CoEv software Manual

CoEv  web platform is a platform that allows the user to either predicts coevolving positions and their evolutionary profile based on the aligned sequences and a phylogenetic tree or simulated co-evolving pairs of sites.

# Overview:

The evaluation software is the implementation of the model described in Dib et al. 2014, where we present a new substitution model that describes the coevolving process along the tree and reconstruct the ancestral states of co-evolving pairs of positions. We propose a new probabilistic Markov model that not only identifies coevolving positions but also estimates the associated coevolving profile. The necessity to estimate profiles emerges when considering the nucleotide alphabet that induces 192 different coevolving profiles and amino acid alphabet that induces a bigger number of profiles. The Coev Markov model is based on 16 states instantaneous matrix where each state represented the a combination. The instantaneous rate matrix Q contains 4 continuous parameters and a discrete parameter. The implementation of the model is written in C (Dennis Ritchie, 1969). Given an alignment file, the phylogenetic tree and a pair of positions the user will be able to assess the score of coevolution and estimate the profile with either the maximum likelihood or Bayesian framework.

The simulation software is based on the same instantaneous matrix (Dib et al., 2014). Given a tree in a nexus format the values of the 4 continuous parameters, the software randomly picks a coevolving profile, a state at the root and let this state evolve along the branches according to coev substitution matrix and a frequency vector that favours the coevoling states defined by the profile. The software therefore simulates nucleic and proteic pairs of positions along a tree by assigning a state composed of two letters from the amino acid or nucleotides alphabet to the leafs.

# For local executions:

System requirements: The software CoEv uses several external tools that should be installed. We optimized the implementation using Lapack (Linear Algebra PACKage) and nlopt (library for nonlinear optimization) downloadable at <http://ab-initio.mit.edu/wiki/index.php/NLopt> and tested it on Linux and mac operating systems.

# Instructions for running CoEv with a command line:

To execute run:

./coev [OPTIONS]

-method s: Use method s (either 'bayes', 'ml' or 'sm').

Default is 'ml: maximum likelihood'.

-data s: Use data type s (either 'nt', 'aa' ). Default is 'nt'.

-tree s: The name of the file containing the input tree in Newick

format. Default is s=treeInput.txt.

-align s: When method is bayes or ml, the name of the file containing the sequence alignment in FASTA format. Default is s=alignment.txt.

-cols v1 v2: The columns in the alignment to use for the analysis.

Default is n1=1 and n2=2.

-s v: ??. Default is v=1.

-d v: ??. Default is v=1.

-r1 v: The r1 parameter rates. Default value is 1.

-r2 v: The r2 parameter rates. Default value is 1.

-out s: The name of the log file to write the results to. Default

is s=output.log.

-IT n: When method is bayes, run for n iterations. Default is n=10000.

- ns n : when the method is sm the parameters sets the number of pairs to simulate. Default value is 1.

-sfreq n: When method is bayes, write every n'th iteration to file. Default is n=1000.

-pfreq n: When method is bayes, print every n'th iteration on the screen.

Default is n=1000.

-burnin n: Number of burn-in iterations. Default is n=0.

-h: Print this help screen and exit.

Example to simulate data:

./coev -data nt -method sm -tree example/tree.txt -out example/simulate.out -s 1 -d 10 -r1 5 -r2 5 –ns 10

Example to evaluate score of co-evolution using maximum likelihood framework

./coev -data nt -method sm –align example/align.txt -tree example/tree.txt –out example/outML.log –cols 1 2

Example to evaluate score of co-evolution using Bayesian framework

./coev -data nt -method bayes -align example/align.txt -tree example/tree.txt -out example/outTracer.out -IT 100000 -sfreq 1000 -burnin 1000 –cols 1 2

# Reading output file:

To read the output, the user can either look at the log file or at the output console. In the log file, each line corresponds to a profile.

## In the case of a Maximum Likelihood framework:

Each line can be truncated in 9 values:

Value 1 corresponds to the profile set: it is a binary vector of length 16 where 1 corresponds to a combination that belong to a profile and 0 when it doesn''t. The binary vector follows the following order AA,AC,AG,AT,CA,CC,CG,CT,GA,GC,GG,GT,TA,TC,TG,TT. For example when the value is 1000010000000000 then it means that the profile is {AA,CC}.

Value 2 corresponds to the estimated value of w1 in the Null model: it is a double, ex:6.928542e-02

Value 3 corresponds to the estimated value of w2 in the Null model: it is a double, ex:2.014202e-01

Value 4 corresponds to the estimated value of Log Likelihood in the Null model: it is a doube, ex:-7.405995e+01

Value 5 corresponds to the estimated value of s in the model: it is a double, ex: 2.735160e-02

Value 6 corresponds to the estimated value of d in the Coev model: it is a double, ex: -4.970079e-01

Value 7 corresponds to the estimated value of r1 in the Coev model: it is a double, ex: 3.103843e-02

Value 8 corresponds to the estimated value of r2 in the Coev model: it is a double, ex: 2.922180e-01

Value 9 corresponds to the estimated value of Log Likelihood in the of the Coev model: it is a double, ex:-6.814893e+01

## In the case of a Bayesian framework:

The log file is conceived to be read by tracer. The first line is a header. The other lines contains values where

Value 1 corresponds to the iteration identifier: it is an integer

Value 2 corresponds to the posterior probability: it is a double

Value 3 corresponds to the LogLikelihood value of Coev: it is a double

Value 4 corresponds to the prior value:  it is a double

Value 5 corresponds to s: it is a double

Value 6 corresponds to d: it is a double

Value 7 corresponds to r1: it is a double

Value 8 corresponds to the r2: it is a double

The rest of the values correspond to the  profile vector  where 1 corresponds to a combination that belong to a profile and 0 when it doesn''t. The binary vector follows the following order AA,AC,AG,AT,CA,CC,CG,CT,GA,GC,GG,GT,TA,TC,TG,TT. For example when the value is 1000010000000000 then it means that the profile is {AA,CC}.

## In the case of a simulation:

The software output a fasta format file.

# Contacts:

For questions, please send an email to www.phylo [at] unil.ch

# Reference:

When using this software, please cite:

L. Dib, D. Silvestro and N. Salamin (2014). Evolutionary footprint of coevolving positions in genes. *Bioinformatics*. 30(9): 1241-9.

L. Dib et al. CoSim , in review.