MAYROSE LAB

ChromEvol version 2.0 Manual

March 2014

ChromEvol is a likelihood-based method for tracking the pattern of chromosome number change along a phylogeny. Given a phylogeny and chromosome number data, the program infers ancestral chromosome numbers as well as the location and type of chromosome number transitions along the tree.

1 Introduction

Chromosome number is a remarkably dynamic feature of eukaryotic evolution. Chromosome numbers can change by a duplication of the whole genome (a process termed polyploidy), or by single chromosome changes (ascending dysploidy via, e.g., chromosome fission or descending dysploidy via, e.g., chromosome fusion). Of the various mechanisms of chromosome number change, polyploidy has received significant attention because of the impact such an event may have on the organism and its subsequent evolution.

ChromEvol implements a series of likelihood models depicting the evolution of chromosome numbers. By comparing the fit of the different models to biological data, it may be possible to gain insight regarding the pathways by which the evolution of chromosome number proceeds. For each model, the program estimates the rates for the possible transitions assumed by the model, infers the set of ancestral chromosome numbers, and estimates the location along the tree for which polyploidy events (and other chromosome number changes) occurred.

2 Download and compilation

Source code (C++) and compiled executables can be downloaded from
http://www.tau.ac.il/~itaymay/cp/chromEvol/index.html

To compile the program on your local Unix\Linux machine, follow the steps below:

1. Use the tar command with the –xzvf flag to extract the files. In the command prompt type:
   ```
   tar -xzvf chromEvoltar.gz
   ```
2. Browse (cd) to the extracted directory
3. In most systems you can use the provided Makefile to compile the program and to create the chromEvol executable. Simply type 'make' in the command prompt. In some operating systems, you may have to use the following command:
   ```
   g++ -o chromEvol.exe -O3 *.cpp –DDOUBLEREP
   ```

3 Running ChromEvol

ChromEvol is run using a plain text control file. To run the program type the path to your chromEvol executable, followed by the path to the control file. The control file specifies the input/output files and the various options to be used. The obligatory inputs to chromEvol are a tree file in Newick format and a chromosome counts data file (see section 3.1.1 for description of the correct format). The user is responsible for a correspondence between the two files so that all extant taxa in the tree have chromosome count data, and that all taxa with count data appear in the tree.

The purpose of the first phase of the analysis is to estimate the optimum parameter values of the underlying model. This is done with standard maximum likelihood techniques. Then, the most likely ancestral chromosome numbers are inferred, as well as the probability of any given chromosome number to exist at any internal node. Finally, the program estimates the expected number of polyploidy and dysploidy transitions along each branch of the phylogeny. The last step is done using simulations, which may be computationally intensive. For this stage, the interplay between accuracy and running time can be controlled using the control file.

Running examples on a sample dataset and examples of chromEvol inputs, are included in the chromEvol package, available from the software webpage.

3.1 Inputs

3.1.1 Chromosome counts file

Chromosome counts data should be supplied in a format that is similar to FASTA sequence file. For each species in the input tree two lines should be specified. The first line lists the species name, preceded by the symbol '>'. The species name must be identical to the name as appear in the input tree file. The second line specified the chromosome number for that species. The numbers should represent the **haploid** chromosome number (n).

Extensions:

1. If the chromosome numer for a certain species is unknown, the symbol 'X' can be used. The program then treats this species as missing data (similar to a gap in molecular sequence data).
2. The proprogram also accepts multiple chromosome counts for a certain species. This represents cases when several alternative counts were sampled for that species. Each alternative count should be followed by its frequency (i.e. probability), and the sum of all possible frequencies must be equal to 1.0. For example if 40% of the sampled haploid chromosome counts for TAXA_A were 24 and 60% were 25 then these two counts should be written in the data file as follows:
   >TAXA_A
   24=0.4_25=0.6

3.1.2 Tree file

A text file containing a tree in Newick format. The tree should have branch lengths information, but should not have bootstrap or other node id information. It must be rooted, but may or may not be ultrametric. The user has to make sure the taxa names in the tree file match those in the chromosome counts file.

3.1.3 Control file

The control file used when running the program is a plain text file. Each of the following parameters and options should appear in a separate line. Parameter names are always preceded by an '_' (underscore) symbol.

Parameter	**Description**	**Default**
_mainType	Possible options: Run_Fix_Param = Run analysis with the specified parameters values. Optimize_Model = Optimize the specified model parameters and then run analysis. All_Models = Run analysis for each of the 10 supported models (see section 3.2). mainSimulate = simulate chromosome number evolution	Optimize_Model

	along the tree (see section 3.6).	
_dataFile	A path to a file with the chromosome count data	Obligatory
_treeFile	A path to a tree file in Newick format	Obligatory
_outDir	A path to the location of the output directory.	RESULTS
_maxChrNum	The maximal chromosome number allowed. Negative values (-X): Set the maximal chromosome number allowed to be X units larger than the maximal chromosome number observed in the data file.	-10
_minChrNum	The minimal chromosome number allowed. Negative values (-X): Set the minimal chromosome number allowed to be X units smaller than the minimal chromosome number observed in the data file.	1
_simulationsNum	The number of simulations for computing the expectation of the number of changes of certain transition type along each branch. Note: This step is computationally extensive. Lower values results in faster computations with decreased accuracy.	10000
_logFile	A path to a log file, describing the process of the program.	log.txt
_branchMul	If different than 1.0 then all branch lengths of the tree are multiplied by this scalar. Should be used if one of the model parameters are close to their boundary value (100), or in order to scale the tree when the branch lengths are exceptionally large or small.	999 (=Scale tree so that total tree length is equal to the number of different character types)
_optimizePointsNum _optimizeIterNum	These two parameters should always be given together. They define the way start-points are chosen for optimization. _optimizePointsNum defines the number of points examined at each step. The log likelihood of each point is calculated, and the best points are used in next steps. The value for this parameter should be a list of comma-separated numbers. _optimizeIterNum defines the number of optimization iterations performed at each step (on each of the points generated by _optimizePointsNum). For example, if _optimizePointsNum 10,3,1 and _optimizeIterNum 0,2,5 are set, then 10 random points are picked, logL is calculated (without doing any optimization), the 3 best points are picked, and 2 optimization iterations are performed. The best point is then picked and up to 5 iterations are performed (may perform less than 5 if there is no improvement in logL).	10,3,1 0,2,5
_freqFile	Root frequency file (see description in section 3.5)	None

In addition, the user may specify model parameter values, and the way they are optimized. Currently the program supports 5 types of transitions between different chromosome numbers:

- **Gain** – ascending dysploidy (e.g from n=10 to n=11)
- **Loss** – descending dysploidy (e.g from n=12 to n=11)
- **Duplication** (or polyploidization) – whole genome duplication, where the number of chromosomes is doubled (e.g from n=10 to n=20)
- **Demi-duplication** – a multiplication of the chromosome number by a factor of 1.5 (e.g from n=12 to n=18)
- **Base-number addition** – given a base-number x, this is an addition of a multiple of x to the number of chromosomes (e.g from n=18 to n=27,36,45,…, for a base-number x=9)

The user may include all parameters in the model or choose to ignore some of them. By specifying different sets of parameters the user may compare different hypotheses regarding the evolution of chromosome number along a given phylogeny. The model parameters are specified in the control file. In order to include a parameter, its name should be followed by a space and a positive number. In case the Optimize_Model option is specified, this number represents the initial parameter value for optimization. If Run_Fix_Param is specified, then the parameter is fixed to that value. In order to exclude a parameter, the parameter name should be omitted from the control file or be followed by the number -999.

Parameter	**Description**
_gainConstR	Rate for single chromosome increases
_gainLinearR	Rates for single chromosome increases are dependent on the current chromosome number
_lossConstR	Rate for single chromosome decreases
_lossLinearR	Rates for single chromosome decreases are dependent on the current chromosome number
_duplConstR	Rate of whole genome duplications (polyploidy)
_demiPloidyR	Rate for demi-polyploidizations. If -2 is specified then the rate of demi-polyploidy is equal to that of polyploidy. Thus, the number of model parameters does not increase

_baseNumber	A specified chromosome number which characterizes a phylogenetic group. This is NOT the chromosome number at the root of the phylogeny.
_baseNumberR	Rate for transitions by base-number (see above), given the base-number of the phylogeny (_baseNumber parameter)
_bOptBaseNumber	Defines whether the base-number is optimized by the program or not. Set this parameter to 1 in order to optimize, and to 0 in order to keep base-number fixed to the value given by _baseNumber (recommended)

3.2 Default chromosome number evolutionary models

When the _mainType All_Models option is used, 10 models of chromosome number evolution are evaluated. Each of these models is characterized by a different set of parameters, as described in the table below.

The maximal log-likelihood values and AIC scores of each model are printed to the output file modelsSummary.txt. These can be used for selecting the best-fitting model for the given dataset. The user may also choose to evaluate a specific model, by setting the _mainType parameter to Optimize_Model, and specifying the desired model parameters.

Model name	**Model parameters**
CONST_RATE	_gainConstR, _lossConstR, _duplConstR
CONST_RATE_DEMI	_gainConstR, _lossConstR, _duplConstR = _demiPloidyR
CONST_RATE_DEMI_EST	_gainConstR, _lossConstR, _duplConstR, _demiPloidyR
CONST_RATE_NO_DUPL	_gainConstR, _lossConstR
LINEAR_RATE	_gainConstR, _gainLinearR, _lossConstR, _lossLinearR, _duplConstR
LINEAR_RATE_DEMI	_gainConstR, _gainLinearR, _lossConstR, _lossLinearR, _duplConstR = _demiPloidyR
LINEAR_RATE_DEMI_EST	_gainConstR, _gainLinearR, _lossConstR, _lossLinearR, _duplConstR, _demiPloidyR
LINEAR_RATE_NO_DUPL	_gainConstR, _gainLinearR, _lossConstR, _lossLinearR
BASE_NUM	_gainConstR, _lossConstR, , _baseNumberR, _baseNumber
BASE_NUM_DUPL	_gainConstR, _lossConstR, _duplConstR, _baseNumberR, _baseNumber

3.3 Usage notes

- The input phylogenetic tree is assumed to be rooted. The root name is printed in the log file and in the chromEvol.res_output file. In order to verify that this is the correct root node, the user should view the file allNodes.tree.

- The branch lengths of the tree represent expected number of chromosome-number transitions along a branch. Usually, however, branch lengths are given in units of time or in average number of nucleotide (or Amino acids) substitutions. Thus, in order to convert the input tree into units of chromosome number transitions, it is scaled (all branch lengths are multiplied by a scalar) so that the total tree length will represent a realistic number of chromosome-number transitions. Use the _branchMul parameter in the control file to scale the tree by a specified scalar or to keep the branch lengths identical to the input tree (_branchMul = 1.0). By default (_branchMul = 999), the tree is scaled so that the total tree length is equal to the number of unique chromosome counts. This scaling usually (but not always) results in adequate branch lengths.
- For efficient multi-dimensional optimization, the program sets an upper bound of 100.0 for the rate parameters. However, in case one of the optimized model parameters is close to this upper bound it is indicative that the model parameters may not have reached their global optimum. The solution is to multiply all branch lengths by a certain factor (e.g., 10) and run the program again. Multiplying all branch lengths by a scalar can be done using the _branchMul option in the control file.

3.4 Outputs

Each ChromEvol run produces the following 8 files in the specified output directory (_outDir parameter):

- **chromEvol.res** - This file includes various run statistics as well as the inferred model parameters, frequencies of the chromosome numbers at the root of the tree, and the log-likelihood value of the optimized parameter set.
- **log.txt** - All outputs that are printed during an execution of the program.
- **allNodes.tree** - A tree file in Newick format that specifies the names for nodes (internal and leaf) of the input tree. Internal node's names are given as the

bootstrap values and can be viewed in tree viewing programs such as njplot or FigTree.

- **mlAncestors.tree** - A Newick tree file with the ancestral chromosome number reconstruction, as obtained using maximum-likelihood. The reconstruction of ancestral states is printed as the bootstrap values and can be viewed using a tree viewing program.
- **posteriorAncestors.tree** - A Newick tree file with the posterior probabilities of the two most probable chromosome numbers at each internal node. These probabilities are printed as the bootstrap values and can be viewed using a tree viewing program.
- **exp.tree** - A Newick tree file with the expected number of chromosome number transitions events that are inferred to occur along each branch. The numbers are given as : gains//loses//polyploidy//demi-polyploidy. These expectations are printed instead of the bootstrap values for the terminal node below the branch (further from the root) and can be viewed using a tree viewing program.
- **ancestorsProbs.txt** - A table with the posterior probability of each chromosome number to occur at each internal node.
- **expectations.txt** - Lists the expected number of transitions along each branch. Branches with an expectation above 0.5 of any transition type are listed. A table at the end of the file lists the expected number of events for all branches of the phylogeny. The name of the branch is given as the name of the node bellow it (further from the root).

When the _mainType option All_Models is specified, an additional output file is created. The **modelsSummary.txt** file lists the log-likelihood and AIC scores of all models. This should be used to choose the model that best fits a particular dataset.

3.5 Testing hypotheses about the root chromosome number

By default, chromEvol calculates the likelihood function by summing over all possible chromosome numbers at the root node. However, it is possible to run the analysis with a fixed chromosome number at the root. This can be useful when trying to test hypotheses regarding this number (sometimes referred to as the base number, although note that in chromEvol the base number is defined as the monoploid number, and may or may not be equal to the root state). For example, one may run ChromEvol twice, with different fixed root chromosome number, and then compare the likelihoods of the results.

To use this option, add the _rootFreqType FIXED parameter to the control file. You'll also need to specify root frequencies in an additional file, and add the _freqFile parameter with the path to your file. For example: _freqFile rootfreq.txt

To set a single fixed root chromosome number, the frequency file should include a single line:

F[X] = 1

where X represents your chromosome number of choice.

It is also possible to specify several root states, each with its corresponding probability. For example, the following lines in the root frequency file:

F[4] = 0.7

F[9] = 0.2

F[12] = 0.1

mean that the probability of the root taxon having 4 chromosomes is 0.7, 0.2 for it to be 9 and 0.1 to be 12.

3.6 Simulating chromosome number evolution

chromEvol is capable of creating simulated data, based on a given set of model parameters. Simulated data can then be used to test the inference robustness. This for example, is done in the ploidy inference pipeline (see section 4).

To run a simulation, set the _mainType parameter to mainSimulate. A tree input is required, given through the _treeFile parameter. You'll also need to set the following parameters:

Parameter	**Description**
_simulationsIter	The number of simulation iterations to perform for the given set of parameters

_maxChrNumForSimulations	The maximum chromosome number allowed to be reached during a simulation. Any simulation reaching this number will be marked as so in the corresponding log file.
_freqFile	Root frequency file (see section 3.5)

Use the rate parameters described in section 3.1.3 to specify transitions types and rates. When simulating a dataset including transitions by base number, it must be specified using the _baseNumber parameter. In addition, the parameter _baseTransitionProbs should be included. This parameter lists the possible base number transitions, with their corresponding probabilities. For example, for a base number x=9, the parameter value could be 9=0.5_18=0.25_27=0.25 , meaning that the probability for addition of 9 to the chromosome number is 0.5, while additions of 18 and 27 have a probability of 0.25. Make sure probabilities sum to 1.

See the example simulation control file in the chromEvol package.

Simulation output files are created in the output directory and include the following:

- **allNodes.tree** - A tree file in Newick format that specifies the names for nodes (internal and external) of the input tree. Internal node's names are given as the bootstrap values and can be viewed in tree viewing programs such as njplot or FigTree.
- **simEvents.txt -** A text file detailing the simulated events in each of the tree's internal and external nodes**.**
- **simCounts.txt -** A FASTA-format file with the simulated chromosome counts for each of the tree external nodes**.**
- **simCountsAllNodes.txt -** A FASTA-format file with the simulated chromosome counts for each of the tree internal and external nodes**.**
- **simTree.phr -** A tree file in Newick format that specifies the names and simulated chromosome numbers for nodes (internal and external) of the input tree. Names and simulated data for internal node are stored as bootstrap values, while for external nodes the simulated data are given as part of the leaf name.

4 The ploidy inference pipeline

In addition to the core chromEvol release, we supply a pipeline that will enable to categorize extant taxa as diploids or polyploids, relative to the ancestral chromosome number of the group examined. By doing so, the common ancestor of the group (i.e., the root of the phylogeny) is treated as diploid.

The pipeline is an external Perl program that runs chromEvol multiple times in order to infer the ploidy level of a set of taxa. The input is a sample of species-trees (e.g., inferred using MrBayes) and the chromosome numbers for these species. The output is the ploidy level of each species and, in case some species could not be reliably inferred according to the procedure described below, a taxon is marked as "NA" (not available). The procedure follows these general steps:

1. Find the optimal chromosome-number evolutionary model for the given dataset using a single tree (in our analyses we used the maximum a-posteriori probability (MAP) tree). This model is used in all subsequent steps. The main aim of this step is to reduce computation time so that not all available models are run in each step. The optimal model is selected by means of the Akaike Information Criterion (AIC).
2. In case multiple trees are available, randomly choose a pre-specified number of trees (e.g., 100) to be used in subsequent steps.
3. The parameters of the model chosen in step 1 are optimized independently for each tree sampled in step 2.
4. Simulate chromosome-number evolution along the input tree(s) using the corresponding model parameters as inferred in step 3. By default, 100 simulations are performed. The simulated chromosome numbers at the tips are then used as data input to chromEvol; thereby comparing "true" (simulated) and inferred ploidy transitions. To make the inference step as realistic as possible, simulated chromosome numbers at the tips are retained only for those species with available chromosome number in the original input data and are converted to 'unknown' for species with missing data. There are two purposes for this simulation step. The first is to detect two thresholds required for inferring ploidy levels: a species is inferred

as polyploid (diploid) if the expected number of polyploidization events from the root to the tip is above (below) the polyploid (diploid) threshold and as NA if this expectation is in between these two thresholds. The optimal thresholds are determined using the Matthews Correlation Coefficient (Matthews 1975) that balances between true/false positives and negatives. The second purpose for using simulations is to detect taxa for which ploidy level could not be reliably inferred according to the underlying model. These are the taxa that suffer from high false positive/negative rates (e.g., a taxon was inferred as diploid while it was simulated as polyploid). A taxon is marked as unreliably-inferred if its ploidy level according to the optimized threshold was erroneously inferred in more than 5% of the simulations. Usually, these taxa reside in a subtree with a large fraction of species with missing chromosome counts, so that the inferences are based mainly on the evolutionary model and less on the observed data. These taxa are marked as NA.

5. In case multiple trees are given, infer ploidy levels across a sample of trees. This step is based on the thresholds found in step 4. Inferring from a sample of trees rather than from a single tree increases the robustness of inference by accounting for phylogenetic uncertainties. For each species, inferences obtained across the trees are compared and those species which do not exhibit the same inference across 95% or more of the topologies are marked as NA.
6. Combine the reliability estimates from steps 4 and 5 and summarize ploidy levels. When multiple trees are given, summary statistics of the optimized model parameters obtained across trees (median and 95% interval) further allows users to evaluate the consistency of the optimization search, thereby possibly locating multiple optima.

4.1 Requirements and inputs

In order to run the ploidy inference pipeline, you will need a working chromEvol executable. The pipeline can only be run on Unix/Linux-based machines. You will also need Perl v5.8 or higher, with the following modules installed: BioPerl (http://search.cpan.org/~cjfields/BioPerl-1.6.922/) and Parallel::ForkManager (http://search.cpan.org/~dlux/Parallel-ForkManager-0.7.5/ForkManager.pm). It should be noted that the inference procedure is computationally intensive, and therefore it is advised to run the analysis on a strong, multiprocessor machine.

Inputs are similar to those described above, and include:

1. A single-tree file, usually the consensus or the maximum a-posteriori probability (MAP) tree. Fron our experience, the MAP tree is preferred over the consensus. All format specifications detailed for the tree input of chromEvol apply here too.
2. A file containing multiple tree topologies, usually the trees sampled in a Bayesian analysis. All trees should be in Newick format, have branch lengths information, and should not have bootstrap or other node id information. The trees **must be rooted** but may or may not be ultrametric.
3. Chromosome counts file – as described in section 3.1.1.

4.2 Usage

The ploidy inference pipeline is used in a similar way to that used for the core chromEvol program. Parameters are passed to the script using a control file. Each line of the control file should include one of the parameters specified below, followed by a space and the required parameter value. Optional parameters can be omitted from the control file, and will receive their default value. To run the pipeline, type in the command line perl PIP.pl PIP_control

Parameter	**Description**	**Type**
_guideTreeFile	A Newick format file containing a single tree, which will be used for model selection. Usually the consensus or MAP tree is used.	Mandatory
_dataFile	Chromosome counts file.	Mandatory
_treesFile	A Newick format file containing multiple tree topologies.Default is running without multiple topologies.	Optional
_topologyNum	The number of different tree topologies (taken from the input trees) to be tested. Default is 100.	Optional
_simNum	The number of simulations to be performed to assess reliability. Default is 100.	Optional
_runModels	Specifies which models will be tested at the model-choice step. Use “ALL” to run all 10 models, or give a list of selected models, separated by spaces	Optional

	(see list of models in section 3.2). Default is "ALL".	
_baseNum	The base chromosome number for the dataset. It is highly recommended to specify base number, but if unknown, the default is for the program to optimize it.	Optional
_ploidyCallType	Determines which events are considered as contributing to polyploidy. Default is duplication event (DUPL), demi-duplication events (DEMI) and increase in base number (BASE). Use this parameter to specify otherwise, by following it with the types of choice, separated by spces. For example, '_ploidyCallType DUPL DEMI' will only consider duplication and demi-duplication as contributing to ploidy increase.	Optional
_paramTemplates	A path to the directory containing parameter template files (Included in the downloaded package)	Mandatory
_outDir	Output directory for the pipeline outputs.	Mandatory
_chromevolExe	A path to the chromEvol executable in use.	Mandatory
_cpusNum	The maximum number of CPUs that should be occupied by the program. You can specify a number, or use all available CPUs by passing the value 0 (default). Pass 1 to avoid multiprocessing.	Optional

4.3 Output

During the analysis, many files are generated, including chromEvol run parameters, inference results and simulated data. However, the most important outputs are:

- **ploidy.txt** – the final output of the ploidy inference pipeline. This file contains all taxa found in the input trees, with their inferred ploidy, which can be either 0 (diploid), 1 (polyploidy) or NA, in case ploidy inference did not yield a reliable result. The content of this file is based upon reliability calculations, also listed in the file. Two reliability scores (between 0 and 1) are given for each taxon, based on simulations and inference from different phylogenies. In case one of these scores is lower than 0.95, the inference is considered unreliable.
- **param_distribution** – includes medians and 90% value intervals for rate parameters of the best-fitted model, based on the inference from different phylogenies.
- **mlAncestors.tree/posteriorAncestors.tree –** ancestral reconstruction of chromosome numbers, obtained by maximum likelihood/posterior probabilities respectively, on the consensus or MAP tree. Reconstructions are based upon the best-fitted model. See section 3.4 for details.