

RNAevol

RNAevol is a program to simulate the evolution of RNA strings in a chemostat-like environment. You start with a homogeneous population of RNA strings. Individual strings can replicate once every time step, the probability that they replicate corresponds to their fitness. Every time step an average growth rate for the whole population is calculated, and according to this growth rate, the dilution rate is determined. The remaining population replicates according to fitness and after replication the total population size will be more or less the original population size again. Fitness can be defined based on a distance to either a target structure or to a target stability, i.e. a target energy.

Options

You can run the program by going to the directory called RNAevol and typing *RNAevol*. In this way, the program is run with its default settings. You can look at these settings and how to change them by typing *RNAevol* and something arbitrary behind it, e.g. '*RNAevol q*'. This will print the following:

```
-Display (1: graphical, 0: non-graphical) default=1
-Maxtime : default=2147483647
-Mut : (mutation rate) default=0.001000
-Pop : (target population size) default=1000
-Sigma : (fitness increase factor)/(1 structure distance) or /(1.(J/mol))
default=1.500000
-Seed : (random seed) default=57
-XmgraceFileName : default=plot-RNAevol.agr
-InitialSeq : default=random
-TargetStr : default = "(((((((...(((.....))))).((((.....)))))......((((.....))))).))))).)". If
"random", then the target is the structure of randomly created sequence (in this
case, set -Length also).
-TargetEne : default=-40.000000. If -InitialSeq is not set, set -Length.
-Selection : (criteria) 0: target structure, 1: target energy. default=0
-Length : (sequence length) It must be 0<x<150. default=50
```

If you want to evolve a population of 2000 RNA strings to a target energy of -30, using a mutation rate of 0.025 and write the graphical output to a file called "/myDir/outGraphRNA", you type '*RNAevol -Pop 2000 -Mut 0.025 -Selection 1 -TargetEne -30 -XmgraceFileName "/home/myDir/outGraphRNA"*'. If you make a typing error, the list above is printed.

You can redirect the output printed in your terminal by RNAevol to a file by typing > "fileName" behind your command, e.g. : *RNAevol -Pop 2000 -Mut 0.025 -Selection 1 -TargetEne -30 -XmgraceFileName "/home/myDir/outGraphRNA" > "/home/myDir/outputRNA"*

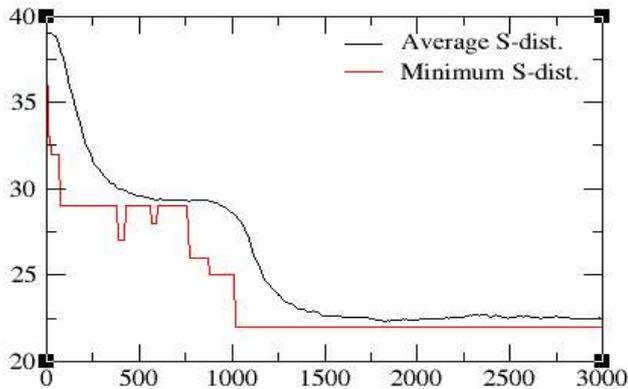
Let's go over all of the options and see what they mean.

- Display: The program outputs information on the most fit and the most abundant string to your terminal. If you set `-Display 0`, this is all the output you will get. If you set `-Display 1`, which is the default, you also get graphical output in Xmgrace. For more information on this graphical output see the separate section "Graphs" in this manual.
- Maxtime: The amount of time steps you want the program to run.
- Mut: The probability of a mutation per base. A mutation is a base changing into an other base, e.g. an 'A' changing into a 'T'.
- Pop: The population size. Note that the number of RNA strings in the population will not always be exactly this number.
- Sigma: This is a selection strength parameter. It gives you how much the fitness will increase when an RNA string gets one step closer to the target. The default is 1.5, this means that when one step closer to the target, the fitness and therefore replication speed of the RNA string will be multiplied by 1.5. How 'one step closer' is defined, depends on whether the RNA strings evolve to a target structure or a target energy. More information on distance measures can also be found in the "Graphs" section in this manual.
NOTE that this sigma is not the same as sigma selection coefficient found in the literature!
- Seed: The seed of the random number generator.
- XmgraceFileName: The name of the file to which Xmgrace writes the graphical output of RNAevol.
- InitialSeq: If you want RNAevol to start with a specific initial sequence, you can provide it here. The initial population will consist of n copies of this sequence (n is population size). Default is a randomly generated sequence of specified (with `-Length`) or default length (= 50).
- TargetStr: If you want the RNA strings to evolve to a particular structure, you can provide it here. You can also put *random* here. Then a random string is generated and the structure of that string is taken as a target structure.
NOTE that if you also specify an `-InitialSeq` with a length different from the default length (see `-Length`, default is 50), you have to set the length with `-Length` to the length of the sequence you provided with `InitialSeq`.
- TargetEne: If you want the RNA strings to evolve to a particular stability, you can provide it here.
- Selection: If you want fitness to be based on distance to a target structure you should put 0 here. If you want it be based on distance towards a stability you should put 1.
- Length: The length of the sequences. It must be $0 < x < 150$. Note that sequence can not change in length because mutations are only changing a base into an other, not inserting or deleting bases.

Graphs

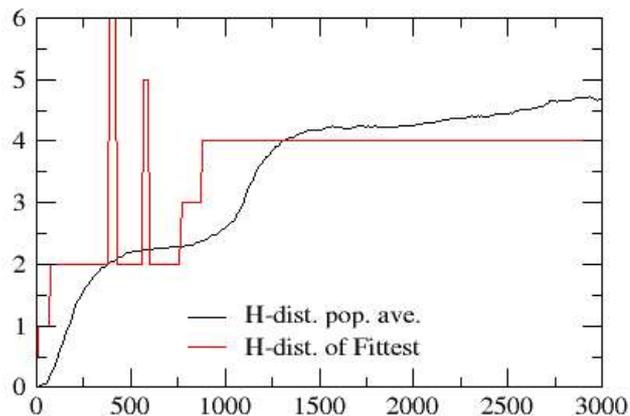
The graphical output by Xmgrace consists of four graphs. All the graphs have time on their x-axis.

In the left upper corner you will find a graph like this:



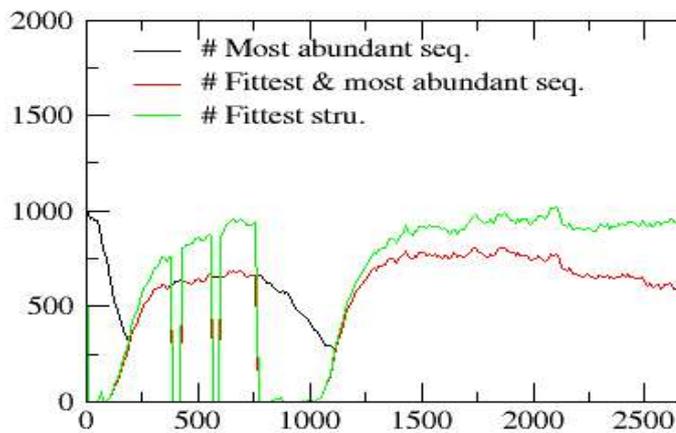
This graph shows you the average and the minimum distance to the target structure. The distance is measured as the number of times one has to open and close brackets to go from one structure to the other. If you evolve towards a target energy, the average and optimal energy is plotted. Then the distance is just the difference between the energy of a RNA string and the target energy in J/mol.

Underneath this graph you will find:



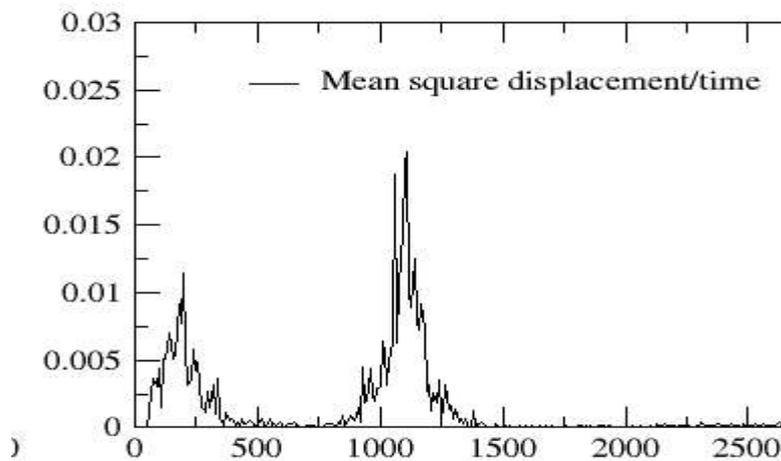
This graph plots the Hamming distance between the initial sequence and the fittest sequence and the average Hamming distance from the initial sequence of the whole population. Hamming distance between two sequences is the number of positions in which the sequences differ.

In the right upper corner:



The black line tells how many of the most abundant sequence are there. The green line plots the amount of RNA strings that have the structure that is closest to the target structure (or energy phenotype closest to target energy, when evolving towards a target energy) . The red line plots most abundant sequence within the group of RNA strings that have the fittest structure (or energy phenotype)

And finally under this graph, in the right lower corner:



Every time step a 4 x *sequencelength* table is made, containing the frequencies of bases at every position in the sequence.

<i>position</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	..
A	0.55	0.18	0.61
U	0.1	0.25
T	0.3	0.33
G	0.05	0.24
Σ	1	1	1	1	1	1	1	1	1	1	1	1	1	..

The mean square displacement / time is the mean squared difference between tables of subsequent time steps.

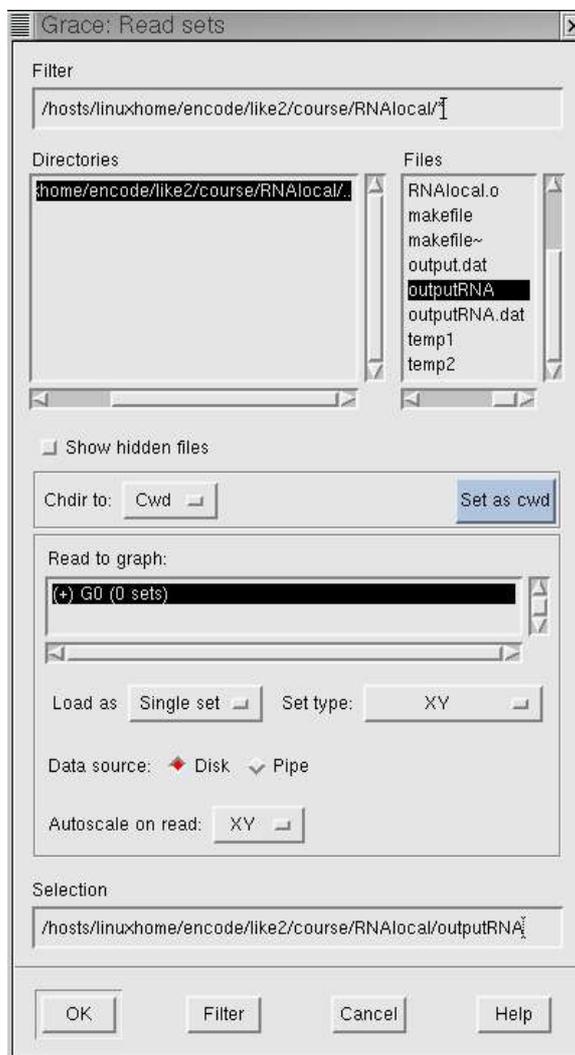
RNA local

RNAlocal is program that gives all one step mutants of a given string, their distance to a target structure and their energy. You can start the program by typing `RNAlocal -InitialSeq ** copy-paste the sequence from which you want all one step mutants here ** -TargetStr ** copy-paste the structure to which you want to compare the structure of the mutants here **`. The program then prints for each mutant:

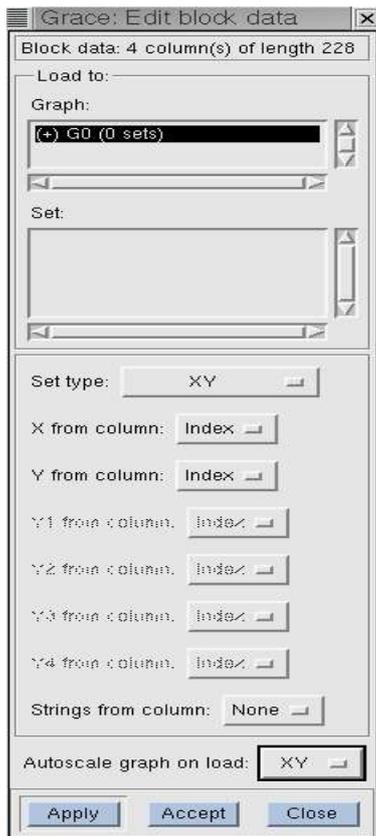
`sequence -tab- structure -tab- distance to target structure -tab- energy`

It is wise to redirect this output to a file by typing `> 'fileName'` behind the command so you can look at it again.

You can make a graph in Xmgrace by typing `xmgrace` and select from the menu `Data -> Import -> ASCII data`. You then get a menu like this:



Here you can choose the file in which you saved your RNAlocal output data. You should load this data as 'Block data', because you have several columns. Change '*Single set*' (a bit under the middle in this picture) into '*Block data*' and say 'OK'. You then get a window like this (see next page):



Here you can choose which column you want on which axis. So if you want to see what are the energies of one step mutants, you put column 4 on the y axis and 'index' (which is just 1,2,3,... *number of mutants*) on the x-axis.