LnLCorr Manual

Zhengyuan Wang and David Pollock, September 26, 2003

Introduction

The *LnLCorr* package is designed to detect pairwise coevolutionary among residues in a set of proteins using likelihood ratio tests. The basic program was described in Pollock, Taylor, and Goldman (1999), and only minor modifications have been made since. In brief, residues at each site are clustered into two groups, and a comparison is made between an independent model and a model in which there is a simple dependency between the sites in their evolutionary process. The clustering can be performed however the user sees fit, but one obvious example is to use vectors that describe the physicochemical properties of the residues, such as charge, hydrophobicity, or size. If this is done, the extra parameter in the dependent model, "residue disequilibrium (RD)", determines the sign (positive or negative) of coevolution. These two directions of coevolution may have different properties, particularly with the charge vector.

The methodology uses a phylogenetic tree, with branch lengths, as input, since phylogenetic relationships will create correlation among residues that acts as noise to drown out coevolutionary signal (Pollock and Taylor, 1997). In this or any other coevolution/covariation analysis, it is highly recommended that a dense phylogeny be used such that the number of long branches is minimized, and the number of taxa is at least twenty (although 40-100+ is better). The proteins should be pre-aligned, and any sites with gaps will be ignored. Taxa that are in the phylogeny or alignment, but not both, will be deleted; output trees and alignments can be viewed to see what data was actually utilized. The statistical evaluation is based on the likelihood maxima of the sequence data given each of the two models and the phylogenetic tree. The distribution of this statistic cannot reliably be assumed to be chi-square (Pollock, Taylor, and Goldman, 1999), so it is necessary to perform parametric bootstrapping based on the maximum likelihood

values of the independent model to obtain significance levels. There is a huge multiple comparisons issue here, and individual comparisons are unlikely to achieve significance after e.g., Bonferroni corrections. It is therefore recommended to pre-specify significance levels of interest (e.g., 0.05, 0.01), and then consider the posterior probability of the number of sites with larger likelihood ratio statistics.

The package includes three programs: *LnLCorr*, *CorrStrap*, and *CoevPairs*. There is also a "control" file and a folder with sample files. *LnLCorr*, the central evaluation program, and *CorrStrap*, the parametric bootstrapping program, are coded in C/C++ and compiled for the Linux operating system. *LnLCorr* analyzes pairwise likelihood ratios through maximum likelihood estimation. *CorrStrap* simulates sequences based on the parameters estimated with the independent model (parametric bootstrapping). *CoevPairs* is a Perl script that extracts pairs showing significant coevolution from the output of the first two programs.

The procedure to use this package is to first analyze the sequence data, producing an output file "corrs.out". Parametric bootstrapping is then done to create simulated sequences, and *LnLCorr* is re-run to obtain a "corrs.out" file for the simulated independent sequences. The two output files are then compared using *CoevPairs* to get the final result. Further analysis, such as correlating interactions with features of function and three-dimensional structures, can be carried out with the user's favorite programs.

How to use LnLCorr

Five input files are needed (control file, sequence alignment, phylogenetic tree, pdb structural information [if available], and vector of amino acid values), and they should be put in the same directory as the programs. (Note: versions of LnLCorr and CorrStrap that do not require pdb files have been added to the web site). An example control file ("control_file") is included, and a set of examples for the other four files using Cytochrome Oxidase I (COI) are included in the "SampleFiles" folder. The amino acid sequences should be aligned and in pir format[1]. The tree file for the phylogeny should be

in the parenthetical Newick format, and should only include bifurcations (check the output tree to see that the input tree was read properly). Protein structure information should be in the pdb format (PDB is notoriously variable, and the structure-related output should be checked to make sure that there were no problems reading this file). The file for state segregation (split definition) should be named "splitdef" and contain two lines: the second line provides the quantitative values corresponding to the amino acid residues in the first line, and all names and values are separated by spaces. The names for the phylogeny, sequence alignment, and pdb files (e.g., intree, seqs.pir, struc3d) are given in the first line of the control file, and all names are case sensitive; later in the manual we will use these example names to refer to these files. Other parameters in the control file are for programming purposes and should not be modified without more detailed instructions, although the number on the third line is the initial seed for the random number generator. For examples of the formats of these files please check the sample files in the "Samplefiles" folder.

If the files are in place[2], type "*LnLCorr*" to run the program. Output will include "corrs.out", "siteL.out", siteL.crypt", "seq.out", "tree.out" and "struc3d.out". The "corrs.out" file includes, for each paired comparison, the maximum likelihood estimated parameters (of state frequences and rates) and pairwise structure information together with the likelihood ratio values. This file is the input file for the *CoevPairs* program. The other output files are needed for subsequent analysis and should not be modified. The seq.out, tree.out, and struc3d.out should be reviewed to check what sequence, tree, and structure information were actually used in the program, and to check for errors. After each run, "corrs.out" should be copied and renamed so that it is not overwritten in subsequent runs. All output files should be removed from the running folder before subsequent runs with different split definitions or different sequence, tree, or structure information.

For more details about the algorithm and implementation of *LnLCorr*, refer to the documentation of *LnLCorr*.

How to use CorrStrap

CorrStrap uses “SiteL.out”, “SiteL.crypt” and all th input files of LnLCorr as its input files, and these files should be in the same directory as *CorrStrap*. After running *CorrStrap*, output files are “simseqs.out” and “seq.out”. The “seq.out” file includes the parametric bootstrapped sequences in pir format and will be used as input sequences for the next round of *LnLCorr* to get the null likelihood ratio distribution (the likelihood ratio distribution without coevolution). This file has to be renamed “seqs.pir” before use as the input sequence alignment file for *LnLCorr*.

How to used CoevPairs

Before using *CoevPairs* the user should have installed Perl and the Perl plot package, “Perl-Plot”. *CoevPairs* takes two files as its input files, “coevIn” and “simuIn”. These two files should be in the same directory and are renamed versions of the “corrs.out” files respectively from the original run and from the second run evaluating the bootstrapped sequences. The first output file, “CoevA”, includes the coevolving pairs. A picture file “LnLDistri” shows the cumulative distribution of the likelihood ratios of the pairs in both “coevIn” and “simuIn”. The black line is the null distribution, while the blue, red, and green lines are the total and separate positive and negative coevolved pairs, respectively. The “DistanDistri” file shows the three-dimensional distance distribution for pairs in “coevIn”. The black line is all pairs, the blue line is the coevolving pairs, and the red line and green line are the positive coevolving pairs and negative coevolving pairs, respectively.

[How do you control the cutoff?] [How do you filter sites based on their separation along the sequence?]

Notes

[1]PIR format has “>P1;[sequence_name]” on the first line, followed by description on the second line, followed by the sequence on subsequent lines. Although this program was

designed for amino acid sequences, it can be modified for nucleotide sequences.

[2]The maximum number sequences is currently set at 300 and the maximum sequence length set at 1000. These can be modified upon request.

[3]A known bug is that the aligned sequence corresponding to the pdb file sequence should not have any gaps, or the distances between sites might not be correct. Users should remove the gaps involved in this sequence in their multiple alignment until this is fixed.

Requests, questions, or bug reports should be directed to Zhengyuan Wang zwang3@lsu.edu or David Pollock (dpollock@lsu.edu).

Future goals

We have a variety of programs and Perl scripts to view the data in different ways, such as selecting out sites based on secondary structure or surface accessibility. These should be incorporated into future packages. We would also like to streamline the control files and include more detailed description of controls on running the programs. Pairwise visualization of probable changes along the tree was previously implemented for the Sun program "TreeView", and we would like to resurrect such analyses for more widely available programs, or automatically generate graphics for display. Future versions may include slightly more complex models, but we are working on alternative methodologies to incorporate a wider variety of complex models (within the limits of data and avoiding overparameterization) and tree variation.