <https://github.com/mmagnus/RNA-Puzzles-Normalized-submissions/tree/master/rp01>.

INSTALL & CONFIGURE

14.1 Pip as a developer

You can get rna-tools and install them from the current directory with this pip:

```
pip install -e git+http://github.com/mmagnus/rna-tools.git#egg=rna-tools
```

This way is better (than `pip install rna-tools`) if you're going to do some coding in the tools.

You can even first get rna-tools with git:

```
git clone http://github.com/mmagnus/rna-tools.git
```

and then do:

```
$ pip install -e .  
Obtaining file:///Users/magnus/work/src/rna-tools
```

to “install” with pip rna-tools based on a set of links to the current directory.

14.2 Test if installed

Test if the package is installed:

```
$ python -c 'import rna_tools'  
# ipython -c 'import rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd' (to test specific  
↪tools)
```

if nothing appears that's good. If you see this:

```
$ python -c 'import rna_tools'  
Traceback (most recent call last):  
  File "<string>", line 1, in <module>  
ImportError: No module named rna_tools
```

then go to your config file for the shell (`~/.zshrc` or `~/.bashrc`) add add:

```
PYTHONPATH=<path to rna-tools>/:$PYTHONPATH  
export PYTHONPATH
```

so in my case this is:

```
PYTHONPATH=<path to rna-tools>/home/magnus/src/rna-tools/:$PYTHONPATH
export PYTHONPATH
```

14.3 Configuration

To set up your own configuration, create `~/.rna_tools.py` in your HOME directory and redefine variables, e.g.:

```
(py37) [mx] rna-tools$ git:(master) cat ~/.rna_tools.py
VARNA_PATH = '/Users/magnus/work/opt/varna'
VARNA_JAR_NAME = 'VARNA.jar'
SIMRNA_DATA_PATH = '/Users/magnus/work/opt/simRNA/SimRNA_64bitIntel_MacOSX_staticLibs/
↳data'
QRNAS_PATH = "/Users/magnus/work/opt/qrnas/"
RCHIE_PATH = "/Users/magnus/work/opt/r-chie/"
RFAM_DB_PATH = "/Users/magnus/work/db/rfam/Rfam.cm"
CONTEXTFOLD_PATH = "/Users/magnus/work/opt/ContextFold_1_00/"
DIFF_TOOL = "open -a diffmerge"
CPUS_CLUSTER = 630
RNASTRUCTURE_PATH = "/Users/magnus/work/opt/RNAstructure/6.1/"
ENTRNA_PATH = "/Users/magnus/work/opt/ENTRNA"
```

14.4 All requirements

To get ALL requirements, use pip:

```
pip install -r docs/requirements.txt
```

By default rna-tools will not install all requirements, because some of them are heavy or might cause various problems, so you will be asked to install them when needed. For example, installing *matplotlib* is not essential for many other tools, *python-Levenshtein* is only used in one function.

14.5 Test all

This is still under active development.

To test (almost) all rna-tools functionality, you can run `rna_tools_test_all.py` to see if you got any errors, this should look like:

```
(py37) [mm] rna-tools$ git:(master) rna_tools_test_all.py
BlastPDB requires urllib3
- Python: 3.7.4 (default, Aug 13 2019, 15:17:50) [Clang 4.0.1 (tags/RELEASE_401/final)]
- rna-tools: b'py2-78-g3b3dd5f'
- RNA_TOOLS_PATH set to /home/magnus/work-src/rna-tools/
- See full list of tools <https://github.com/mmagnus/rna-tools/blob/master/rna-tools-
↳index.csv>
Seems OK
```

or for Python 2:

```
(base) [mm] rna-tools$ git:(master) rna_tools_test_all.py
- Python: 2.7.16 |Anaconda, Inc.| (default, Mar 14 2019, 16:24:02) [GCC 4.2.1 Compatible,
↳ Clang 4.0.1 (tags/RELEASE_401/final)]
- rna-tools: py2-78-g3b3dd5f
- RNA_TOOLS_PATH set to /home/magnus/work-src/rna-tools/
- See full list of tools <https://github.com/mmagnus/rna-tools/blob/master/rna-tools-
↳ index.csv
Seems OK
```

For crude testing you can also use `./test.sh` script and then see for errors in output and also check `output/` folder to see if there are differences between your output and output committed to GitHub by me.

14.6 PyMOL4RNA: adv config

For some extra functions you might also follow this. Open your `~/.pymolrc` and set up following variables as you need:

```
# rna-tools
EXECUTABLE="/bin/zsh" # set up your shell, usually /bin/bash or /bin/zsh
SOURCE="source ~/.zshrc" # set up the path to the file where you keep your shell,
↳ variables

CLARNA_RUN="/Users/magnus/work-src/clarna_play/clarna_run.py" # if you want to run,
↳ clarna_run.py set up the path

run ~/work-src/rna-tools/rna_tools/tools/pymol_drawing/pymol_drawing.py
run ~/work-src/rna-tools/rna_tools/tools/rna_filter/pymol_dists.py
```

The plugins have been tested with MacPyMOL version 1.7.4.5 Edu.

GIT QUICKREF



For Git in a scientific environment, please read:

Perez-Riverol, Y. et al., 2016. Ten Simple Rules for Taking Advantage of Git and GitHub S. Markel, ed. PLoS computational biology, 12(7), p.e1004947.

Blischak, J.D., Davenport, E.R. & Wilson, G., 2016. A Quick Introduction to Version Control with Git and GitHub F. Ouellette, ed. PLoS computational biology, 12(1), pp.e1004668–18.

Git is a version control system that is used for software development that helps you to keep track of versions of your program. To start using git you have to know only these two commands below. If you want to contribute to the package you need a few more, but it's not important right now :-)

To get the package for the first time on your computer go to the selected package and on the top left corner select clone or download. Copy the link and run it on console with git clone:

```
$ git clone git@github.com:mmagnus/rna-pdb-tools.git
```

and if you want to update the package later run:

```
$ git pull # be in the folder like ~/src/rna-pdb-tools/ <here>
```

if you see something like this:

```
$ git pull
Already up-to-date.
```

it means that your version of the package is up to date, congrats! :-)

If you see something like this:

```
$ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:mmagnus/rna-pdb-tools
```

(continues on next page)

(continued from previous page)

```
69c4ee3..7f90739 master -> origin/master
Updating 69c4ee3..7f90739
Fast-forward
install_links_bin.sh | 1 +
1 file changed, 1 insertion(+)
```

it means that there is a small change in `install_links_bin.sh` and you are up to date, congrats as well! You might need to run `./install_links_bin.sh` to “install” new tools that were added to the packages (if this is the case). If you get any error then talk to me magnus@genesilico.pl.

15.1 Git sheet cheat

```
$ git init # start git repo in a folder
$ git add <file> # add file to stage area
$ git commit -m <text> # send file to
$ git push # sent this remote
$ git gui # install `apt-get install git-gui`
$ git log #
$ git status # get status of your repo

$ git clone git@gitlab.genesilico.pl:magnus/git_crash_course.git

$ git remote add origin git@gitlab.genesilico.pl:magnus/git_crash_course.git

$ git push -u origin master
Counting objects: 45, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (41/41), done.
Writing objects: 100% (45/45), 4.97 KiB | 0 bytes/s, done.
Total 45 (delta 12), reused 0 (delta 0)
To git@gitlab.genesilico.pl:magnus/git_crash_course.git
* [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

15.2 Learn Git

Model used by Git vs Svn:

```
file <-> stage area <-> local repo <-> git repo/gitlab
file <-----> SVM repo
```

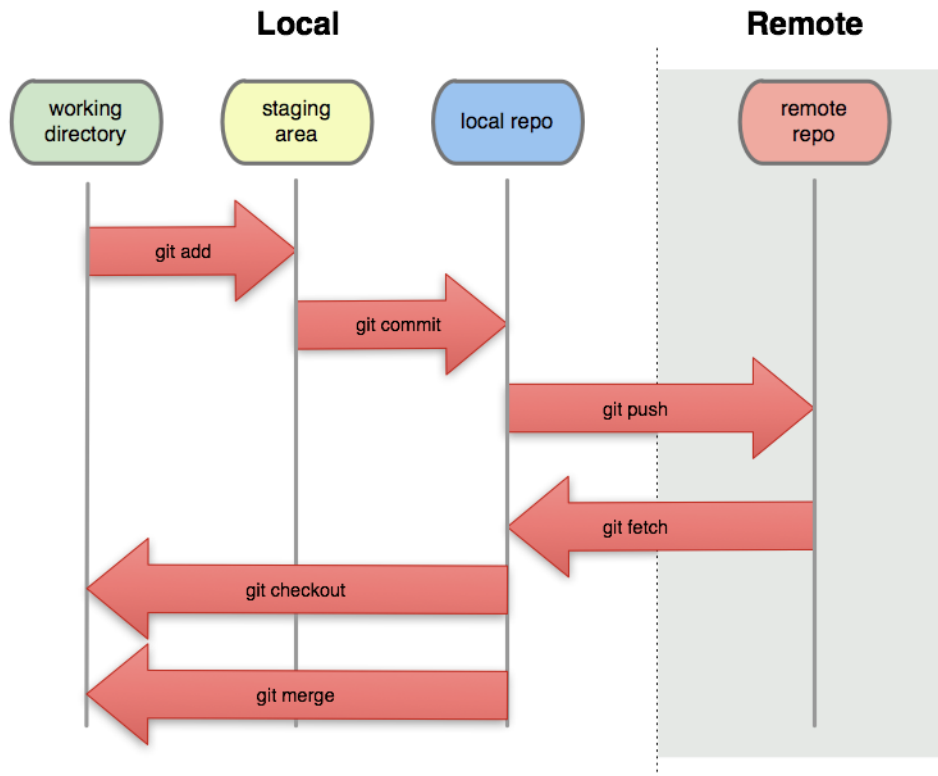


Figure. from: <https://greenido.files.wordpress.com/2013/07/git-local-remote.png?w=696&h=570>

For more see, [git - the simple guide](#) (just a simple guide for getting started with git. no deep shit ;))

<http://rogerdudler.github.io/git-guide/>

A git cheat sheet http://rogerdudler.github.io/git-guide/files/git_cheat_sheet.pdf .

If you want to read more, we highly recommend a book our collages, Kristian (<http://www.apress.com/us/book/9781484222409>).

To understand the principles of Git, a bit more advance, by Linus Torvals (an author of Git) <https://www.youtube.com/watch?v=4XpnKHJAok8>

How to learn Git in 20min https://www.youtube.com/watch?v=Y9XZQO1n_7c

CONFIGURATION

Keep configuration syntax like:

```
from rna_tools.rna_tools_config import CPUS_CLUSTER
# since we use export PYTHONPATH=$PYTHONPATH:/home/magnus/src/rna-tools/
```

vs:

```
try:
    RNA_ROSETTA_RUN_ROOT_DIR_MODELING = os.environ['RNA_ROSETTA_RUN_ROOT_DIR_MODELING']
except:
    print ('Set up RNA_ROSETTA_RUN_ROOT_DIR_MODELING in .bashrc')
```


DOCUMENTATION

We are using (at least we are moving towards) the Google style docstrings via Napoleon. Napoleon is a Sphinx Extension that enables Sphinx to parse both NumPy and Google style docstrings - the style recommended by Khan Academy. <http://www.sphinx-doc.org/en/stable/ext/napoleon.html#type-annotations>

17.1 Generate docs

At the moment the generation of the docs is quite tricky, because of old dependencies on Python2 and old version of Sphinx. To generate docs locally please switch to Python2 and then run `cd docs/; make clean; make html`.

It should be updated at some point.

ADD A NEW TOOL TO THE PACKAGE

1. Create a new folder in `rna-tools/rna_tools/tools` with your tool. The folder will be seen online after your push at https://github.com/mmagnus/rna-tools/tree/master/rna_tools/tools. We will walk you through this simple example https://github.com/mmagnus/rna-tools/tree/master/rna_tools/tools/renum_pdb_to_aln.
2. Make sure that there is a simple test as `test.sh`:

```
#!/bin/bash
python renum_pdb_to_aln.py --residue_index_start 1 obj1 test_data/ALN_OBJ1_OBJ2.fasta
↪ test_data/obj01.pdb
```

and there is a `test_data` folder with some test inputs and outputs. See the example.

3. Add your tool to `install_links_bin.sh` at the top folder of `rna-tools`:

```
ln -s $curr_dir/rna_tools /tools/<tool folder>/<util script name with .py> $curr_
↪ dir/bin/<util script name with .py>

e.g.

ln -s $curr_dir/rnatools/utils/renum_pdb_to_aln/renum_pdb_to_aln.py $curr_dir/bin/
↪ rna_renum_pdb_to_aln.py
```

This will “install” you script in bin directory of the project so it can be used system-wide.

Run this script to see if there is any error, `./install_links_bin.sh`.

4. Add your tool to the documentation. The tool has to be “importable”, so don’t forget to create `__init__.py` inside your tool directory. Next, go to `rna-tools/docs/source` and edit `tools.rst`. Add, wherever you think your tool will fit, lines like:

```
Renumber a pdb file according to alignment
~~~~~
.. autoprogram:: rna_tools.tools.<tool folder>.<tool script name>.get_parser()
   :prog: <util script name>

.. automodule:: rna_tools.tools.<tool folder>.<tool script name>
   : members:

e.g.:

Renumber a pdb file according to alignment
~~~~~
.. autoprogram:: rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln.get_parser()
```

(continues on next page)

(continued from previous page)

```
:prog: renum_pdb_to_aln
.. automodule:: rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln
:members:
```

and run `make html` in the folder to check if the documentation is compiled without any errors.

If you are using any external library such as `scipy`, please make sure that they are listed in `rna-tools/docs/requirements.txt`. If the library is not there, please add it. This file is read by the Read The Docs to compile the documentation online and also by Travis for continuous testing.

You can open the documentation compiled locally under a link `file://<path to rna-tools>/rna-tools/docs/build/html/index.html`, e.g. `file:///Users/magnus/work/src/rna-tools/docs/build/html/index.html`.

5. The very last step is to add your tool `test.sh` to the main testing script. Edit `rna-tools/test.sh` and add

```
cd ./tools/<tool folder>/
./test.sh
cd ../../

e.g.

cd ./tools/renum_pdb_to_aln/
./test.sh
cd ../../
```

6. Run this main test (`./test.sh`) and see if the tool works as expected.
7. Now we are ready to push the changes. In the terminal, type:

```
$ git pull
$ git add <files> # or use git gui
$ git commit -m <desc the tool>
$ git push
```

to commit all your changes and push it to the Github repository!

Warning

This testing is very, very rough and we are moving to have more test in `py.test` at some point.

INDICES AND TABLES

- [genindex](#)
- [modindex](#)
- [search](#)

PYTHON MODULE INDEX

r

`rna_tools.BlastPDB`, 59
`rna_tools.RfamSearch`, 60
`rna_tools.rna_tools_lib`, 27
`rna_tools.SecondaryStructure`, 56
`rna_tools.Seq`, 51
`rna_tools.tools.clanstix.rna_clanstix`, 141
`rna_tools.tools.clarna_app.rna_clarna_app`, 108
`rna_tools.tools.cluster_load.cluster_load`, 148
`rna_tools.tools.diffpdb.diffpdb`, 138
`rna_tools.tools.extra_functions.select_fragment`, 171
`rna_tools.tools.misc.rna_add_chain`, 47
`rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry`, 117
`rna_tools.tools.mq.ClashScore.ClashScore`, 116
`rna_tools.tools.mq.Dfire.Dfire`, 110
`rna_tools.tools.mq.eSCORE.eSCORE`, 112
`rna_tools.tools.mq.FARNA.FARNA`, 115
`rna_tools.tools.mq.RASP.RASP`, 109
`rna_tools.tools.mq.RNA3DCNN.RNA3DCNN`, 113
`rna_tools.tools.mq.RNAkb.RNAkb`, 111
`rna_tools.tools.mq.RNAscore.RNAscore`, 113
`rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists`, 48
`rna_tools.tools.pymol_drawing.pymol_drawing`, 155
`rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln`, 88
`rna_tools.tools.rna_alignment.random_assignment_of_nucleotides`, 83
`rna_tools.tools.rna_alignment.rna_alignment`, 61
`rna_tools.tools.rna_calc_inf.rna_calc_inf`, 100
`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd`, 92
`rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all`, 98
`rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_pk_to_simrna`, 59
`rna_tools.tools.rna_filter.pymol_dists`, 103
`rna_tools.tools.rna_rosetta.rna_rosetta_check_progress`, 130
`rna_tools.tools.rna_rosetta.rna_rosetta_cluster`, 127
`rna_tools.tools.rna_rosetta.rna_rosetta_extract_lowscore_c`, 129
`rna_tools.tools.rna_rosetta.rna_rosetta_head`, 125
`rna_tools.tools.rna_rosetta.rna_rosetta_min`, 128
`rna_tools.tools.rna_rosetta.rna_rosetta_n`, 123
`rna_tools.tools.rna_rosetta.rna_rosetta_run`, 120
`rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket`, 146
`rna_tools.tools.rna_x3dna.rna_x3dna`, 106
`rna_tools.tools.simrna_trajectory.simrna_trajectory`, 133

Symbols

- 0
 - rna_rosetta_n.py command line option, 123
- alignfn
 - random_assignment_of_nucleotides.py command line option, 83
- all-stars
 - rna_align_foldability.py command line option, 82
- backbone-only
 - rna_pdb_tools.py command line option, 21
 - rna_standardize.py command line option, 38
- bases-only
 - rna_pdb_tools.py command line option, 21
 - rna_standardize.py command line option, 39
- bfactor
 - rna_pdb_edit_occupancy_bfactor.py command line option, 45
- bins
 - rna_plot_hist.py command line option, 145
- chain
 - rna_add_chain.py command line option, 47
 - rna_ec2x.py command line option, 104
- chain-first
 - rna_pdb_tools.py command line option, 20
- check-geometry
 - rna_standardize.py command line option, 38
- cif
 - rna_standardize.py command line option, 38
- cif2pdb
 - rna_pdb_tools.py command line option, 20
- clean
 - rna_pdb_tools.py command line option, 18
- cleanup
 - rna_simrna_extract.py command line option, 131
- cluster
 - rna_simrnaweb_download_job.py command line option, 133
- collapsed-view
 - rna_pdb_tools.py command line option, 19
- color-seq
 - rna_pdb_tools.py command line option, 19
- column
 - rna_plot_density.py command line option, 146
 - rna_plot_hist.py command line option, 145
- compact
 - rna_pdb_tools.py command line option, 19
- conect
 - rna_pdb_tools.py command line option, 17
 - rna_standardize.py command line option, 38
- conect-no-linkage
 - rna_pdb_tools.py command line option, 17
 - rna_standardize.py command line option, 38
- cpus
 - rna_rosetta_min.py command line option, 128
 - rna_rosetta_run.py command line option, 120
- csv
 - rna_alignment_calc_energy.py command line option, 75
- cv
 - rna_pdb_tools.py command line option, 19
- dca
 - rna_dca_mapping.py command line option, 102
- debug
 - rna_alignment_calc_energy.py command line option, 75
 - rna_alignment_get_species.py command line option, 73
 - rna_calc_inf.py command line option, 100
- delete
 - rna_pdb_tools.py command line option, 20
- delete-anisou
 - rna_pdb_tools.py command line option, 17

```

--dev
  rna_align_foldability.py command line
    option, 82
--dist
  rna_pairs2SimRNArestrs.py command line
    option, 105
--dont-fix-missing-atoms
  rna_pdb_tools.py command line option, 19
  rna_standardize.py command line option,
    38
--dont-remove-sel-files
  rna_calc_inf.py command line option, 100
--dont-rename-chains
  rna_pdb_tools.py command line option, 19
  rna_standardize.py command line option,
    38
--dont-replace-hetatm
  rna_standardize.py command line option,
    38
--dont-report-missing-atoms
  rna_pdb_tools.py command line option, 21
  rna_standardize.py command line option,
    38
--download-trajectory
  rna_simrnaweb_download_job.py command
    line option, 133
--ec-pairs
  rna_ec2x.py command line option, 104
--edit
  rna_pdb_tools.py command line option, 20
--evo-mapping
  rna_alignment_get_species.py command
    line option, 73
--evo-mapping-default
  rna_alignment_get_species.py command
    line option, 74
--extract
  rna_pdb_tools.py command line option, 20
  rna_simrnaweb_download_job.py command
    line option, 133
--extract-chain
  rna_pdb_tools.py command line option, 20
--fasta
  rna_pdb_tools.py command line option, 20
--fetch
  rna_pdb_tools.py command line option, 18
--fetch-ba
  rna_pdb_tools.py command line option, 18
--fetch-chain
  rna_pdb_tools.py command line option, 18
--fetch-cif
  rna_pdb_tools.py command line option, 18
--fetch-fasta
  rna_pdb_tools.py command line option, 18
--fetch-header
  rna_pdb_tools.py command line option, 18
--fix
  rna_pdb_tools.py command line option, 18
--flanks
  rna_alignment_calc_energy.py command
    line option, 76
--force
  rna_calc_inf.py command line option, 100
--get-chain
  rna_pdb_tools.py command line option, 18
--get-rnapuzzle-ready
  rna_pdb_tools.py command line option, 21
--get-seq
  rna_pdb_tools.py command line option, 19
--get-ss
  rna_pdb_tools.py command line option, 19
--go
  rna_rosetta_min.py command line option,
    128
  rna_rosetta_run.py command line option,
    120
--gseq
  rna_dca_mapping.py command line option,
    102
--helices
  rna_rosetta_run.py command line option,
    120
--help
  pdbs_measure_atom_dists.py command line
    option, 48
  random_assignment_of_nucleotides.py
    command line option, 83
  renum_to_aln.py command line option, 88
  rna_add_chain.py command line option, 47
  rna_align_distance_to_seq.py command
    line option, 77
  rna_align_foldability.py command line
    option, 82
  rna_align_get_ss_from_fasta.py command
    line option, 76
  rna_align_get_ss_from_stk.py command
    line option, 76
  rna_alignment_calc_energy.py command
    line option, 75
  rna_alignment_get_species.py command
    line option, 73
  rna_cal_rmsd_trafl_plot command line
    option, 97
  rna_calc_evo_rmsd command line option, 97
  rna_calc_inf.py command line option, 99
  rna_calc_rmsd command line option, 91
  rna_calc_rmsd_all_vs_all command line
    option, 98

```

rna_calc_rmsd_multi_targets.py command line option, 95
 rna_dca_mapping.py command line option, 102
 rna_dot2ct.py command line option, 58
 rna_ec2x.py command line option, 104
 rna_filter.py command line option, 101
 rna_mq_rnakb.py command line option, 112
 rna_pairs2SimRNArests.py command line option, 104
 rna_pairs_diff.py command line option, 106
 rna_pdb_edit_occupancy_bfactor.py command line option, 45
 rna_pdb_tools.py command line option, 17
 rna_plot_density.py command line option, 146
 rna_plot_hist.py command line option, 145
 rna_refinement.py command line option, 138
 rna_rosetta_cluster.py command line option, 126, 130
 rna_rosetta_extract_lowscore_decoys.py command line option, 129
 rna_rosetta_min.py command line option, 128
 rna_rosetta_n.py command line option, 123, 125
 rna_rosetta_run.py command line option, 120
 rna_sali2dotbracket command line option, 146
 rna_simrna_extract.py command line option, 131
 rna_simrna_lowest.py command line option, 131
 rna_simrnaweb_download_job.py command line option, 132
 rna_ss_get_bps.py command line option, 105
 rna_standardize.py command line option, 37
 rna_torsions.py command line option, 44
 --here
 rna_pdb_tools.py command line option, 21
 rna_standardize.py command line option, 38
 --hide-warnings
 rna_pdb_tools.py command line option, 19
 --id-width
 rna_alignment_get_species.py command line option, 73
 --ignore-files
 rna_calc_inf.py command line option, 99
 rna_calc_rmsd command line option, 91
 rna_pdb_tools.py command line option, 19
 --init
 rna_rosetta_run.py command line option, 120
 --inplace
 rna_pdb_tools.py command line option, 21
 rna_standardize.py command line option, 38
 --input-dir
 rna_calc_rmsd_all_vs_all command line option, 98
 --inspect
 rna_pdb_tools.py command line option, 19
 --interactive
 rna_refinement.py command line option, 138
 --is-nmr
 rna_pdb_tools.py command line option, 18
 --is-pdb
 rna_pdb_tools.py command line option, 18
 --keep-hetatm
 rna_pdb_tools.py command line option, 21
 rna_standardize.py command line option, 38
 --kill
 rna_rosetta_cluster.py command line option, 130
 --limit-clusters
 rna_rosetta_cluster.py command line option, 127
 --loop-seq
 rna_alignment_calc_energy.py command line option, 75
 --matrix-fn
 rna_calc_rmsd_all_vs_all command line option, 98
 --mdr
 rna_pdb_tools.py command line option, 20
 rna_standardize.py command line option, 38
 --method
 rna_alignment_calc_energy.py command line option, 75
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92
 rna_calc_rmsd_all_vs_all command line option, 98
 --min-only
 rna_rosetta_cluster.py command line option, 130
 --model-ignore-selection
 rna_calc_rmsd command line option, 91
 --model-selection

```

    rna_calc_inf.py command line option, 100
    rna_calc_rmsd command line option, 91
    rna_calc_rmsd_multi_targets.py command
        line option, 95
--models
    rna_calc_rmsd_multi_targets.py command
        line option, 95
--more_clusters
    rna_simrna_web_download_job.py command
        line option, 133
--motif
    rna_rosetta_run.py command line option,
        120
--mss
    rna_dca_mapping.py command line option,
        102
--mutate
    rna_pdb_tools.py command line option, 19
--name-rmsd-column
    rna_calc_rmsd command line option, 92
--nmr-dir
    rna_pdb_tools.py command line option, 18
--no_select
    rna_rosetta_cluster.py command line
        option, 126
--no-backbone
    rna_pdb_tools.py command line option, 21
    rna_standardize.py command line option,
        39
--no-hr
    rna_pdb_tools.py command line option, 19
    rna_standardize.py command line option,
        38
--no-progress-bar
    rna_pdb_tools.py command line option, 17
    rna_standardize.py command line option,
        37
--no-stacking
    rna_calc_inf.py command line option, 100
--noshort
    rna_dca_mapping.py command line option,
        102
--noss
    rna_dca_mapping.py command line option,
        102
--nstruc
    rna_rosetta_n.py command line option, 125
    rna_rosetta_run.py command line option,
        120
    rna_simrna_lowest.py command line
        option, 131
    rna_simrna_web_download_job.py command
        line option, 132
--number_of_structures
    rna_simrna_extract.py command line
        option, 131
--number-of-threads
    rna_calc_inf.py command line option, 99
--occupancy
    rna_pdb_edit_occupancy_bfactor.py
        command line option, 45
--offset
    rna_dca_mapping.py command line option,
        102
    rna_filter.py command line option, 102
    rna_pairs2SimRNArests.py command line
        option, 104
    rna_ss_get_bps.py command line option,
        105
--one
    rna_alignment_calc_energy.py command
        line option, 75
    rna_alignment_get_species.py command
        line option, 74
--oneline
    rna_pdb_tools.py command line option, 20
--orgmode
    rna_pdb_tools.py command line option, 18
--osfn
    rna_alignment_get_species.py command
        line option, 74
--out_fn
    rna_calc_inf.py command line option, 100
--outfn
    random_assignment_of_nucleotides.py
        command line option, 83
    renum_to_aln.py command line option, 88
--output
    rna_pdb_edit_occupancy_bfactor.py
        command line option, 45
    rna_plot_density.py command line option,
        146
    rna_plot_hist.py command line option, 145
--output_file
    rna_refinement.py command line option,
        138
--output-csv
    rna_calc_rmsd_multi_targets.py command
        line option, 95
--p-only
    rna_pdb_tools.py command line option, 21
--pairs-delta
    rna_ec2x.py command line option, 104
--pdb2cif
    rna_pdb_tools.py command line option, 20
--potential
    rna_mq_rnakb.py command line option, 112
--prefix

```

rna_simrnaweb_download_job.py command line option, 132
 --print-progress
 rna_calc_rmsd command line option, 92
 --print-results
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92
 --quiet
 rna_standardize.py command line option, 38
 --radius-inc-step
 rna_rosetta_cluster.py command line option, 127
 --ref-frame-only
 rna_pdb_tools.py command line option, 21
 --remove0
 rna_pdb_tools.py command line option, 18
 --remove-trajectory
 rna_simrnaweb_download_job.py command line option, 133
 --rename-chain
 rna_pdb_tools.py command line option, 20
 --renum-atoms
 rna_pdb_tools.py command line option, 17
 --renum-nmr
 rna_pdb_tools.py command line option, 17
 rna_standardize.py command line option, 38
 --renum-residues-dirty
 rna_pdb_tools.py command line option, 17
 --renumber-residues
 rna_calc_inf.py command line option, 100
 rna_pdb_tools.py command line option, 19
 rna_standardize.py command line option, 38
 --replace-chain
 rna_pdb_tools.py command line option, 20
 --replace-hetatm
 rna_pdb_tools.py command line option, 21
 --replace-htm
 rna_pdb_tools.py command line option, 20
 --report
 rna_pdb_tools.py command line option, 17
 --residue_index_start
 renum_to_aln.py command line option, 88
 --restraints_fn
 rna_filter.py command line option, 102
 --rfam
 rna_alignment_get_species.py command line option, 74
 --rgyration
 rna_pdb_tools.py command line option, 19
 --rmsds-fn
 rna_calc_rmsd command line option, 92
 --rosetta
 rna_rosetta_run.py command line option, 120
 --rosetta2generic
 rna_pdb_tools.py command line option, 19
 --rpr
 rna_pdb_tools.py command line option, 21
 --sandbox
 rna_rosetta_run.py command line option, 120
 --save-single-res
 rna_pdb_tools.py command line option, 21
 --select
 rna_pdb_edit_occupancy_bfactor.py command line option, 45
 rna_rosetta_cluster.py command line option, 130
 --select-atoms
 rna_pdb_edit_occupancy_bfactor.py command line option, 46
 --sep
 rna_ec2x.py command line option, 104
 rna_plot_density.py command line option, 146
 rna_plot_hist.py command line option, 145
 --seq
 rna_dca_mapping.py command line option, 102
 --seqfn
 random_assignment_of_nucleotides.py command line option, 83
 --set-chain
 rna_pdb_tools.py command line option, 20
 --set-not-selected-to
 rna_pdb_edit_occupancy_bfactor.py command line option, 45
 --set-to
 rna_pdb_edit_occupancy_bfactor.py command line option, 45
 --skip-mcfold
 rna_align_foldability.py command line option, 82
 --sort-results
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92
 --split-alt-locations
 rna_pdb_tools.py command line option, 18
 --ss
 rna_calc_inf.py command line option, 100
 --ss-pairs
 rna_ec2x.py command line option, 104
 --steps
 rna_refinement.py command line option, 138

```

--suffix
  rna_pdb_tools.py command line option, 21
  rna_standardize.py command line option,
  38
--swap-chains
  rna_pdb_tools.py command line option, 20
--target_fn
  rna_calc_inf.py command line option, 99
--target-column-name
  rna_calc_rmsd command line option, 92
--target-fn
  rna_calc_rmsd command line option, 91
--target-ignore-selection
  rna_calc_rmsd command line option, 91
--target-selection
  rna_calc_inf.py command line option, 100
  rna_calc_rmsd command line option, 91
  rna_calc_rmsd_multi_targets.py command
  line option, 95
--targets
  rna_calc_rmsd_multi_targets.py command
  line option, 95
--template
  rna_alignment_calc_energy.py command
  line option, 75
  rna_simrna_extract.py command line
  option, 131
--to-mol2
  rna_pdb_tools.py command line option, 18
--top100
  rna_simrnaweb_download_job.py command
  line option, 133
--top200
  rna_simrnaweb_download_job.py command
  line option, 133
--trafl
  rna_simrna_extract.py command line
  option, 131
--un-nmr
  rna_pdb_tools.py command line option, 18
--undo
  rna_pdb_tools.py command line option, 17
--uniq
  rna_pdb_tools.py command line option, 20
--verbose
  pdbs_measure_atom_dists.py command line
  option, 48
  random_assignment_of_nucleotides.py
  command line option, 83
  renum_to_aln.py command line option, 88
  rna_align_foldability.py command line
  option, 82
  rna_alignment_calc_energy.py command
  line option, 76
  rna_alignment_get_species.py command
  line option, 73
  rna_calc_inf.py command line option, 100
  rna_calc_rmsd command line option, 92
  rna_calc_rmsd_multi_targets.py command
  line option, 95
  rna_dca_mapping.py command line option,
  102
  rna_dot2ct.py command line option, 58
  rna_filter.py command line option, 102
  rna_mq_rnakb.py command line option, 112
  rna_pairs2SimRNArestrs.py command line
  option, 105
  rna_pairs_diff.py command line option,
  106
  rna_pdb_edit_occupancy_bfactor.py
  command line option, 46
  rna_pdb_tools.py command line option, 19
  rna_refinement.py command line option,
  138
  rna_rosetta_cluster.py command line
  option, 130
  rna_rosetta_extract_lowscore_decoys.py
  command line option, 129
  rna_rosetta_n.py command line option,
  123, 125
  rna_ss_get_bps.py command line option,
  105
  rna_standardize.py command line option,
  38
  rna_torsions.py command line option, 45
--version
  rna_pdb_tools.py command line option, 17
  rna_standardize.py command line option,
  37
--way
  rna_calc_rmsd command line option, 92
--web
  rna_calc_inf.py command line option, 100
--web-models
  rna_simrnaweb_download_job.py command
  line option, 133
--weight
  rna_pairs2SimRNArestrs.py command line
  option, 104
--well
  rna_pairs2SimRNArestrs.py command line
  option, 105
--zero
  rna_rosetta_n.py command line option, 123
-c
  rna_add_chain.py command line option, 47
  rna_pdb_tools.py command line option, 18
  rna_rosetta_min.py command line option,

```

128
 rna_rosetta_run.py command line option,
 120
 rna_simrna_extract.py command line
 option, 131
 rna_simrnaweb_download_job.py command
 line option, 133
 -d
 rna_simrnaweb_download_job.py command
 line option, 133
 -e
 rna_rosetta_run.py command line option,
 120
 rna_simrnaweb_download_job.py command
 line option, 133
 -f
 rna_calc_inf.py command line option, 100
 rna_simrna_extract.py command line
 option, 131
 -g
 rna_rosetta_min.py command line option,
 128
 rna_rosetta_run.py command line option,
 120
 -h
 pdb measure_atom_dists.py command line
 option, 48
 random_assignment_of_nucleotides.py
 command line option, 83
 renum_to_aln.py command line option, 88
 rna_add_chain.py command line option, 47
 rna_align_distance_to_seq.py command
 line option, 77
 rna_align_foldability.py command line
 option, 82
 rna_align_get_ss_from_fasta.py command
 line option, 76
 rna_align_get_ss_from_stk.py command
 line option, 76
 rna_alignment_calc_energy.py command
 line option, 75
 rna_alignment_get_species.py command
 line option, 73
 rna_cal_rmsd_trafl_plot command line
 option, 97
 rna_calc_evo_rmsd command line option, 97
 rna_calc_inf.py command line option, 99
 rna_calc_rmsd command line option, 91
 rna_calc_rmsd_all_vs_all command line
 option, 98
 rna_calc_rmsd_multi_targets.py command
 line option, 95
 rna_dca_mapping.py command line option,
 102
 rna_dot2ct.py command line option, 58
 rna_ec2x.py command line option, 104
 rna_filter.py command line option, 101
 rna_mq_rnacb.py command line option, 112
 rna_pairs2SimRNArestrs.py command line
 option, 104
 rna_pairs_diff.py command line option,
 106
 rna_pdb_edit_occupancy_bfactor.py
 command line option, 45
 rna_pdb_tools.py command line option, 17
 rna_plot_density.py command line option,
 146
 rna_plot_hist.py command line option, 145
 rna_refinement.py command line option,
 138
 rna_rosetta_cluster.py command line
 option, 126, 130
 rna_rosetta_extract_lowscore_decoys.py
 command line option, 129
 rna_rosetta_min.py command line option,
 128
 rna_rosetta_n.py command line option,
 123, 125
 rna_rosetta_run.py command line option,
 120
 rna_sali2dotbracket command line option,
 146
 rna_simrna_extract.py command line
 option, 131
 rna_simrna_lowest.py command line
 option, 131
 rna_simrnaweb_download_job.py command
 line option, 132
 rna_ss_get_bps.py command line option,
 105
 rna_standardize.py command line option,
 37
 rna_torsions.py command line option, 44
 -i
 rna_calc_rmsd_all_vs_all command line
 option, 98
 rna_refinement.py command line option,
 138
 rna_rosetta_run.py command line option,
 120
 -k
 rna_rosetta_cluster.py command line
 option, 130
 -m
 rna_calc_inf.py command line option, 99
 rna_calc_rmsd command line option, 92
 rna_calc_rmsd_all_vs_all command line
 option, 98

rna_rosetta_cluster.py command line option, 130
 rna_rosetta_run.py command line option, 120
 rna_simrnaweb_download_job.py command line option, 133
 -n
 rna_rosetta_n.py command line option, 125
 rna_rosetta_run.py command line option, 120
 rna_simrna_extract.py command line option, 131
 rna_simrna_lowest.py command line option, 131
 rna_simrnaweb_download_job.py command line option, 132
 -o
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92
 rna_calc_rmsd_all_vs_all command line option, 98
 rna_pdb_edit_occupancy_bfactor.py command line option, 45
 rna_plot_density.py command line option, 146
 rna_plot_hist.py command line option, 145
 rna_refinement.py command line option, 138
 -p
 rna_mq_rnakb.py command line option, 112
 rna_simrnaweb_download_job.py command line option, 132
 -pp
 rna_calc_rmsd command line option, 92
 -pr
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92
 -r
 rna_filter.py command line option, 102
 rna_pdb_tools.py command line option, 17
 rna_rosetta_run.py command line option, 120
 rna_simrnaweb_download_job.py command line option, 133
 -s
 rna_calc_inf.py command line option, 100
 rna_filter.py command line option, 102
 rna_refinement.py command line option, 138
 rna_rosetta_cluster.py command line option, 130
 -sr
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92

-t
 rna_calc_inf.py command line option, 99
 rna_calc_rmsd command line option, 91
 rna_filter.py command line option, 102
 rna_simrna_extract.py command line option, 131
 -v
 pdbs_measure_atom_dists.py command line option, 48
 random_assignment_of_nucleotides.py command line option, 83
 renum_to_aln.py command line option, 88
 rna_align_foldability.py command line option, 82
 rna_alignment_calc_energy.py command line option, 76
 rna_alignment_get_species.py command line option, 73
 rna_calc_inf.py command line option, 100
 rna_calc_rmsd command line option, 92
 rna_calc_rmsd_multi_targets.py command line option, 95
 rna_dot2ct.py command line option, 58
 rna_filter.py command line option, 102
 rna_mq_rnakb.py command line option, 112
 rna_pairs2SimRNArestrs.py command line option, 105
 rna_pairs_diff.py command line option, 106
 rna_pdb_tools.py command line option, 19
 rna_refinement.py command line option, 138
 rna_rosetta_cluster.py command line option, 130
 rna_rosetta_extract_lowscore_decoys.py command line option, 129
 rna_rosetta_n.py command line option, 123, 125
 rna_ss_get_bps.py command line option, 105
 rna_standardize.py command line option, 38
 rna_torsions.py command line option, 45

A

aa3to1() (*in module rna_tools.rna_tools_lib*), 35
 add_ids() (*rna_tools.tools.clanstix.rna_clanstix.RNAStructClans method*), 143
 align_seq() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignmer method*), 68
 alignfn
 pdbs_measure_atom_dists.py command line option, 48
 renum_to_aln.py command line option, 88

alignment
 rna_alignment_calc_energy.py command line option, 75
 rna_alignment_get_species.py command line option, 73

AnalyzeGeometry (class in *rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry*), 117

Atom (class in *rna_tools.tools.simrna_trajectory.simrna_trajectory*), 133

B

best_energy (*rna_tools.tools.mq.FARNA.FARNA.FARNA* attribute), 115

BlastPDB (class in *rna_tools.BlastPDB*), 59

C

calc_rmsd() (in module *rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd*), 93

calc_rmsd() (in module *rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_paths*), 99

calc_rmsd_pymol() (in module *rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd*), 93

check_geometry() (*rna_tools.rna_tools_lib.RNAStructure* method), 27

check_symmetric() (in module *rna_tools.tools.clanstix.rna_clanstix*), 143

clarna_compare() (in module *rna_tools.tools.clarna_app.rna_clarna_app*), 108

clarna_run() (in module *rna_tools.tools.clarna_app.rna_clarna_app*), 109

ClashScore (class in *rna_tools.tools.mq.ClashScore.ClashScore*), 116

clean_up() (*rna_tools.tools.rna_x3dna.rna_x3dna.x3DNA* method), 107

cleanup() (*rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry.AnalyzeGeometry* method), 117

cleanup() (*rna_tools.tools.mq.ClashScore.ClashScore.ClashScore* method), 116

cleanup() (*rna_tools.tools.mq.FARNA.FARNA.FARNA* method), 115

cluster() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_cluster*), 127

cluster_loop() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_cluster*), 127

CMAAlign (class in *rna_tools.tools.rna_alignment.rna_alignment*), 85

cmscan() (*rna_tools.RfamSearch.RfamSearch* method), 61

collapsed_view() (in module *rna_tools.rna_tools_lib*), 35

colour_by_local_score() (class in *rna_tools.tools.mq.RASP.RASP.RASP* method), 109

convert_sali2dotbracket() (in module *rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket*), 147

copy_ss_cons_to_all() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 68

copy_ss_cons_to_all_editing_sequence() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 68

CustomFormatter (class in *rna_tools.tools.rna_rosetta.rna_rosetta_run*), 121

D

db_path (*rna_tools.tools.mq.FARNA.FARNA.FARNA* attribute), 115

describe() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 68

Dfire (class in *rna_tools.tools.mq.Dfire.Dfire*), 110

dir
 rna_rosetta_cluster.py command line option, 130

dist_from_matrix() (*rna_tools.tools.clanstix.rna_clanstix.RNAStructClans* method), 143

dist_from_matrix_mp() (*rna_tools.tools.clanstix.rna_clanstix.RNAStructClans* method), 143

do_job() (in module *rna_tools.tools.rna_calc_inf.rna_calc_inf*), 100

draw_circle() (in module *rna_tools.tools.pymol_drawing.pymol_drawing*), 155

draw_circle_selection() (in module *rna_tools.tools.pymol_drawing.pymol_drawing*), 156

draw_dist() (in module *rna_tools.tools.pymol_drawing.pymol_drawing*), 156

draw_dists() (in module *rna_tools.tools.pymol_drawing.pymol_drawing*), 156

draw_ss() (in module *rna_tools.SecondaryStructure*), 56

draw_ss() (*rna_tools.tools.rna_alignment.rna_alignment.RNASeq* method), 62

draw_vector() (in module *rna_tools.tools.pymol_drawing.pymol_drawing*), 156

156

E

`edit_occupancy_of_pdb()` (in module `rna_tools.tools.rna_pdb_edit_occupancy_bfactor`), 46

`edit_occupancy_of_pdb()` (`rna_tools.rna_tools_lib.RNAStructure` method), 28

`edit_pdb()` (in module `rna_tools.rna_tools_lib`), 27, 35

`eSCORE` (class in `rna_tools.tools.mq.eSCORE.eSCORE`), 112

`eval()` (`rna_tools.Seq.RNASequence` method), 53

`ExceptionOpenPairsProblem`, 56

`exe()` (in module `rna_tools.tools.mq.eSCORE.eSCORE`), 113

`executable` (`rna_tools.tools.mq.FARNA.FARNA.FARNA` attribute), 115

`executable` (`rna_tools.tools.mq.RASP.RASP.RASP` attribute), 109

`executable` (`rna_tools.tools.mq.RNAkb.RNAkb.RNAkb` attribute), 111

`extract()` (in module `rna_tools.tools.rna_rosetta.rna_rosetta_cluster`), 127

F

`FARNA` (class in `rna_tools.tools.mq.FARNA.FARNA`), 115

`fetch()` (in module `rna_tools.rna_tools_lib`), 26, 35

`fetch_ba()` (in module `rna_tools.rna_tools_lib`), 26, 35

`fetch_cif()` (in module `rna_tools.rna_tools_lib`), 36

`fetch_cif_ba()` (in module `rna_tools.rna_tools_lib`), 36

file

- `rna_add_chain.py` command line option, 47
- `rna_align_distance_to_seq.py` command line option, 77
- `rna_align_foldability.py` command line option, 82
- `rna_align_get_ss_from_fasta.py` command line option, 76
- `rna_align_get_ss_from_stk.py` command line option, 76
- `rna_cal_rmsd_trafl_plot` command line option, 97
- `rna_dot2ct.py` command line option, 57
- `rna_mq_rnakb.py` command line option, 112
- `rna_pdb_edit_occupancy_bfactor.py` command line option, 45
- `rna_pdb_tools.py` command line option, 17
- `rna_plot_density.py` command line option, 146
- `rna_plot_hist.py` command line option, 145

- `rna_rosetta_cluster.py` command line option, 126
- `rna_rosetta_extract_lowscore_decoys.py` command line option, 129
- `rna_rosetta_main.py` command line option, 128
- `rna_rosetta_n.py` command line option, 123, 125
- `rna_rosetta_run.py` command line option, 119
- `rna_ss_get_bps.py` command line option, 105
- `rna_standardize.py` command line option, 37
- `rna_torsions.py` command line option, 44

filename

- `rna_sali2dotbracket` command line option, 146

files

- `rna_calc_inf.py` command line option, 99
- `rna_calc_rmsd` command line option, 91

`find_core()` (in module `rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists`), 48

`find_core()` (`rna_tools.tools.rna_alignment.rna_alignment.RNAalignment` method), 68

`find_seq()` (`rna_tools.tools.rna_alignment.rna_alignment.RNAalignment` method), 69

`find_seq_exact()` (`rna_tools.tools.rna_alignment.rna_alignment.RNAalignment` method), 69

`fix_0_in_UC()` (`rna_tools.rna_tools_lib.RNAStructure` method), 28

`fix_op_atoms()` (`rna_tools.rna_tools_lib.RNAStructure` method), 28

`fix_with_qrnas()` (`rna_tools.rna_tools_lib.RNAStructure` method), 28

fn

- `rna_refinement.py` command line option, 138

fn (`rna_tools.rna_tools_lib.RNAStructure` attribute), 22, 25, 27, 39

`format_annotation()` (`rna_tools.tools.rna_alignment.rna_alignment.RNAalignment` method), 69

`Frame` (class in `rna_tools.tools.simrna_trajectory.simrna_trajectory`), 133

G

`generate_conect_records()` (in module `rna_tools.rna_tools_lib`), 36

`get_align()` (in module `rna_tools.tools.rna_alignment.random_assignment_of_nucleotides`), 84

`get_all_chain_ids()`

<code>(rna_tools.rna_tools_lib.RNAStructure</code>	<code>get_no_structures_in_first_cluster()</code> (in module
<code>method)</code> , 28	<code>rna_tools.tools.rna_rosetta.rna_rosetta_cluster)</code> ,
<code>get_atom_code()</code> (rna_tools.rna_tools_lib.RNAStructure	127
<code>method)</code> , 28	<code>get_parser()</code> (in module
<code>get_atom_coords()</code> (rna_tools.rna_tools_lib.RNAStructure	<code>rna_tools.tools.clanstix.rna_clanstix)</code> , 144
<code>method)</code> , 28	<code>get_parser()</code> (in module
<code>get_atom_num()</code> (rna_tools.rna_tools_lib.RNAStructure	<code>rna_tools.tools.clarna_app.rna_clarna_app)</code> ,
<code>method)</code> , 28	109
<code>get_atoms()</code> (rna_tools.tools.simrna_trajectory.simrna_trajectory	<code>get_parser()</code> (in module
<code>method)</code> , 134	<code>rna_tools.tools.cluster_load.cluster_load)</code> ,
<code>get_center()</code> (rna_tools.tools.simrna_trajectory.simrna_trajectory	<code>Residue</code>
<code>method)</code> , 134	<code>get_parser()</code> (in module
<code>get_ClaRNA_output_from_dot_bracket()</code> (in module	<code>rna_tools.tools.misc.rna_add_chain)</code> , 47
<code>rna_tools.tools.clarna_app.rna_clarna_app)</code> ,	<code>get_parser()</code> (in module
109	<code>rna_tools.tools.mq.Dfire.Dfire)</code> , 110
<code>get_clean_ss()</code> (rna_tools.tools.rna_alignment.rna_alignment	<code>RNAAlignment</code> (in module
<code>method)</code> , 69	<code>rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_d</code>
<code>get_conserved()</code> (rna_tools.tools.rna_alignment.rna_alignment	<code>RNASeq</code>
<code>method)</code> , 63	<code>get_parser()</code> (in module
<code>get_coord()</code> (rna_tools.tools.simrna_trajectory.simrna_trajectory	<code>Atom</code> <code>rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln)</code> ,
<code>method)</code> , 133	89
<code>get_distance_to()</code> (rna_tools.tools.rna_alignment.rna_alignment	<code>RNASeq</code> (in module
<code>method)</code> , 63	<code>rna_tools.tools.rna_alignment.random_assignment_of_nucleotide</code>
<code>get_distances()</code> (rna_tools.tools.rna_alignment.rna_alignment	<code>RNAAlignment</code>
<code>method)</code> , 69	<code>get_parser()</code> (in module
<code>get_dot_bracket_from_ClaRNAoutput()</code> (in module	<code>rna_tools.tools.rna_calc_inf.rna_calc_inf)</code> ,
<code>rna_tools.tools.clarna_app.rna_clarna_app)</code> ,	100
109	<code>get_parser()</code> (in module
<code>get_foldability()</code> (rna_tools.Seq.RNASequence	<code>rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd)</code> ,
<code>method)</code> , 53	94
<code>get_gc_rf()</code> (rna_tools.tools.rna_alignment.rna_alignment	<code>get_parser()</code> (in module
<code>method)</code> , 85	<code>rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all)</code> ,
<code>get_gc_rf()</code> (rna_tools.tools.rna_alignment.rna_alignment	<code>RNAAlignment</code>
<code>method)</code> , 70	<code>get_parser()</code> (in module
<code>get_info_chains()</code> (rna_tools.rna_tools_lib.RNAStructure	<code>rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_pk_to_s</code>
<code>method)</code> , 28	59
<code>get_ion_water_report()</code>	<code>get_parser()</code> (in module
<code>(rna_tools.tools.rna_x3dna.rna_x3dna.x3DNA</code>	<code>rna_tools.tools.rna_rosetta.rna_rosetta_check_progress)</code> ,
<code>method)</code> , 7, 107	130
<code>get_modifications()</code>	<code>get_parser()</code> (in module
<code>(rna_tools.tools.rna_x3dna.rna_x3dna.x3DNA</code>	<code>rna_tools.tools.rna_rosetta.rna_rosetta_cluster)</code> ,
<code>method)</code> , 7, 107	127
<code>get_multiple_lines()</code> (in module	<code>get_parser()</code> (in module
<code>rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_pk_to_s</code>	<code>tools.rna_rosetta.rna_rosetta_extract_lowscore_decoys</code>
59	129
<code>get_no_structures()</code> (in module	<code>get_parser()</code> (in module
<code>rna_tools.tools.rna_rosetta.rna_rosetta_cluster)</code> ,	<code>rna_tools.tools.rna_rosetta.rna_rosetta_head)</code> ,
127	126
<code>get_no_structures()</code> (in module	<code>get_parser()</code> (in module
<code>rna_tools.tools.rna_rosetta.rna_rosetta_min)</code> ,	<code>rna_tools.tools.rna_rosetta.rna_rosetta_min)</code> ,
129	129
<code>get_no_structures()</code> (in module	<code>get_parser()</code> (in module
<code>rna_tools.tools.rna_rosetta.rna_rosetta_n)</code> ,	<code>rna_tools.tools.rna_rosetta.rna_rosetta_n)</code> ,
123	123

L

lines (*rna_tools.rna_tools_lib.RNAStructure* attribute),
 22, 25, 27, 39
 load_fasta_ss_into_RNAseqs() (in module
rna_tools.Seq), 56
 load_from_file() (*rna_tools.tools.simrna_trajectory.simrna_trajectory*
 method), 134
 load_from_list() (*rna_tools.tools.simrna_trajectory.simrna_trajectory*
 method), 134
 load_from_string() (*rna_tools.tools.simrna_trajectory.simrna_trajectory*
 method), 134
 load_rnas() (in module *rna_tools.rna_tools_lib*), 36
 log_stdout_stderr()
 (*rna_tools.tools.mq.RNAkb.RNAkb.RNAkb*
 method), 111

M

main() (in module *rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry*),
 118
 main() (in module *rna_tools.tools.mq.eSCORE.eSCORE*),
 113
 main() (in module *rna_tools.tools.mq.RNA3DCNN.RNA3DCNN*),
 114
 main() (in module *rna_tools.tools.mq.RNAscore.RNAscore*),
 113
 main() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_run*),
 121
 map_coords_atom() (in module
rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists),
 49
 map_seq_on_align() (*rna_tools.tools.rna_alignment.rna_alignment*
 method), 70
 map_seq_on_seq() (*rna_tools.tools.rna_alignment.rna_alignment*
 method), 71
 max_seq_len (*rna_tools.tools.mq.RNA3DCNN.RNA3DCNN.RNA3DCNN*
 attribute), 114
 MethodNotChosen, 52
 MethodUnknown, 27
 min() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_min*),
 129
 module
 rna_tools.BlastPDB, 59
 rna_tools.RfamSearch, 60
 rna_tools.rna_tools_lib, 27
 rna_tools.SecondaryStructure, 56
 rna_tools.Seq, 51
 rna_tools.tools.clanstix.rna_clanstix,
 141
 rna_tools.tools.clarna_app.rna_clarna_app,
 108
 rna_tools.tools.cluster_load.cluster_load,
 148
 rna_tools.tools.diffpdb.diffpdb, 138

rna_tools.tools.extra_functions.select_fragment,
 171
rna_tools.tools.misc.rna_add_chain, 47
rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry,
 117
rna_tools.tools.mq.ClashScore.ClashScore,
 116
rna_tools.tools.mq.Dfire.Dfire, 110
rna_tools.tools.mq.eSCORE.eSCORE, 112
rna_tools.tools.mq.FARNA.FARNA, 115
rna_tools.tools.mq.RASP.RASP, 109
rna_tools.tools.mq.RNA3DCNN.RNA3DCNN, 113
rna_tools.tools.mq.RNAkb.RNAkb, 111
rna_tools.tools.mq.RNAscore.RNAscore, 113
rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_a
 48
rna_tools.tools.pymol_drawing.pymol_drawing,
 155
rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln,
 88
rna_tools.tools.rna_alignment.random_assignment_of_nuc
 83
rna_tools.tools.rna_alignment.rna_alignment,
 61
rna_tools.tools.rna_calc_inf.rna_calc_inf,
 100
rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd,
 92
rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all
 98
rna_tools.tools.rna_convert_pseudoknot_formats.rna_ss_
 59
rna_tools.tools.rna_filter.pymol_dists,
 105
rna_tools.tools.rna_rosetta.rna_rosetta_check_progress
 130
rna_tools.tools.rna_rosetta.rna_rosetta_cluster,
 127
rna_tools.tools.rna_rosetta.rna_rosetta_extract_lowscor
 129
rna_tools.tools.rna_rosetta.rna_rosetta_head,
 125
rna_tools.tools.rna_rosetta.rna_rosetta_min,
 128
rna_tools.tools.rna_rosetta.rna_rosetta_n,
 123
rna_tools.tools.rna_rosetta.rna_rosetta_run,
 120
rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracke
 146
rna_tools.tools.rna_x3dna.rna_x3dna, 6,
 106
rna_tools.tools.simrna_trajectory.simrna_trajectory,
 133

mqap() (*rna_tools.tools.mq.FARNA.FARNA.FARNA* method), 115

N

n
rna_rosetta_cluster.py command line option, 126

name (*rna_tools.rna_tools_lib.RNAStructure* attribute), 22, 25, 27, 39

nstruc
rna_rosetta_extract_lowscore_decoys.py command line option, 129

O

open_pdb() (in module *rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists*), 49

open_pdb() (in module *rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln*), 89

output
rna_align_distance_to_seq.py command line option, 77
rna_align_foldability.py command line option, 82

P

pairs
rna_pairs2SimRNArests.py command line option, 104

pairs1
rna_pairs_diff.py command line option, 106

pairs2
rna_pairs_diff.py command line option, 106

parse_vienna_to_pairs() (in module *rna_tools.SecondaryStructure*), 57

PDBFetchError, 27

pdbfm
renum_to_aln.py command line option, 88

pdbfm1
pdbs_measure_atom_dists.py command line option, 48

pdbfm2
pdbs_measure_atom_dists.py command line option, 48

pdbs_measure_atom_dists.py command line option
 --help, 48
 --verbose, 48
 -h, 48
 -v, 48
 alignfn, 48

pdbfm1, 48
 pdbfm2, 48
 seqid1, 48
 seqid2, 48

per_user() (in module *rna_tools.tools.cluster_load.cluster_load*), 148

plot() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 71

plot_cov() (*rna_tools.tools.rna_alignment.rna_alignment.RChie* method), 87

plot_energy() (*rna_tools.tools.simrna_trajectory.simrna_trajectory.SimrnaTrajectory* method), 134

predict_ss() (*rna_tools.Seq.RNASequence* method), 53

prepare_folder() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_run*), 121

prepare_helices() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_run*), 122

prepare_rosetta() (in module *rna_tools.tools.rna_rosetta.rna_rosetta_run*), 122

program_name (*rna_tools.tools.mq.FARNA.FARNA.FARNA* attribute), 115

program_name (*rna_tools.tools.mq.RASP.RASP.RASP* attribute), 109

program_name (*rna_tools.tools.mq.RNAscore.RNAscore.RNAscore* attribute), 113

R

random_assignment_of_nucleotides.py command line option
 --alignfn, 83
 --help, 83
 --outfn, 83
 --seqfn, 83
 --verbose, 83
 -h, 83
 -v, 83

RASP (class in *rna_tools.tools.mq.RASP.RASP*), 109

RChie (class in *rna_tools.tools.rna_alignment.rna_alignment*), 87

reload() (*rna_tools.rna_tools_lib.RNAStructure* method), 33

reload_alignment() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 71

remove() (*rna_tools.rna_tools_lib.RNAStructure* method), 33

remove_columns() (*rna_tools.tools.rna_alignment.rna_alignment.RNAscore* method), 63

remove_empty_columns() (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 71

method), 71
remove_gaps() (*rna_tools.tools.rna_alignment.rna_alignment.RNASeq*, *method*), 63
remove_ion() (*rna_tools.rna_tools_lib.RNAStructure* *method*), 33
remove_water() (*rna_tools.rna_tools_lib.RNAStructure* *method*), 33
rename_chain() (*rna_tools.rna_tools_lib.RNAStructure* *method*), 33
renum_atoms() (*rna_tools.rna_tools_lib.RNAStructure* *method*), 33
renum_to_aln.py command line option
 --help, 88
 --outfn, 88
 --residue_index_start, 88
 --verbose, 88
 -h, 88
 -v, 88
 alignfn, 88
 pdbfn, 88
 seqid, 88
renumber() (*in module rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln*), 89
repl() (*in module rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket*), 147
replace_atoms() (*in module rna_tools.rna_tools_lib*), 36
replace_chain() (*in module rna_tools.rna_tools_lib*), 37
Residue (*class in rna_tools.tools.simrna_trajectory.simrna_trajectory*), 133
RfamSearch (*class in rna_tools.RfamSearch*), 60
RfamSearchError, 61
rmsd_to() (*rna_tools.tools.simrna_trajectory.simrna_trajectory.Fram* *method*), 133
rmsds_fn
 rna_calc_evo_rmsd command line option, 97
RNA3DCNN (*class in rna_tools.tools.mq.RNA3DCNN.RNA3DCNN*), 114
rna_add_chain.py command line option
 --chain, 47
 --help, 47
 -c, 47
 -h, 47
 file, 47
rna_align_distance_to_seq.py command line option
 --help, 77
 -h, 77
 file, 77
 output, 77
rna_align_foldability.py command line option
 --all-stars, 82
 --help, 82
 --skip-mcfold, 82
 --verbose, 82
 -h, 82
 -v, 82
 file, 82
 output, 82
rna_align_get_ss_from_fasta.py command line option
 --help, 76
 -h, 76
 file, 76
rna_align_get_ss_from_stk.py command line option
 --help, 76
 -h, 76
 file, 76
rna_alignment_calc_energy.py command line option
 --csv, 75
 --debug, 75
 --flanks, 76
 --help, 76
 --loop-seq, 75
 --method, 75
 --one, 75
 --template, 75
 --verbose, 76
 -v, 76
 alignment, 75
rna_alignment_get_species.py command line option
 --debug, 73
 --evo-mapping, 73
 --evo-mapping-default, 74
 --help, 73
 --id-width, 73
 --one, 74
 --osfn, 74
 --rfam, 74
 --verbose, 73
 -h, 73
 -v, 73
 alignment, 73
rna_cal_rmsd_trafl_plot command line option
 --help, 97
 -h, 97
 file, 97
rna_calc_evo_rmsd command line option
 --help, 97
 -h, 97

```

    rmsds_fn, 97
    struc1, 97
    struc2, 97
    trafl, 97
rna_calc_inf.py command line option
  --debug, 100
  --dont-remove-sel-files, 100
  --force, 100
  --help, 99
  --ignore-files, 99
  --method, 100
  --model-selection, 100
  --no-stacking, 100
  --number-of-threads, 99
  --out_fn, 100
  --print-results, 100
  --renumber-residues, 100
  --sort-results, 100
  --ss, 100
  --target_fn, 99
  --target-selection, 100
  --verbose, 100
  --web, 100
  -f, 100
  -h, 99
  -m, 99
  -o, 100
  -pr, 100
  -s, 100
  -sr, 100
  -t, 99
  -v, 100
  files, 99
rna_calc_rmsd command line option
  --help, 91
  --ignore-files, 91
  --method, 92
  --model-ignore-selection, 91
  --model-selection, 91
  --name-rmsd-column, 92
  --print-progress, 92
  --print-results, 92
  --rmsds-fn, 92
  --sort-results, 92
  --target-column-name, 92
  --target-fn, 91
  --target-ignore-selection, 91
  --target-selection, 91
  --verbose, 92
  --way, 92
  -h, 91
  -m, 92
  -o, 92
  -pp, 92
  -pr, 92
  -sr, 92
  -t, 91
  -v, 92
  files, 91
rna_calc_rmsd_all_vs_all command line
  option
  --help, 98
  --input-dir, 98
  --matrix-fn, 98
  --method, 98
  -h, 98
  -i, 98
  -m, 98
  -o, 98
rna_calc_rmsd_multi_targets.py command line
  option
  --help, 95
  --model-selection, 95
  --models, 95
  --output-csv, 95
  --target-selection, 95
  --targets, 95
  --verbose, 95
  -h, 95
  -v, 95
rna_dca_mapping.py command line option
  --dca, 102
  --gseq, 102
  --help, 102
  --mss, 102
  --noshort, 102
  --noss, 102
  --offset, 102
  --seq, 102
  --verbose, 102
  -h, 102
rna_dot2ct.py command line option
  --help, 58
  --verbose, 58
  -h, 58
  -v, 58
  file, 57
rna_ec2x.py command line option
  --chain, 104
  --ec-pairs, 104
  --help, 104
  --pairs-delta, 104
  --sep, 104
  --ss-pairs, 104
  -h, 104
  interaction_fn, 104
rna_filter.py command line option
  --help, 101

```

```

--offset, 102
--restraints_fn, 102
--verbose, 102
-h, 101
-r, 102
-s, 102
-t, 102
-v, 102
rna_mq_rnakb.py command line option
--help, 112
--potential, 112
--verbose, 112
-h, 112
-p, 112
-v, 112
file, 112
rna_pairs2SimRNArestrs.py command line
option
--dist, 105
--help, 104
--offset, 104
--verbose, 105
--weight, 104
--well, 105
-h, 104
-v, 105
pairs, 104
rna_pairs_diff.py command line option
--help, 106
--verbose, 106
-h, 106
-v, 106
pairs1, 106
pairs2, 106
rna_pdb_edit_occupancy_bfactor.py command
line option
--bfactor, 45
--help, 45
--occupancy, 45
--output, 45
--select, 45
--select-atoms, 46
--set-not-selected-to, 45
--set-to, 45
--verbose, 46
-h, 45
-o, 45
file, 45
rna_pdb_tools.py command line option
--backbone-only, 21
--bases-only, 21
--chain-first, 20
--cif2pdb, 20
--clean, 18
--collapsed-view, 19
--color-seq, 19
--compact, 19
--conect, 17
--conect-no-linkage, 17
--cv, 19
--delete, 20
--delete-anisou, 17
--dont-fix-missing-atoms, 19
--dont-rename-chains, 19
--dont-report-missing-atoms, 21
--edit, 20
--extract, 20
--extract-chain, 20
--fasta, 20
--fetch, 18
--fetch-ba, 18
--fetch-chain, 18
--fetch-cif, 18
--fetch-fasta, 18
--fetch-header, 18
--fix, 18
--get-chain, 18
--get-rnapuzzle-ready, 21
--get-seq, 19
--get-ss, 19
--help, 17
--here, 21
--hide-warnings, 19
--ignore-files, 19
--inplace, 21
--inspect, 19
--is-nmr, 18
--is-pdb, 18
--keep-hetatm, 21
--mdr, 20
--mutate, 19
--nmr-dir, 18
--no-backbone, 21
--no-hr, 19
--no-progress-bar, 17
--oneline, 20
--orgmode, 18
--p-only, 21
--pdb2cif, 20
--ref-frame-only, 21
--remove0, 18
--rename-chain, 20
--renum-atoms, 17
--renum-nmr, 17
--renum-residues-dirty, 17
--renumber-residues, 19
--replace-chain, 20
--replace-hetatm, 21

```

```

--replace-htm, 20
--report, 17
--rgyration, 19
--rosetta2generic, 19
--rpr, 21
--save-single-res, 21
--set-chain, 20
--split-alt-locations, 18
--suffix, 21
--swap-chains, 20
--to-mol2, 18
--un-nmr, 18
--undo, 17
--uniq, 20
--verbose, 19
--version, 17
-c, 18
-h, 17
-r, 17
-v, 19
file, 17
rna_plot_density.py command line option
--column, 146
--help, 146
--output, 146
--sep, 146
-h, 146
-o, 146
file, 146
rna_plot_hist.py command line option
--bins, 145
--column, 145
--help, 145
--output, 145
--sep, 145
-h, 145
-o, 145
file, 145
rna_refinement.py command line option
--help, 138
--interactive, 138
--output_file, 138
--steps, 138
--verbose, 138
-h, 138
-i, 138
-o, 138
-s, 138
-v, 138
fn, 138
rna_rosetta_cluster.py command line option
--help, 126, 130
--kill, 130
--limit-clusters, 127
--min-only, 130
--no_select, 126
--radius-inc-step, 127
--select, 130
--verbose, 130
-h, 126, 130
-k, 130
-m, 130
-s, 130
-v, 130
dir, 130
file, 126
n, 126
rna_rosetta_extract_lowscore_decoys.py
    command line option
--help, 129
--verbose, 129
-h, 129
-v, 129
file, 129
nstruc, 129
rna_rosetta_min.py command line option
--cpus, 128
--go, 128
--help, 128
-c, 128
-g, 128
-h, 128
file, 128
rna_rosetta_n.py command line option
-0, 123
--help, 123, 125
--nstruc, 125
--verbose, 123, 125
--zero, 123
-h, 123, 125
-n, 125
-v, 123, 125
file, 123, 125
rna_rosetta_run.py command line option
--cpus, 120
--go, 120
--helices, 120
--help, 120
--init, 120
--motif, 120
--nstruc, 120
--rosetta, 120
--sandbox, 120
-c, 120
-e, 120
-g, 120
-h, 120
-i, 120

```

-m, 120
 -n, 120
 -r, 120
 file, 119
 rna_sali2dotbracket command line option
 --help, 146
 -h, 146
 filename, 146
 rna_simrna_extract.py command line option
 --cleanup, 131
 --help, 131
 --number_of_structures, 131
 --template, 131
 --trafl, 131
 -c, 131
 -f, 131
 -h, 131
 -n, 131
 -t, 131
 rna_simrna_lowest.py command line option
 --help, 131
 --nstruc, 131
 -h, 131
 -n, 131
 trafl, 131
 rna_simrnaweb_download_job.py command line option
 --cluster, 133
 --download-trajectory, 133
 --extract, 133
 --help, 132
 --more_clusters, 133
 --nstruc, 132
 --prefix, 132
 --remove-trajectory, 133
 --top100, 133
 --top200, 133
 --web-models, 133
 -c, 133
 -d, 133
 -e, 133
 -h, 132
 -m, 133
 -n, 132
 -p, 132
 -r, 133
 job_id, 132
 rna_ss_get_bps.py command line option
 --help, 105
 --offset, 105
 --verbose, 105
 -h, 105
 -v, 105
 file, 105
 rna_standardize.py command line option
 --backbone-only, 38
 --bases-only, 39
 --check-geometry, 38
 --cif, 38
 --conect, 38
 --conect-no-linkage, 38
 --dont-fix-missing-atoms, 38
 --dont-rename-chains, 38
 --dont-replace-hetatm, 38
 --dont-report-missing-atoms, 38
 --help, 37
 --here, 38
 --inplace, 38
 --keep-hetatm, 38
 --mdr, 38
 --no-backbone, 39
 --no-hr, 38
 --no-progress-bar, 37
 --quiet, 38
 --renum-nmr, 38
 --renumber-residues, 38
 --suffix, 38
 --verbose, 38
 --version, 37
 -h, 37
 -v, 38
 file, 37
 rna_tools.BlastPDB
 module, 59
 rna_tools.RfamSearch
 module, 60
 rna_tools.rna_tools_lib
 module, 27
 rna_tools.SecondaryStructure
 module, 56
 rna_tools.Seq
 module, 51
 rna_tools.tools.clanstix.rna_clanstix
 module, 141
 rna_tools.tools.clarna_app.rna_clarna_app
 module, 108
 rna_tools.tools.cluster_load.cluster_load
 module, 148
 rna_tools.tools.diffpdb.diffpdb
 module, 138
 rna_tools.tools.extra_functions.select_fragment
 module, 171
 rna_tools.tools.misc.rna_add_chain
 module, 47
 rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry
 module, 117
 rna_tools.tools.mq.ClashScore.ClashScore
 module, 116

rna_tools.tools.mq.Dfire.Dfire module, 110	rna_torsions.py command line option --help, 44
rna_tools.tools.mq.eSCORE.eSCORE module, 112	--verbose, 45
rna_tools.tools.mq.FARNA.FARNA module, 115	-h, 44
rna_tools.tools.mq.RASP.RASP module, 109	-v, 45
rna_tools.tools.mq.RNA3DCNN.RNA3DCNN module, 113	file, 44
rna_tools.tools.mq.RNAkb.RNAkb module, 111	RNAalignment (class in rna_tools.tools.rna_alignment.rna_alignment), 67
rna_tools.tools.mq.RNAScore.RNAScore module, 113	RNAkb (class in rna_tools.tools.mq.RNAkb.RNAkb), 111
rna_tools.tools.pdbs_measure_atom_dists.pdbs_measure_atom_dists module, 48	RNAScore (class in rna_tools.tools.mq.RNAScore.RNAScore), 113
rna_tools.tools.pymol_drawing.pymol_drawing module, 155	RNASeq (class in rna_tools.tools.rna_alignment.rna_alignment), 62
rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln module, 88	RNASequences (class in rna_tools.Seq), 52
rna_tools.tools.rna_alignment.random_assignment_of_nucleotides module, 83	RNAStructClans (class in rna_tools.tools.clanstix.rna_clanstix), 143
rna_tools.tools.rna_alignment.rna_alignment module, 61	RNAStructure (class in rna_tools.rna_tools_lib), 22, 25, 27, 39
rna_tools.tools.rna_calc_inf.rna_calc_inf module, 100	rosetta_extract_lowscore_decoys() (in module rna_tools.tools.rna_rosetta.rna_rosetta_extract_lowscore_decoys), 129
rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd module, 92	run() (in module rna_tools.tools.rna_rosetta.rna_rosetta_cluster), 127
rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_align() module, 98	run() (in module rna_tools.tools.rna_rosetta.rna_rosetta_head), 126
rna_tools.tools.rna_convert_pseudoknot_formats.pseudoknot_to_sashima module, 59	run() (in module rna_tools.tools.rna_rosetta.rna_rosetta_min), 129
rna_tools.tools.rna_filter.pymol_dists module, 103	run() (in module rna_tools.tools.rna_rosetta.rna_rosetta_n), 123
rna_tools.tools.rna_rosetta.rna_rosetta_check_progress module, 130	run() (in module rna_tools.tools.mq.AnalyzeGeometry.AnalyzeGeometry.AnalyzeGeometry), 117
rna_tools.tools.rna_rosetta.rna_rosetta_cluster module, 127	run() (rna_tools.tools.mq.ClashScore.ClashScore.ClashScore method), 116
rna_tools.tools.rna_rosetta.rna_rosetta_extract_lowscore_decoys module, 129	run() (rna_tools.tools.mq.Dfire.Dfire.Dfire method), 110
rna_tools.tools.rna_rosetta.rna_rosetta_head module, 125	run() (rna_tools.tools.mq.eSCORE.eSCORE.eSCORE method), 113
rna_tools.tools.rna_rosetta.rna_rosetta_min module, 128	run() (rna_tools.tools.mq.FARNA.FARNA.FARNA method), 115
rna_tools.tools.rna_rosetta.rna_rosetta_n module, 123	run() (rna_tools.tools.mq.RASP.RASP.RASP method), 110
rna_tools.tools.rna_rosetta.rna_rosetta_run module, 120	run() (rna_tools.tools.mq.RNA3DCNN.RNA3DCNN.RNA3DCNN method), 114
rna_tools.tools.rna_sali2dotbracket.rna_sali2dotbracket module, 146	run() (rna_tools.tools.mq.RNAkb.RNAkb.RNAkb method), 111
rna_tools.tools.rna_x3dna.rna_x3dna module, 6, 106	run() (rna_tools.tools.mq.RNAScore.RNAScore.RNAScore method), 113
rna_tools.tools.simrna_trajectory.simrna_trajectory module, 133	run_cmdalign() (rna_tools.tools.rna_alignment.rna_alignment.CMAlign method), 86
	run_cmd() (in module rna_tools.tools.rna_rosetta.rna_rosetta_check_progress), 130
	run_cmd() (in module rna_tools.tools.rna_x3dna.rna_x3dna.x3DNA method), 8, 108

S

- `sandbox()` (*rna_tools.tools.mq.FARNA.FARNA.FARNA* method), 115
- `save()` (*rna_tools.tools.simrna_trajectory.simrna_trajectory.SimRNATrajectory* method), 135
- `search()` (*rna_tools.BlastPDB.BlastPDB* method), 60
- `select_pdb_fragment()` (in module *rna_tools.tools.extra_functions.select_fragment*), 171
- `select_pdb_fragment_pymol_style()` (in module *rna_tools.tools.extra_functions.select_fragment*), 171
- `seq_no_gaps` (*rna_tools.tools.rna_alignment.rna_alignment.RNASeq* attribute), 62
- `seqid`
 `renum_to_aln.py` command line option, 88
- `seqid1`
 `pdb_measure_atom_dists.py` command line option, 48
- `seqid2`
 `pdb_measure_atom_dists.py` command line option, 48
- `set_atom_code()` (*rna_tools.rna_tools_lib.RNAStructure* method), 33
- `set_atom_coords()` (*rna_tools.rna_tools_lib.RNAStructure* method), 34
- `set_atom_occupancy()`
 (*rna_tools.rna_tools_lib.RNAStructure* method), 34
- `set_chain_for_struct()` (in module *rna_tools.rna_tools_lib*), 37
- `set_occupancy_atoms()`
 (*rna_tools.rna_tools_lib.RNAStructure* method), 34
- `set_res_code()` (*rna_tools.rna_tools_lib.RNAStructure* method), 34
- `show()` (*rna_tools.tools.rna_alignment.rna_alignment.RChie* method), 88
- `SimRNATrajectory` (class in *rna_tools.tools.simrna_trajectory.simrna_trajectory*), 134
- `sort()` (*rna_tools.tools.simrna_trajectory.simrna_trajectory.SimRNATrajectory* method), 135
- `sort_nicely()` (in module *rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd*), 94
- `sort_nicely()` (in module *rna_tools.tools.rna_calc_rmsd.rna_calc_rmsd_all_vs_all*), 99
- `sort_strings()` (in module *rna_tools.rna_tools_lib*), 37
- `src_bin` (*rna_tools.tools.mq.FARNA.FARNA.FARNA* attribute), 115
- `ss_cons_std` (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* property), 71
- `ss_cons_with_pk` (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* property), 72
- `ss_cons_with_pk_std` (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* property), 72
- `ss_no_gaps` (*rna_tools.tools.rna_alignment.rna_alignment.RNASeq* attribute), 62
- `ss_to_bps()` (*rna_tools.tools.rna_alignment.rna_alignment.RNASeq* method), 67
- `stats_for_cluster()` (in module *rna_tools.tools.cluster_load.cluster_load*), 148
- `stats_for_user()` (in module *rna_tools.tools.cluster_load.cluster_load*), 148
- `std_resn()` (*rna_tools.rna_tools_lib.RNAStructure* method), 34
- `struc1`
 `rna_calc_evo_rmsd` command line option, 97
- `struc2`
 `rna_calc_evo_rmsd` command line option, 97
- `subset()` (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 72
- ## T
- `tail()` (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 72
- `test()` (in module *rna_tools.tools.mq.ClashScore.ClashScore*), 116
- `test()` (in module *rna_tools.tools.mq.Dfire.Dfire*), 110
- `trafl`
 `rna_calc_evo_rmsd` command line option, 97
- `rna_simrna_lowest.py` command line option, 131
- `trimmed_rf_and_ss()`
 (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 72
- ## U
- `un_nmr()` (*rna_tools.rna_tools_lib.RNAStructure* method), 34
- ## W
- `write()` (*rna_tools.rna_tools_lib.RNAStructure* method), 35
- `write()` (*rna_tools.tools.rna_alignment.rna_alignment.RChie* method), 88
- `write()` (*rna_tools.tools.rna_alignment.rna_alignment.RNAalignment* method), 72
- `write_align()` (in module *rna_tools.tools.rna_alignment.random_assignment_of_nucleotide*), 84

`write_seq()` (in *module*
rna_tools.tools.rna_alignment.random_assignment_of_nucleotides),
84

`write_struct()` (in *module*
rna_tools.tools.renum_pdb_to_aln.renum_pdb_to_aln),
89

X

`x3DNA` (class in *rna_tools.tools.rna_x3dna.rna_x3dna*), 7,
107

`x3DNAMissingFile`, 8, 108