

MC-Symizer user manual

*Stephen Leong Koan
RNA Engineering Laboratory
Institute for Research in Immunology and Cancer (IRIC)
Université de Montréal*

2014-Feb-14

Introduction

MC-Symizer is a new version of mcsymize by Marc Parisien which is used to automatically generate a MC-Sym script.

The main difference between this program and its predecessor is the addition of an annealed mode which allows to model two RNA sequences that anneal to each other.

Installation

Platform

MC-Symizer was used on Linux, Windows and MacOSX.

Requirements

To generate an MC-Sym script using MC-Symizer, the followings are needed:

- Python 2.7 or newer installed on your computer
- a copy of MC-Sym's NCM database (usually named "MCSYM-DB")
- the sequence(s)
- the secondary structure(s) in the dot-bracket notation

Running MC-Symizer

MC-Symizer can be called using

```
mcsymizer.py --sequence1 SEQUENCE1 --structure1 STRUCTURE1 --db_path DB_PATH [options]
```

Here is the complete list of parameters that can be given to MC-Symizer, the order in which they are declared is not important (i.e. `mcsymizer.py --sequence1 AGGUUUGUUUA --structure1 "((((...)))".` is same as `mcsymizer.py --structure1 "((((...)))".` --sequence1 AGGUUUGUUUA).

The mandatory parameters are:

Keyword	Alternative Keyword	Option
--sequence1	-s1	An RNA sequence. The following characters are accepted: A, U, T, G, C, a, u, t, g, c
--structure1	-S1	An RNA structure in the dot bracket notation (the same format as an MC-Fold output)
--db_path	-D	The path to the MCSYM-DB directory

The optional parameters are:

Keyword	Alternative Keyword	Option	Comment
--sequence2	-s2	An RNA sequence. The following characters are accepted: A, U, T, G, C, a, u, t, g, c	The annealing sequence to sequence1
--structure2	-S2	An RNA structure in the dot bracket notation (the same format as an MC-Fold output)	The annealing structure to structure1
--name	-n	Name given to the model. Default is "structure"	The name of the structures generated by MC-Sym is set by this parameter (i.e. if "test" is used, the generated models will be named <code>test-xxxx.pdb.gz</code>)
--help	-h		Display the help text on the terminal
--use_high_res_ncm			Use high-resolution NCMs for canonical stacks of base pairs
--merge_rmsd	-mr	The Merge RMSD threshold in Angstrom. Default is 1.5	The quality of the welding of two consecutive NCM is controlled through this parameter
--model_diversity	-md	The Model Diversity threshold in Angstrom. Default is 3.0	This parameter controls the resemblance of generated models

--clash_thresho ld	-ct	The Clash Threshold in Angstrom. Default is 1.5	This parameter prevents non- bonded atoms to be too close to one another
--bond_thresho ld	-bt	The Bond Threshold in Angstrom. Default is 2.0	The bond threshold for covalent bonds in the backbone
--construction_ method	-cm	The accepted values are "ccm" or "estimate". Default is "ccm"	"ccm" stands for Cyclic Coordinate Minimization. "estimate" stands for Interpolation Estimation
--exploration_ method	-em	The accepted values are "probabilistic" or "exhaustive". Default is "probabilistic"	"probabilistic" allows back-jumps and random domain assignments. "exhaustive" is a classic back-track algorithm
--max_number	-mn	Default is 1000	The maximum number of models generated by MC-Sym
--timeout	-t	The timeout in minutes. Default is 30	MC-Sym will stop its execution when the timeout is reached or when the maximum number of models is reached
--no_dangling	-nd		Do not include dangling ends in the models
--unzipped	-u		If this option is activated, the output structures will not be in a compressed format
--external_libra ry	-el		Use the pdb.gz fragment as library, at defined position (e.g. fragment1.pdb.gz,5,7,21,23;fr agment2.pdb.gz,9,12,17,19). N.B: we recommend protecting the input string using quotes ("")

--use_relative_ path	-ur		Use relative path to the MCSYM-DB directory (legacy behavior)
---------------------------------	------------	--	--

Additional information about the parameters can be found in pages 36-37 of the MC-Pipeline manual (<http://www.major.irc.ca/MC-Pipeline/manual.pdf>).

FAQ

Q: Why the models generated using mcsymize and MC-Symizer are not always identical?

A: This can be due to various reasons; the first being that MC-Symizer uses all the NCMs that are included in the MCSYM-DB directory (that's why the path to MCSYM-DB has to be given in the parameters); the second being a difference in the computation of the distance constraints, MC-Symizer allows "looser" constraints (using the equation found here http://www.major.irc.ca/MC-Sym/faq.html#section_dc).

Q: Does MC-Symizer support pseudoknots?

A: The answer is YES. However, modeling a pseudoknot is not a trivial problem. You might need to edit the script yourself to obtain better results. A short guide on modeling a pseudoknot can be found here http://www.major.irc.ca/MC-Sym/faq.html#section_pk.

Troubleshooting

Read LibraryFG is empty

This is probably due to the corresponding NCM being empty. We recommend that you look at the output of MC-SYM and identify the last NCM read by MC-Sym, delete it from MCSYM-DB and generate a new script using MC-Symizer.

Warning: skipped model #XX from file ...

This can be due to a malformed model in the NCM. You can edit the corresponding pdb file in MCSYM-DB or leave it be as this warning is harmless.