REvolver: Modeling sequence evolution under domain constrains.

Manual for Version 1.0, April 2011.

Tina Koestler[1]

email: tina.koestler(at)univie.ac.at

Arndt von Haeseler[1]

email: arndt.von.haeseler(at)univie.ac.at

Ingo Ebersberger[1]

email: ingo.ebersberger(at)univie.ac.at

[1] Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, AUSTRIA.

Contents

1 License Agreement

2 Availability and Requirements

- `Revolver` is written in java - you need to have java 6 installed on your computer.
- The HMMER3 software package (Finn, Clements and Eddy, 2010) need to be installed. `hmmscan` and `hmmfetch` from the HMMER3 package are used to annotate a sequence with protein domains in a pHMM database and to extract a pHMM from a pHMM, respectively. Please note, `hmmscan` need to be run with the option **`-notextw`**.
- Refer to the program website `http://www.cibiv.at/software/revolver` for the program manual, the jar file of `Revolver` and example data.

3 Introduction

`REvolver` is a program to simulate protein sequence evolution. `REvolver` automatically integrates domain information described by a profile Hidden Markov Model (pHMM) into the simulation. In the simulation of protein evolution it often had been assumed that sites evolve identically and independently from each other. This simplification is necessary since information concerning site specific evolution is frequently unavailable. However, homologous sequences and domains have been collected, aligned, and pHMMs built. The pHMM describes the variability and shared characteristics of sequences that share a common ancestor. Here we do have knowledge about what sites are conserved, at what positions in the sequences insertions are more likely, or what sites can be deleted. Pfam (Finn et al., 2010) and SMART (Letunic, Doerks and Bork, 2009) are examples for databases providing such data. `REvolver` is the first method, for simulating protein sequence evolution that integrates this pre-existing information about evolution in an automatic fashion.

4 Method

The method is presented in:

Tina Koestler, Arndt von Haeseler and Ingo Ebersberger (2011) REvolver: Modeling sequence evolution under domain constraints. *in preparation.*

5 Input files

5.1 Tree

The input tree needs to be in Newick format. Inner nodes need to be labeled, if you want to change parameter values for different branches (`LINAGE SPECIFIC EVOLUTION`; Section 6.1.2).

5.2 pHMM database

The hmm database has to be built with

```
hmmbuilt hmmfile.hmm input_alignment.fa
```

where the `input_alignment.fa` file needs to be in `STOCKHOLM` format. You can also download a HMM database for example from `ftp://ftp.sanger.ac.uk/pub/databases/Pfam`. Next you need to build a binary file from your HMM database using

```
hmmpress hmmfile.hmm
```

For more details about HMMER3 please see `http://hmmer.janelia.org/`.

5.3 Substitution model

The following amino acid substitution models are implemented in `REvolver`: JTT (Jones, Taylor and Thornton, 1992), JTT_dcmut (Kosiol and Goldman, 2005), Dayhoff (Dayhoff, Schwartz and Orcutt, 1978), Dayhoff_dcmut (Kosiol and Goldman, 2005), WAG (Whelan and Goldman, 2001), mtMAM (Yang, Nielsen and Hasegawa, 1998), mtART (Abascal, Posada and Zardoya, 2007), mtREV (Adachi and Hasegawa, 1996), cpREV (Adachi et al., 2000), Vt (Müller and Vingron, 2000), Blosum62 (Henikoff and Henikoff, 1992), LG (Le and Gascuel, 2008), HIVb (Nickle et al., 2007), HIVw (Nickle et al., 2007).

You can also specify your own substitution model in the following format: Starting with the lower triangle of the exchangeability matrix $R = \{r_{ij}\}$, two empty lines, then the equilibriums frequencies π_i, two empty lines, finally a line of the alphabet in one letter code ordered according to their occurrence in the equilibriums frequency as well as in the exchangeability matrix.

5.4 Root sequence

The user-defined root sequence needs to be provided in fasta format. If you want to simulate under domain constraints you also have to provide a HMM annotation file of the root sequence. Such an annotation file can be produced by:

```
hmmscan --notextw hmmfile.hmm root_sequence.fa > root_sequence.hmm
```

The `hmmfile.hmm` can be a single pHMM or a pHMM database. Please do not forget to run `hmmscan` with the option **`--notextw`**.

Alternatively, the root sequence can be randomly generated. To this end, you can define a domain architecture: i.e. the linear order of domain names and lengths of linker regions (unconstrained segments). The domain architecture can be made out of one domain, several domains, unconstrained segments, or any combination of domains and unconstrained segments. To generate domain instances, REvolver extracts the pHMM

from the pHMM database (make sure that the domain names fit those in the pHMM database) and randomly passes through the pHMM. By that amino acids are emitted according to the emission probabilities of the pHMM state. For unconstrained segments we sample amino acids from the equilibriums frequency of the substitution model Q.

6 Starting `REvolver` and available options

6.1 Interactive terminal

To start `REvolver` using the interactive terminal move into the directory where REvolver is located and type:

```
java -cp "REvolver.jar" revolver
```

or use the path to the java file:

```
java -cp "path/to/REvolver.jar" revolver
```

We recommend to set the parameters for `REvolver` using the interactive terminal. As soon as all parameters are set, the program produces a file called `input_parameter.xml`. This file contains all the parameters in xml format. For any further use or for the integration of `REvolver` in a pipline you can start the program with the xml file as input (Section 6.2).

6.1.1 Required input parameters

You will be asked the following questions:

- `Please enter a file name for the tree data:`
- `Please enter a file name for root sequence OR random sequence [R]:`
  The root sequence can either be user-defined or random. In the first case, enter the name of the fasta formatted root file.
  - `For simulations under domain constraints enter a name for the root annotation file OR [Q]:`
    If you enter `Q`, the root sequence will be simulated without domain constraints otherwise the root sequence will be simulated under domain constrains. To this end, all significant domains in the root annotation file are considered. Next, you will be asked:
  - `Please enter file name for hmm database:`

  In the second case, enter `R`.
  - `Please enter file name for hmm database:`
  - `Random sequence [R] OR enter domain name:`
    - `R`
      - `Please enter length:`
        Enter the number of amino acids that will be chosen randomly according to the equilibriums frequency given in the evolutionary model.
    - `domain name`

- `Domain out of only match states (perfect) [P] OR random path through HMM [R]:`
  Both options generate a sequence by passing through the pHMM. The path in the `P` option starts with M_1, emits an amino acid proportional to the state M_1 specific emission probabilities and proceeds with the next match state. The path enters solely and every match state till the silent *End* state is reached. The path with the `R` option starts with the silent *Start* state. The next state is chosen randomly proportional to the transition probabilities. If the path enters a match or an insertion state, an amino acid will be emitted according to the state's specific emission probabilities. The path ends as soon as the silent *End* state is reached.

  Next, you have again the option to add further sequence parts, either random sequences or domains, to the root sequence. Pressing `Q` terminates the definition of the root sequence.

- `Please enter output path:`
  This is the path where all output files will be stored.

Now, the minimal number of parameters is set. All other options are preselected and displayed in a menu. They can be selected and changed.

6.1.2 Parameter menu

```
================== parameter settings ==================
Root sequence:  root_sequence.fa
TREE:     tree.newick
  u scaling factor:   1
--------------------------------------------------------
RATE
  r homogeneity

MODEL
  s substitution model:  WAG
  i insertion rate:   0
  I insertion distribution: -
  d deletion rate:    0
  D deletion distribution: -

OUTPUT:
  o path:           ./
  t true alignment:      false
  h annotate domain architectures: false
  k paint domain architectures:  false
  f separate output files:    false

LINAGE SPECIFIC EVOLUTION:
  l No

================== parameter settings ==================

Quit [q], confirm [y], or change [menu] settings:
```

By typing the characters given at the beginning of each parameter you can change the settings.

TREE

u the branch lengths of the tree will be scaled by this factor.

RATE

r relative rates are assigned to amino acids of non-domain sequence parts. It can be switched between:

- homogeneity
- heterogeneity

```
  r heterogeneity
  a alpha:  1.0
  c categories: No
```

In the case of rate heterogeneity you can assign alpha parameters and categories for a discrete gamma distribution.

MODEL

s substitution model

- WAG
- JTT
- JTT_dcmut
- Dayhoff
- Dayhoff_dcmut
- mtMAM
- mtART
- mtREV
- rtREV
- cpREV
- Vt
- Blosum62
- LG
- HIVb
- HIVw
- user defined
  - `Enter path to substitution model OR (s) for next model:`

i insertion rate

- `enter insertion rate >= 0:`

Per default no insertions are simulated (insertion rate=0).

I insertion distribution

- geometric
- Zipfian
- Zipfian with maximal insertion length

For small parameter values of the Zipfian distribution, we highly recommend to specify a maximal insertion length because of the fat-tailed shape of the distribution.

d deletion rate

- `enter deletion rate >= 0:`

Per default no deletions are simulated (deletion rate=0).

D deletion distribution

- geometric
- Zipfian
- Zipfian with maximal deletion length

For small parameter values of the Zipfian distribution, we highly recommend to specify a maximal deletion length because of the fat-tailed shape of the distribution.

OUTPUT

o You can change the path where all the result files will be stored.

t Per default, no true alignment will be calculated. If a true alignment should be calculated there are three options:

- **leaf**: only leaf node sequences are present in the alignment.
- **ancestral**: leaf node sequences and the root sequence are present in the alignment.
- **all**: leaf node sequences and the root sequence and all inner node sequences are present in the alignment.

h Per default the simulated sequences and the root sequence will not be annotated with protein domains. If you change to `true`, Revolver runs `hmmscan` with each of the sequences against the hmm database you provided. The domain annotation files are stores in the output path/hmmerOutput/. The parameter will be automatically set to `true`, if `paint domain architecture` is set to `true`.

k Per default no image of the domain architectures of the sequences will be created. If you change to `true`, Revoler reads all annotation files from the output path/hmmerOutput/ and paints the domain architectures of each sequence. The image is stored in the file output_path/domainArchitecture.gif.

f Per default all simulated sequences are stored in a multiple fasta file output_path/out.fa. If you cange to `true`, Revolver stores all sequences separately in output_path/fastas/. The parameter will be automatically set to `true`, if `annotate domain architectures` or `paint domain architectures` is set to `true`.

LINAGE SPECIFIC EVOLUTION

l Per default all parameter settings apply to all branches in the tree. If you want to assign different settings to individual branches type `l`. Notice, different parameter settings are stored under the term `model`. You have to assign a name for each model. Afterwards you can define node-model associations. Such an association means, that the simulated evolution of a sequence on the branch leading from the parent node to the node with the given name, will be performed with the parameter settings of the model.

- `Add a node-model association [a] OR modify a node-model [model name]:`
- `Enter a name for the model:` For example `m1`.

```
MODEL m1
  s substitution model: WAG
  i insertion rate:  0
  I insertion distribution: -
  d deletion rate:  0
  D deletion distribution: -

Confirm [y], or change [menu] settings:
```

The substitution model, insertion rate, insertion distribution, deletion rate, and deletion distribution can be defined.

- `Enter node name OR [y] to finish model-node assignments for m1:` Next, you can assign the parameterized model to any node in the tree. The node-model associations are listed as follws:

```
LINAGE SPECIFIC EVOLUTION:
  l Modify parameters for linage specific evolution
  A : m1
  B : m2
  C : m2
```

In this example the simulated evolution of a sequence on the branch leading from the parent node of A to node A are performed under the model m1; from the parent node of B to node B under m2; from the parent node of C to node C also under m2. On all other branches the global settings are applied. By typing `l` the models and the node-model associations can be changed.

y confirms all settings and starts the simulation. All parameter setting will be stored in the file `parameter_input.xml`.

6.2 Parameter File

To start `REvolver` using a xml formatted parameter file move into the directory where `REvolver` is located and type:

```
java -cp "REvolver.jar" revolver parameter_input.xml
```

or use the path to the java file:

```
java -cp "path/to/REvolver.jar" revolver parameter_input.xml
```

We recommend to set all parameters via the interactive terminal as described in Section 6.1. All parameter settings will be stored in the file `parameter_input.xml`. Next time, you can start `REvolver` using the `parameter_input.xml`, without any manual interaction. If you want to write the xml formatted parameter file on your own, please look an the xml schema (`schema.txt`).

7 Examples

All example files are provided in `example.tar.gz`, which you can download from `http://www.cibiv.at/software/revolver/`. The path to the programs `hmmscan` and `hmmfetch` are set to `/usr/local/bin/hmmscan` and `/usr/local/bin/hmmfetch`. You can check whether your `hmmscan` and `hmmfetch` installations are located in the same directory using:

```
which hmmscan
```

and

```
which hmmfetch
```

If this is not the case, please change the paths in the input_parameter.xml files.

7.1 Simulated evolution of a human protein under domain constraints

Here, we simulate the evolution of the human Olfactory receptor 11H6 (O11H6_HUMAN). The sequence is annotated with Pfam using `hmmscan --notextw`. The resulting sequences are stored in `out.fa` and the true alignment is calculated and stored in `trueAlignment.fa`, both in fasta format. The `input_parameter_O11H6_HUMAN.xml` is the parameter_input.xml file. You can run this example as follows:

```
java -cp "REvolver.jar" revolver input_parameter_O11H6_HUMAN.xml
```

if you are in the same directory as `REvolver.jar` is located or:

```
java -cp "path/to/REvolver.jar revolver input_parameter_O11H6_HUMAN.xml
```

otherwise.
The results are stored in results_examlpe1/.

7.2 Simulate proteins of a user-defined domain architecture

In this example, REvolver generates a sequence of a user-defined domain architecture as follows, where Pfam was used as domain database:

```
10 - RLI - 10 - Fer4 - 30 - ABC_tran - 20 - ABC_tran - 30
```

Thus, the sequence consists of an RLI domain (Possible metal-binding domain in RNase L inhibitor; PF04068), an Fer4 domain (4Fe-4S binding domain; PF00037), and two ABC_tran domains (ABC transporter; PF00005) separated by unconstraint segments of different lengths. The evolution of this protein is then simulated with domain constraints. The resulting sequences are automatically annotated with the pHMM database and an image of the domain architectures is generated. The `input_parameter_domain_archi.xml` is the parameter_input.xml file. You can run this example as follows:

```
java -cp "REvolver.jar" revolver input_parameter_domain_archi.xml
```

if you are in the same directory as `REvolver.jar` is located or:

```
java -cp "path/to/REvolver.jar" revolver input_parameter_domain_archi.xml
```

otherwise.
The results are stored in results_examlpe2/.

7.3 Simulation using a custom pHMM

In this example, we simulate the evolution of a G protein-coupled receptor (GPCR) under domain constraints. This time, the pHMM database is a single pHMM (custom.hmm), which we generated based on 29 GPCR sequences using `hmmbuild`. The root sequence is annotated (`hmmscan`) with the custom.hmm. The simulated sequences are also automatically annotated with the custom.hmm. Another option would be to add the custom.hmm to a publicly available phmm database (e.g. Pfam). The `input_parameter_custom_pHMM.xml` is the parameter_input.xml file. You can run this example as follows:

```
java -cp "REvolver.jar" revolver input_parameter_custom_pHMM.xml
```

if you are in the same directory as REvolver.jar is located or:

```
java -cp "path/to/REvolver.jar" revolver input_parameter_custom_pHMM.xml
```

otherwise.
The results are stored in results_examlpe3/.

8 Acknowledgment

Support from the Wiener Wissenschafts-, Forschungs- and Technologiefonds (WWTF) to this work is greatly appreciated. We also acknowledge the funding from the SPP 1174 (Deep Metazoan Phylogeny) project.

References

Abascal F, Posada D, Zardoya R. 2007. MtArt: a new model of amino acid replacement for arthropoda. Molecular Biology and Evolution. 24:1 –5.

Adachi J, Hasegawa M. 1996. Model of amino acid substitution in proteins encoded by mitochondrial DNA. Journal of Molecular Evolution. 42:459–468.

Adachi J, Waddell PJ, Martin W, Hasegawa M. 2000. Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. Journal of Molecular Evolution. 50:348–358.

Dayhoff M, Schwartz R, Orcutt B. 1978. A model of evolutionary change in proteins. Atlas of Protein Sequence and Structure. 5:345–352.

Finn R, Clements J, Eddy SR. 2010. HMMER. http://hmmer.janelia.org/. Version 3.0.

Finn RD, Mistry J, Tate J, et al. (14 co-authors). 2010. The pfam protein families database. Nucleic Acids Research. 38:D211–222.

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences of the United States of America. 89:10915–10919.

Jones DT, Taylor WR, Thornton JM. 1992. The rapid generation of mutation data matrices from protein sequences. CABIOS. 8:275–282.

Kosiol C, Goldman N. 2005. Different versions of the dayhoff rate matrix. Molecular Biology and Evolution. 22:193–199.

Le SQ, Gascuel O. 2008. An improved general amino acid replacement matrix. Molecular Biology and Evolution. 25:1307 –1320.

Letunic I, Doerks T, Bork P. 2009. SMART 6: recent updates and new developments. Nucleic Acids Research. 37:D229–232.

Müller T, Vingron M. 2000. Modeling amino acid replacement. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology. 7:761–776.

Nickle DC, Heath L, Jensen MA, Gilbert PB, Mullins JI, Pond SLK. 2007. HIV-specific probabilistic models of protein evolution. PloS One. 2:e503.

Whelan S, Goldman N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. Molecular Biology and Evolution. 18:691–699.

Yang Z, Nielsen R, Hasegawa M. 1998. Models of amino acid substitution and applications to mitochondrial protein evolution. Molecular Biology and Evolution. 15:1600 –1611.