ALTree: Association and Localisation tests using haplotype phylogenetic Trees

Claire Bardel, Vincent Danjean, Pierre Darlu and Emmanuelle Génin

Version 1.2

Contents

Chapter 1

Introduction

1.1 What's new?

Problem with PAUP* It seems that PAUP* does not run anymore on recent versions of Linux. We have not tested windows or MacOS version. If PAUP* does not work on your system, example files using PAUP* (in the paup directory) will not run because PAUP* output files will not be produced.

Version 1.1.0: modification of `ALTree` and `ALTree-add-S` to deal with quantitative data The software now deals with quantitative data. For the association test, series of one-way ANOVA are performed instead of the homogeneity tests. For the localisation test, only the definition of the S character is different for quantitative traits. Currently, `ALTree-convert` has not been modified and does not deal with quantitative data.

1.2 Overview of the software

This software is designed to perform phylogeny-based analysis: first, it allows the detection of an association between a candidate gene and a disease or a quantitative trait, and second, it enables us to make hypothesis about the susceptibility loci.

It contains three programs: `ALTree`, `ALTree-convert` and `ALTree-add-S`. The connections between these programs are described in Figure I

1.2.1 `ALTree`

Association test

The test consists in performing series of nested tests at different level of a phylogenetic tree. These tests compare either the number of cases and controls (for case/control data) in the different groups (or clades) defined on the tree or the variance of the trait within each clade to the variance between these different clades (for quantitative data). The nested algorithm is detailed on Figure II (figure from Bardel et al. [2005], slightly modified). Then, a global p-value is calculated for the tree by using a permutation procedure such as the one described by Ge et al. [2003] and Becker and Knapp [2004].

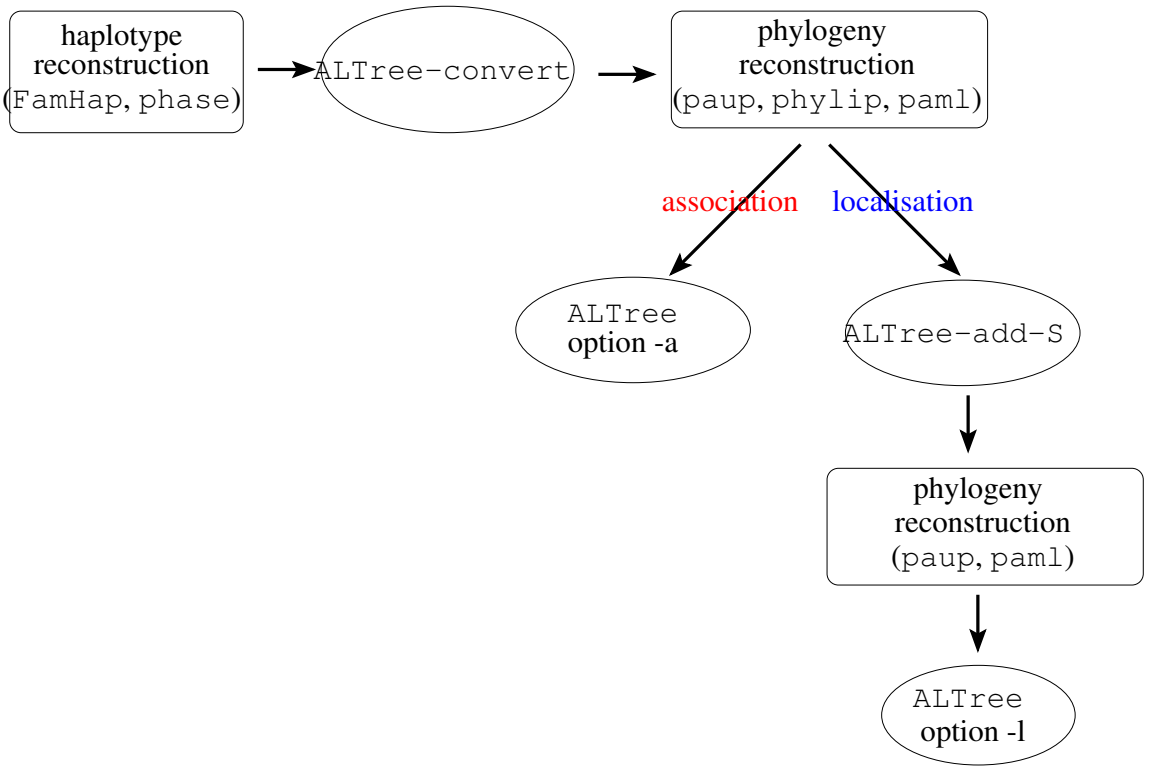


Figure I: ALTree programs

Localisation of the susceptibility loci

To perform the localisation analysis, for each haplotype h, the user must previously define a new character (called character S) whose state depends on the proportion of cases (resp. individual with high quantitative trait values) carrying haplotype h and optimise it on the haplotype phylogeny. The program ALTree then looks for sites that co-mutates with the character S by calculating a co-mutation index called V_i for each site i and for each character state transition ($0 \rightarrow 1$ for example). The higher the V_i, the higher the probability of i being the susceptibility site.

The method implemented in ALTree has been fully described in Bardel et al. [2005]. Please refer to this article for a more complete description.

1.2.2 ALTree-convert

Before running ALTree, you will generally have to reconstruct haplotypes The output of the haplotype reconstruction programs are totally different from the input files necessary for the phylogenetic reconstruction programs. ALTree-convert was then written to convert the outputs of haplotype reconstruction programs to input files for phylogenetic reconstruction programs. It may be particularly useful if you want to use paup because paup has a very high number of options. If you use ALTree-convert, an input file with all the options necessary to further run ALTree is produced.

Currently, ALTree-convert can deal with two haplotype reconstruction programs: phase [Stephens et al., 2001; Stephens and Donnelly, 2003] and FamHap [Becker and Knapp, 2004] and can produce files for three phylogeny reconstruction programs: paup [Swofford, 2002], phylip [Felsenstein, 2004] and paml [Yang].

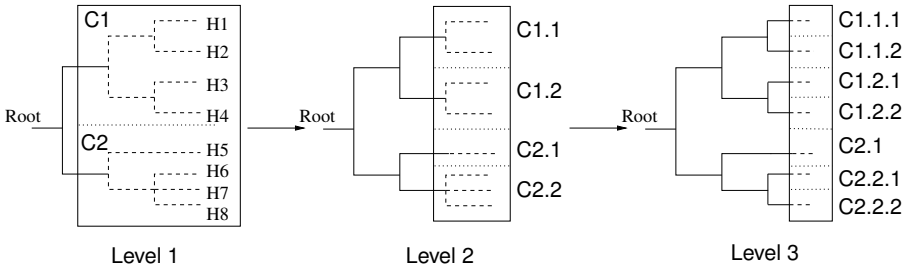


Figure II: Description of the nested clade analysis (without the permutation procedure)
(A) shows the homogeneity test or the ANOVA performed at level k (between clades C_1 and C_2). Then (B), a test will be performed at the following level (k+1), between all the sub-clades descending from clades C_1 and C_2, i.e between clades $C_{1.1}$, $C_{1.2}$, $C_{2.1}$ and $C_{2.2}$.

1.2.3 `ALTree-add-S`

To perform the localisation analysis, a new character S must be added to each haplotype h. The state of S depends on the proportion of cases (resp. individual with high quantitative trait values) carrying the haplotype h. You can use your own criterion to determine the state of S and add it manually to the input file of the phylogeny reconstruction program that will optimise the character states changes on the tree.

If you do not want to add the character S manually, you can use `ALTree-add-S`. For case/control data, the state of the character S is allocated depending on the proportion (p_h) of cases carrying the haplotype h compared to the proportion p_0 of cases in the whole sample.

- if $p_h < p_0 - \epsilon\sqrt{\frac{p_h \times (1-p_h)}{n_h}}$, S is coded "C" or "0" (high number of controls);
- if $p_h > p_0 + \epsilon\sqrt{\frac{p_h \times (1-p_h)}{n_h}}$, S is coded "G" or "1" (high number of cases);
- else, S is coded "?" (undetermined).

with n_h being the number of individuals carrying the haplotype h.

For quantitative data, the state of "S" depends on the mean of the quantitative trait in a given branch of the tree μ, the mean of the quantitative trait in the whole data set μ_0 and the standard deviation of the quantitative trait in the whole data set, σ_0.

- if $\mu > \mu_0 + \epsilon \times \frac{\sigma_0}{\sqrt{n}}$ "S" is coded "H" (high level);
- if $\mu < \mu_0 - \epsilon \times \frac{\sigma_0}{\sqrt{n}}$ "S" is coded "L" (low level);
- else, "S" is coded "?" (undetermined)

n being the sample size, and ϵ, an inflation coefficient to be chosen by the user.

1.2.4 Computation time (in 2005)

We measured the computation time on a Pentium III, 930 MHz, 512 Mo of RAM. We used a data set of 363 individuals (cases and controls) genotyped for 7 SNPs defining 33 different haplotypes. The reconstructed phylogenetic tree had 6 levels. On this data set, the association test runs in about 24 hours (p-value evaluated by 100 000 permutations, the complexity of the program being linear with respect to the number of permutations). The localisation test runs in about 10 seconds (2 000 equiparsimonious trees analysed, the complexity of the program being linear with respect to the number of analysed trees).

In fact, for the association test, the computation time increases with the number of permutation and with the number of levels in the tree (the number of levels being tightly linked to the number of haplotypes in the data sets, which depends on the number of SNPs and of the LD between the SNPs). We tested the software for up to 1000 SNPs corresponding to 417 haplotypes. In this case, the association test runs in about 6 minutes (for one permutation only). Three to four minutes should be added per supplementary permutation. With such a data set, we can see that the evaluation of the p-values with the permutation procedure (10 000 to 100 000 are required) is not realistic on this kind of computer. However, the software can be used to look for association without using the permutation procedure.

The localisation test runs very quickly and depends on the number of equiparsimonious trees analysed. On the data set with 1000 SNPs, the localisation test runs in 10 seconds for one tree.

Chapter 2

Installing the software

The software can run on various Linux/Unix platform.

2.1 Requirements

2.1.1 Phylogeny reconstruction programs

Before using `ALTree`, you must build a phylogeny of the haplotypes. Three phylogeny software are compatible with our program:

- `paup` Swofford [2002]: available at `http://paup.csit.fsu.edu/`. This software is not free software and must be purchased (100$ for the unix version).
- `phylip` Felsenstein [2004]: freely available at `http://evolution.genetics.washington.edu/phylip.html` (only usable for the association test)
- `paml` Yang: freely available at `http://abacus.gene.ucl.ac.uk/software/paml.html`. As stated by its author, `paml` is not good at tree making. So we advise you to use another software to build the tree (for example, `phyml` Guindon and Gascuel [2003] and then to use `paml` to estimate the character states at each node.

Note: *Currently, only the outputs from the parsimony method implemented in `paup` (command `set`, option `criterion` set to "parsimony") and in `phylip` (program `mix`) are compatible with our software. If you want to use maximum likelihood (ML), we suggest you to use your favorite software to compute the ML tree and then, to use `paml` to estimate the character states at each node.*

2.1.2 Required tools

`perl` is required to run `ALTree`. `perl` version 5.8.7 or higher should work. Lower versions can work, but they have not been tested.

If you want to build the program from sources, you will also need a C compiler such as `gcc` and the GNU `make` program. They are available on most Unix plateforms. Otherwise, a debian package containing the binary files is available.

2.2 Installation on a linux platform

To install this module type the following:

```
perl Makefile.PL
make
make test
make install
```

If you prefer to install it in your home directory, then type the following:

```
perl Makefile.PL PREFIX=~
make
make test
make install
```

In this case, do not forget to add `~/bin` in your `PATH` and `~/lib/perl/`*`perl_version`*`/` in `PERL5LIB` if they are not already present. For example:

```
PATH=~/bin:$PATH
export PATH
PERL5LIB=~/lib/perl/5.8.7/:$PERL5LIB
export PERL5LIB
```

Chapter 3

ALTree-convert

This program converts the output of the haplotype reconstruction programs to input for phylogeny reconstruction programs. Each option has a long name (which must be preceded by --) and some of them also have a short name (which must be preceded by -).

3.1 Summary of the different options

`--version`	program version
`--short-help|-h`	brief help message
`--help`	help message with option descriptions
`--man`	full documentation
`--first-input-file|-i` *file*	Input file 1
`--second-input-file|-j` *file*	Input file 2 (not mandatory, see explanations below)
`--output-file|-o` *file*	Output file 1
`--case-control-output|-c` *file*	Output containing the number of cases/controls
`--reconstruct-prog|-r` **`phase|famhap`**	Name of the haplotype reconstruction program
`--phylo-prog|-p` **`paup|phylip`**	Name of the phylogeny reconstruction program
`--data-type|-t` **`DNA|NUM`**	Type of data: DNA (ATGCU) or NUM (0-9)

3.2 How to get help?

3.2.1 option --short-help or -h

This option displays a short help message which recapitulates all the options available.

3.2.2 option --help

This option displays a message with a description of the different options.

3.2.3 option --man

This option displays the man page for the program.

3.2.4 option --version

This option gives the number of the version currently used.

3.3 Input files

This program takes as input files the output files of the haplotype recontruction programs. Currently, only `phase` (for case/control data) and `FamHap` (for family data) output files are allowed, but we plan to extend the number of haplotype recontruction programs usable. The name of the haplotype reconstruction program used to generate the input files must be specified after the -r option.

3.3.1 Using `phase` output file

Two different cases must be considered:

- The case-control status of each individual has been specified in the input file for phase and phase has been run with the -c-1 option. In this case only one input file is necessary for `ALTree-convert`: the phase output file (let's call it out.phase). In this case, the program must be run like this:

```
altree-convert -r phase -i out.phase -other_options
```

- The case-control status of each individual has not been specified in the input file for phase. In this case, two input files are necessary: the phase output file (out.phase) and another file which specifies the disease status for each individual (status.phase). This file consists in two rows: the first contains the individual's ID and the second, their disease status (0=control, 1=case). In this case, the program must be run like this:

```
altree-convert -r phase -i out.phase -j
status.phase  -other_options
```

3.3.2 Using `FamHap` output files

The program `ALTree-convert` is designed to use files generated with `FamHap` 15 (and not with `FamHap` 12!), `FamHap` 16 has not been tested yet. Two options are necessary:

dp : to take the disease status into account in the haplotype reconstruction

P : to make sure that all the haplotypes are present in the output file

Two input files are necessary for `ALTree-convert`: the `FamHap` output file whose name has been chosen by the user (let's call it out.famhap), and the output file called input_name_H1_HAPLOTYPES. In this case, the program must be run like this:

```
altree-convert -r famhap -i out.famhap -j
H1_HAPLOTYPES -other_options
```

3.4 Output files

Two different output files are generated:

The main output file. Its name should follow the -o option. This file is an input file for the phylogeny reconstruction programs `paup`, `paml` or `phylip`.

The second output file. Its name should follow the -c option. It contains the number of times a given haplotype is carried by case and control individuals.

3.4.1 Generating `paup` input files (*.paup)

The file generated is a nexus file containing the options for `paup` necessary to run `ALTree` after `paup`. This is only an example of a paup file: we choose to root the tree using an ancestral sequence, but this is not necessary and this file should be modified according to your data. Examples of `paup` input files can be found in the test directory: they are labeled *.paup.

The output file is not a valid `paup` input file. Some options are indicated within square brackets and must be specified by the user before running `paup`:

- the sequence of the ancestral haplotype
- the maximum number of trees `paup` must find
- the method to optimise character state changes (`acctran`/`deltran`)
- the name of the different files generated
- the number of trees described by paup in the log file (we advise you to keep all the trees, and to limit the number of trees that are analysed later, when running `ALTree`).

The chosen option must be put out of the square brackets because `paup` ignores what is written within square brackets.

3.4.2 Generating `phylip` or `paml` input files (*.phy)

The file generated is the simplest phylip (also used by `paml`) format. The first line contains the number of haplotypes and the number of sites and the following lines contains an identifier for the haplotype (Hxxx) and the haplotype sequence.

3.4.3 The second output file

The name of the second output file must follow the -c option. This file contains the number of times a given haplotype is found in cases and controls. The file format is the following: the label of each haplotype and the number of cases and controls carrying it are specified, separated by spaces or tabulations. The number of cases carrying a given haplotype is preceded by the letter "m" and the number of controls is preceded by the letter "c".

Example of such a file:

```
H002    m015    c001
H003    m000    c001
H001    m000    c002
```

Other examples may be found in the test directory. These files are always labeled "nb_cas_control.txt".

3.5 Other options

3.5.1 The phylogeny reconstruction program

You must specify the name of the phylogeny reconstruction software that will be used after the option -p or --phylo-prog option so that the corresponding output file can be generated.

3.5.2 The haplotype reconstruction program

The name of the haplotype reconstruction program (`FamHap` or `phase`) must be specified after the option -r or --recontruct-prog.

3.5.3 The type of data

The user must specify if the data are of type DNA (ATGC) or NUM (number from 0 to 9, for the current version of the program, numbers superior to 9 cannot be used). It must be specified after the -r option.

Chapter 4

ALTree-add-S

This program adds a new character (named S) to each haplotype corresponding to its disease status. Each option has a long name (which must be preceded by --) and some of them also have a short name (which must be preceded by -).

4.1 Summary of the different options

`--version`	program version
`--short-help|-h`	brief help message
`--help`	help message with option descriptions
`--man`	full documentation
`--first-input-file|-i` *file*	Input file 1
`--second-input-file|-j` *file*	Input file 2: nb cases/controls per haplotype
`--output-file|-o` *file*	Output file
`--epsilon|-e` *number*	ϵ parameter
`--data-type|-t` **`DNA|SNP`**	data type: SNP or DNA
`--proportion|-p` *number*	proportion of cases in the sample
`--data-qual|q` **`qualitative|quantitative`**	data type: qualitative or quantitative
`--outgroup|-g` *outgroup_name*	Name of the outgroup (if necessary)
`--low|-l`	forces the state of character S to be "?" for haplotypes carried by 1 individual

4.2 How to get help?

See the same section for the program `ALTree-convert` (page 9).

4.3 Input files

4.3.1 The sequence file (-i option)

The name of the input file containing the sequences must be specified after the -i option. This input file must be a valid `paup` (nexus) or `phylip`/`paml` input file. If it is a `paup` file, make sure that the line following the description of the last haplotype in the data block includes a semi colon only.

4.3.2 The trait file (-j option)

The name of the file containing informations about the trait must be specified after the -j option.

- If your trait is quantitative, the file must contain haplotype labels followed by the quantitative values measured for the individuals carrying these haplotypes. or homozygous individuals, quantitative values must be repeated twice;
- If your data are qualitative, the file must contain haplotype labels folled by the number of cases and controls carrying this haplotype separated by spaces or tabulations. The number of cases should be preceded by a "m" (or the word "case", possibly followed by a "_"), the number of controls should be preceded by the letter "c" (or the word "control", possibly followed by a "_").

Examples:

Case/control data		
H002	m12	c5
H019	m2	c6
H007	m54	c78

Quantitative data				
H008	9.54	11.45		
H005	7.73	11.43	10.6	13.8
H018	8.98			

4.4 Output file (-o option)

The name of the output file can be specified after the -o option. If the -o option is not present, the standard output is used.

The output file is a `paup` or `paml` input file. The character S is coded “G” or “1” for cases or high values of the quantitative trait and “C” or “0” for controls or low values of the quantitative trait. In the `paup` input file generated, a new command is added, which excludes the character S from the tree reconstruction process, and includes it in the table of apomorphies. If you want to use `paml`, no such command exists. We advise you to reconstruct the phylogeny on the data set without the character S by using your favorite phylogeny reconstruction program. Then, you give that tree and the data-set with the S-character to `paml` to obtain the apomorphie list.

4.5 Other options

4.5.1 Proportion of cases in the sample (qualitative data only)

The proportion of cases in the sample must be specified after the -p option.

4.5.2 The epsilon value

It corresponds to the parameter ϵ (see the description of the program in section 1.2.3, page 5). If ϵ is high, haplotypes will more often have a character S coded “?”. To give an idea, in our article [Bardel et al., 2005], ϵ was set to 1.

4.5.3 Data type: sequence

The -t option must be followed either by *SNP* or by *DNA*. *SNP* should be used if you have numerical data (characters coded from 0 to 9). *DNA* must be used if you have DNA data (A, T, G, C).

4.5.4 Data type: trait

The software can deal with qualitative data (case/control) or quantitative data. The -q (or –data-qual) option must be followed by either *qualitative* or *quantitative*, depending on your data.

4.5.5 Haplotypes carried by only 1 individual

The -l option is not mandatory: if it is present, S is coded "?" for all the haplotypes present only once in the sample, whatever the disease status of the individual carrying it. If -l is not specified, the state of S will be chosen according to the formula (see section 1.2.3, page 5).

4.5.6 Name of the outgroup

If the outgroup is not specified in the file containing the number of cases an controls but is in the file containing the sequences, the name of the outgroup must be provided to `ALTree-add-S` so that the program can identify the outgroup sequence. For this sequence, the state of the character S will be "?".

Chapter 5

ALTree

This program can perform either an association test or a localisation test. Each option has a long name (which must be preceded by --) and some of them also have a short name (which must be preceded by -).

5.1 Summary of the different options

`--version`	program version
`--short-help|-h`	brief help message
`--help`	help message with option descriptions
`--man`	full documentation
`--association|-a`	perform the association test
`--s-localisation|-l`	perform the localisation test
`--first-input-file|-i` *file*	output file from phylogeny program
`--second-input-file|-j` *file*	nb cases/controls per haplotype
`--output-file|-o` *file*	output file
`--data-type|-t` **DNA|SNP**	type of data
`--data-qual|-q` **qualitative|quantitative**	data type: qualitative or quantitative
`--remove-outgroup`	remove the outgroup sequence for the analysis
`--outgroup` *outgroup_name*	specify the name of the outgroup sequence
`--anc-seq` *ancestral_sequence*	ancestral sequence (only useful with `phylip`)
`--tree-building-program|-p` **PHYLIP|PAUP|PAML**	phylip or paup or paml
`--no-prolongation`	no prolongation of branches
`--chi2-threshold|-n` *value*	threshold value
`--permutations|-r` *number*	number of permutations to perform
`--number-of-trees-to-analyse` *number*	total number of trees to analyse
`--tree-to-analyse` *number*	number of the tree to analyse
`--s-site-number` *number*	position of the S character in the sequence
`--s-site-characters` *anc_state*`->`*der_state*	ancestral state -> derived state for S
`--co-evo|-e` **simple|double**	simple or double
`--print-tree`	print the tree with the character state changes in the output file

5.2 How to get help?

See the same section for the program ALTree-convert (page 9).

5.3 General options

These options are used both for association and for localisation test.

5.3.1 First input file (option --first-input-file or -i)

This file is the output file of the phylogeny reconstruction program.

If paup is used

To run ALTree, some informations must be present in the input file for ALTree (=output file of paup). In particular, the apomorphy list and a table contining branch lengths must be present (though branch lengths are not taken into account in the analysis, they are juste used to check if they are consistent with the apomorphy list). For these information to be present, in the *describetrees* command you must use the following options: *brlens=yes* and *apolist=yes*.

Examples of paup input files containing the options necessary to run ALTree are provided in the test/paup directory. These files are labeled ***.paup**.

If phylip is used

The input file for ALTree is the output file named "outfile" by phylip. Currently, ALTree only works with output data from the program MIX (0/1 data). We plan to adapt it to other reconstruction program.

To generate a correct input file for ALTree, you must use different options for phylip depending on your rooting method:

- If you want to root the tree using an outgroup: you must root the tree on the chosen outgroup by using the option *o*.
- If you want to root the tree using the ancestral character states: you have to prepare a file named **ancestors** containing the ancestral sequence. Then, when running phylip, you must use the option *a* (see the phylip manual for more information).

Moreover, the states at all nodes of the tree must appear in the output file, so you must set the option *5* to *yes*.

If paml is used

The input file for ALTree is the output file named "rst" by paml.

5.3.2 The trait input file (option --second-input-file or -j)

If you analyze case/control data, this input file consists in lines containing the label of each haplotype followed by the number of cases and controls carrying it separated by spaces or tabulations. The number of cases should be preceded by a "m"(or the word "case", possibly followed by a "_"), the number of controls should be preceded by the letter "c" (or the word "control", possibly

followed by a "_").
Example of such files are given in the test directory. These files are always labeled **nb_cas_control.txt**.

If your trait is quantitative, the file must contain haplotype labels followed by the quantitative values measured for the individuals carrying these haplotypes. or homozygous individuals, quantitative values must be repeated twice.

5.3.3 Output file (option --output-file or -o)

You can choose the name of the output file by using the --output-file or -o option. If this option is not specified, the standard output is used.

5.3.4 Name of the phylogeny program used (option --tree-building-program or -p)

After the option -p, you must specify which phylogeny reconstruction program (`paup`, `phylip` or `paml`) was used to generate the first input file.

5.3.5 Data type: sequence (option --data-type or -t)

The option -t must be followed either by *SNP* or by *DNA*. *SNP* should be used if you have numerical data (from 0 to 9). *DNA* must be used if you have DNA data (A, T, G, C). Warning: the DNA option currently does not work if you have reconstructed the phylogeny with phylip.

5.3.6 Data type: trait (option --data-qual or -q)

The software can deal with qualitative data (case/control) or quantitative data. The -q (or -data-qual) option must be followed by either qualitative or quantitative, depending on your data.

5.3.7 Print tree (option --print-tree)

If this option is specified, the tree with the character state changes along the branches will be written in the output file. It may especially be useful when you are performing the localisation analysis, because in this case, the tree is not written in the output file by default.

Warning: if several trees are analysed, with the --print-tree option, they will all *be printed in the output file.*

5.4 Association test (option --association or -a)

When the -a option is used, the program will perform the phylogeny-based association test.

5.4.1 Options specific to the association test

Removing the outgroup (option --remove-outgroup)

The outgroup may either be a sequence of the sample for which the number of cases and controls carrying it is defined or a sequence for which we don't have any cases or controls (for example, an ape sequence). In this last case, the outgroup must be removed from the sample before the analysis. It is possible by specifying the option --remove outgroup.

Name of the outgroup (option --outgroup)

This option will be useful in two cases:

- If you work on an unrooted tree (with `paml` or `paup`): for the association test, the tree *must be rooted* because the analysis starts from the root. You must then specify the name of the sequence you choose as outgroup after the option --outgroup and `ALTree` will perform the rooting.
- If you want to remove an outgroup from the analysis: if you work with `phylip`, you will have to specify the name of the outgroup which should be removed after the option --outgroup. This is not necessary with `paup` because the outgroup is identified by a star in the paup output file, so `ALTree` can find it.

Number of permutations (option --permutations or -r)

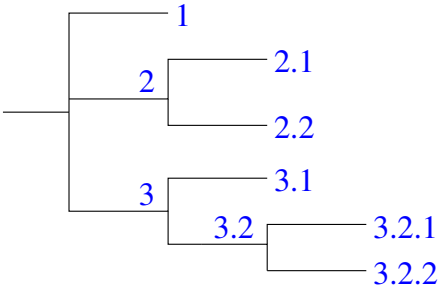
The program can compute a type I error corrected for multiple testing associated with the test by using a permutation procedure such as the one described in Ge et al. [2003] and in Becker and Knapp [2004]. In this case, the user must define the number of permutations to perform to evaluate the type I error by using the --permutation (or -r) option. This number should be high, but the higher it is, the longer the computation time will be. Depending on the studied data sets, we suggest this number to be chosen between 10000 and 100000.

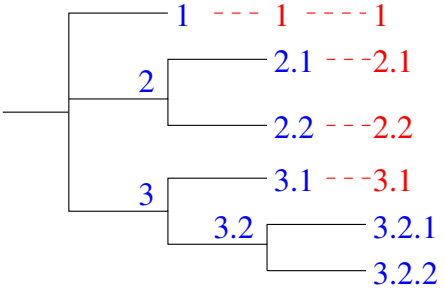
Threshold for chi-square significance (option chi2-threshold or -n)

If you do not want to compute the exact type I error by permutation, a significance threshold for the chi-squares can be chosen by the user using the --chi2-threshold (or -n) option. In this case, you must put the --permutation option to zero.

Branch prolongation (option --no-prolongation)

If the --no-prolongation option is specified in the command line, the different branches of the tree will not be prolonged. (see figure I).





	With the --no-prolongation option	Without the --no-prolongation option
	The chi-squares are calculated between[1]:	
Level 1:	1 - 2 and 3	1 2 and 3
Level 2:	2.1 - 2.2 - 3.1 - 3.2	1 - 2.1 - 2.2 - 3.1 and 3.2
Level 3:	3.2.1 and 3.2.2	1 - 2.1 - 2.2 - 3.1 - 3.2.1 and 3.2.2

Figure I: Effect of the -b option

Warning: This option is currently under development. At present, the program has only been tested without *the **--no-prolongation** option specified. If you choose not to, you may encounter some problems.*

Ancestral sequence (option --anc-seq)

This option is only necessary when the tree is rooted using an ancestral sequencce with `phylip`. In this case, the ancestral sequence not being in the output file of `phylip`, `ALTree` cannot read it directly. So, you have to enter it manually after the --anc-seq option.

Choice of the tree that will be analysed (option --tree-to-analyse)

This option enables the user to specify the number of the tree that will be analysed among all the equiparsimonious trees present in the input file (the --tree-to-analyse must be followed by the number of the chosen tree in the input file). If this option is not specified, the tree will be randomly drawn among all the equiparsimonious trees.

5.4.2 Description of the output file

Examples of output files are displayed in the test directory. They are labelled *.asso.

The output file shows the tree, with the number of cases an controls at each nodes. At the root of the tree, there is a list of the different tests performed on the tree: the level of the test is indicated within square brackets, followed by the number of degrees of freedom (df=), the value of the chi-square test and the corresponding p-value. In a second part of the file, a list of the p-values estimated by permutations (but non corrected for multiple testing) for each level of the tree is provided. Then, the last line gives the corrected p-value for the test.

5.5 Localisation test (option --s-localisation or -l)

5.5.1 Options specific to the localisation

Number of trees (option --number-of-trees-to-analyse)

With this option, you choose the number of trees to use in the localisation test. These trees are randomly sampled without replacement among all the equiparsimonious trees in the first input file.

--s-site-number

With this option, you specify the position of the character S in the haplotypes. The first site is numbered 1.

--s-site-characters ancestral state -> derived state

With this option, you specify which state is the ancestral state and which state is the derived state for the character S. The two states must be separated by the symbol "->". For example, if the character S has two states 1 and 2, 1 being the ancestral state, you will use the option as follows:
`ALTree` [other options] --s-site-characters "1->2"
Be careful: this option is *case sensitive* and the *quotes are mandatory*.

--co-evo|e simple|double

This option enables the user to choose how the V_i are calculated.

option "simple" This option corresponds to the calculation of V_i described in Bardel et al. [2005]. Please refer to this publication for more information.

option "double" This option corresponds to a new method to calculate V_i. This method seems to be more appropriate because it takes into account the two senses of character state changes. Here is a short description of this new calculation method (one studied tree):

- Let $E_i^{0\to1}$ be the number of expected co-mutations of S (2 character states: T [control] and M [case]) and i (2 character states 0 and 1)

$$E_i^{0\to1} = \frac{(m_i^{0\to1} \times s^{T\to M}) + (m_i^{1\to0} \times s^{M\to T})}{b}$$

  where:

  $m_i^{0\to1}$ **(resp.** $m_i^{1\to0}$**) :** nb transitions $0 \to 1$ (resp. $1 \to 0$) of i
  $s^{T\to M}$ **(resp.**$s^{M\to T}$**) :** nb transitions $T \to M$ (resp. $M \to T$) of S
  b **:** nb branches of tree t

- Let $R_i^{0\to1}$ be the number of observed co-mutations of S and i on tree t
- $V_i^{0\to1}$ is calculated as defined in Bardel et al. [2005]:

$$\begin{cases} V_i^{0\to1} = 0 & if\ E_i^{0\to1} = 0 \\ V_i^{0\to1} = \dfrac{R_i^{0\to1} - E_i^{0\to1}}{\sqrt{E_i^{0\to1}}} & if\ E_i^{0\to1} \neq 0 \end{cases}$$

If more than one tree is studied, the $V_i^{0\to1}$ must be summed for all the trees.

5.5.2 Description of the output file

The output file contains only a list of the different V_i (in ascending order) for the different sites and the different character state transitions. The sites with the highest V_i are putative susceptibility sites.

Chapter 6

Example files

The **test** directory contains example files for the three phylogeny reconstruction programs. The files are grouped in four directories:

- **create_file** which contains files and instructions necessary to obtain `paup` or `paml`/`phylip` file formats from output files of the haplotype reconstruction program.
- **`paup`**, **`phylip`** and **`paml`** which contain files and instructions necessary to perfom association and localisation tests

In each directory, all the input and output files for all the programs and a bash script containing the different command lines are provided.

6.1 Obtention of input files for phylogeny reconstruction programs

The **create_file** directory is divided into 2 sub-directories: **paup_file** and **phy-paml_file**. In these directories, we present how to obtain input files for `paup` and `phylip` (or `paml`) from output files of the haplotype reconstruction programs `phase` and `FamHap`.

6.1.1 Creating `paup` input files from `phase` output file

The **paup_file** directory contains case/control data (12 SNPs genotyped for 100 case and 100 control individuals). The haplotypes are reconstructed using `phase` and the `paup` input file is generated using `ALTree-convert`. The different files available in this directory are the following:

caco.phase : an input file for `phase` containing the disease status for each individual

caco.phase.out : the main `phase` output file. It is used as the input file for `ALTree-convert`

caco.phase.out_* : other `phase` output files. They are not useful to run `ALTree`

caco.prepaup : the main `ALTree-convert` output file. It must be completed to become a valid `paup` input file

nb_cas_control.txt : the `ALTree-convert` output file containing the number of cases and controls carrying each haplotype

create_file : a bash script containing the two command lines to run respectively `phase` and `ALTree-convert`

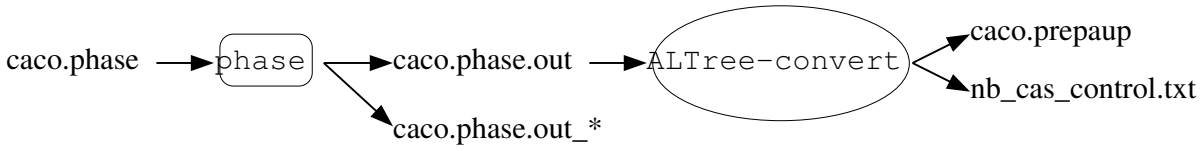


Figure I: Summary of the files and programs used to obtain input files for `paup`

6.1.2 Creating `phylip/paml` input files from `FamHap` output files

The **phy-paml_file** directory contains family data (10 SNPs genotyped for 100 trios: 2 parents + 1 affected child). The haplotypes are reconstructed using `FamHap` and the `phylip/paml` input file is generated using `ALTree-convert`. The different files available in this directory are the following:

fam19_0 : an input file for `FamHap` (linkage format without headers)

trio.fmh and fam19_0_H1_HAPLOTYPES : the two `FamHap` output files used by `ALTree-convert`

fam19_0_* : all other `FamHap` output files. They are not useful to run `ALTree`

trio.phy : the main `ALTree-convert` output file. It is an input file for `phylip`

nb_cas_control.txt : the `ALTree-convert` output file containing the number of cases and controls carrying each haplotype

create_file : a bash script containing the two command lines to run respectively `FamHap` and `ALTree-convert`

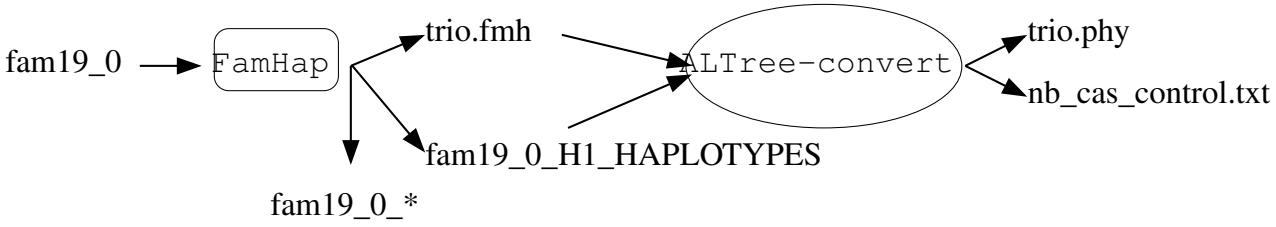


Figure II: Summary of the files and programs used to obtain input files for `phylip` or `paml`

6.2 Analysing `paup` files

In the **"test/paup/"** directory, six sub-directories can be found, each corresponding to a different way to root (or not) the tree using `paup`:

- **ancestor_absent**: In this directory, the tree is rooted using an ancestral sequence which is not in the data set (it can be a consensus sequence for example)
- **ancestor_present**: In this directory, the tree is rooted using an ancestral sequence which is in the data set (it can be the most frequent haplotype in the sample for example)
- **outgr_absent**: In this directory, the tree is rooted using an outgroup which is not carried by case or control individuals (it can be an ape sequence for example)

- **outgr_present**: In this directory, the tree is rooted using an outgroup which is in the data set
- **unrooted_absent**: In this directory, the tree is not rooted with `paup`. For the analysis, `ALTree` roots the tree using an outgroup (we choose the sequence H000), but *do not* take the outgroup into account for the association test
- **unrooted_present**: In this directory, the tree is not rooted with `paup`. For the analysis, `ALTree` roots the tree using an outgroup (we choose the sequence H000), and this haplotype *is* taken into account for the association test

All these directories are split in two sub-directories containing the files used to perform the association test (directory **association**) or the localisation test (directory **localisation**)

Warning: For the localisation test, as the rooting is not necessary, the question of the presence or not of an outgroup is irrelevant put the --permutation to zero(directories unrooted_absent and unrooted_present). The two localisation sub-directories thus only correspond to two different data sets analysed with `ALTree`.

6.2.1 Association test

All the association directories contain the same files:

caco.paup : the valid `paup` file

nb_cas_control.txt : the file containing the number of time each haplotype is carried by case and control individuals

test.res.log : the `paup` output file which is used as an `ALTree` input file

test.tree: the other `paup` output file, which is not useful for `ALTree`

1_caco.asso: the `ALTree` output file. The number of permutation being limited to 1, the corrected p-value doesn't mean anything!

run_altree: a bash script containing the two command lines to run respectively `paup` and `ALTree` (association test)

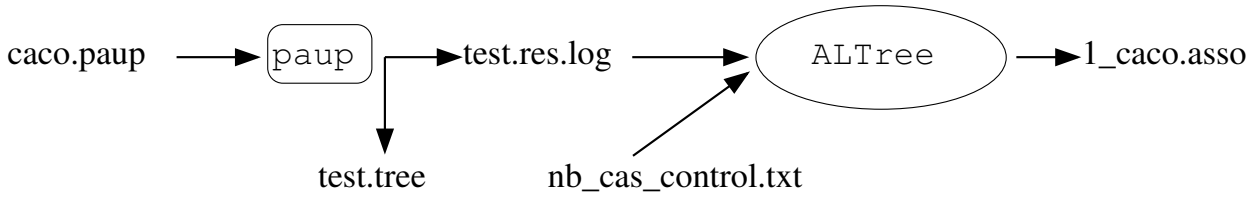


Figure III: Summary of the different files and programs used for the association test (using `paup`)

6.2.2 Localisation test

All the localisation directories contain the same files:

caco.paup: the valid `paup` file

nb_cas_control.txt : the file containing the number of time each haplotype is carried by case and control individuals

et_caco.paup: the output file of `ALTree-add-S`. It is a valid `paup` input file in which the character S has been added

test.res.log : the `paup` output file which is used as an `ALTree` input file

test.tree: the other `paup` output file, which is not useful for `ALTree`

caco.loc: the `ALTree` output file, result of the localisation test

run-prog: a bash script containing the three command lines to run respectively `ALTree-add-S`, `paup` and then `ALTree` (localisation test)

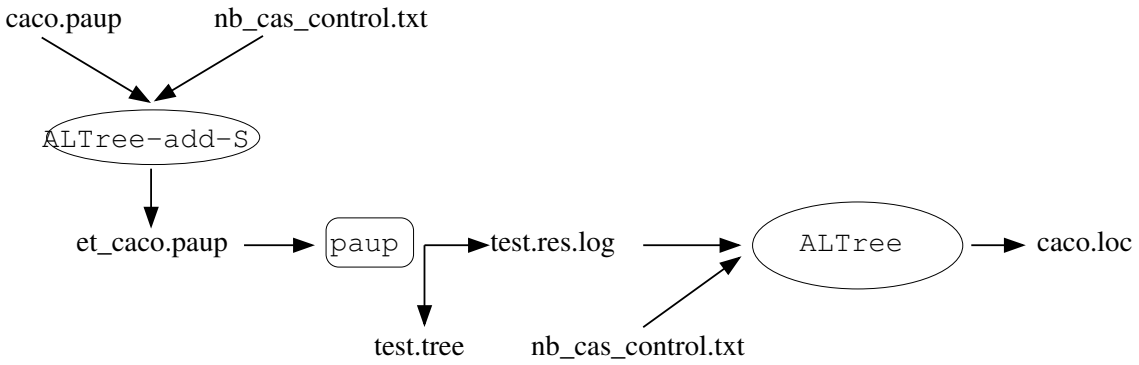


Figure IV: Summary of the different files and programs used for the localisation test (using `paup`)

6.3 Analysing `phylip` files

In the **`phylip`** directory, four sub-directories can be found, corresponding to various rooting methods. These directories are similar to the ones described for `paup`. They contains only one sub-directory named **association**. For the moment, `ALTree` cannot deal with `phylip` files as input files for the localisation test because when there are ambiguities in the apomorphie reconstructions, `phylip` keeps them in the output file and the state "?" is assigned to the ambiguous character. At present, `ALTree` cannot deal with these ambiguities.

6.3.1 Association test

All the association directories contain almost the same files:

trio.phy: the `phylip` input file

nb_cas_controls.txt: it contains the number of time each haplotype is carried by case and control individuals

outfile: the `phylip` output file which is used as an `ALTree` input file

outtre: the other `phylip` output file, which is not useful for `ALTree`

1_trio_phy.asso: the `ALTree` output file, result of the localisation test

run-altree: a bash script containing the two command lines to run respectively `phylip` and `ALTree` (association test)

ancestors: it contains the ancestral sequence. This file is needed only if the tree is rooted using an ancestral sequence (**ancestor_absent** and **ancestor_present** directories)

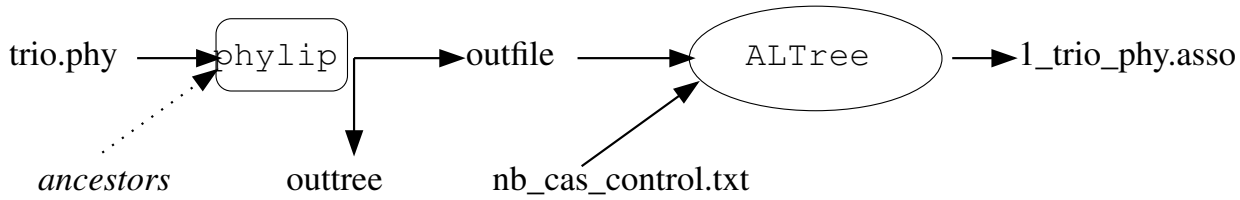


Figure V: Summary of the different files and programs used for the association test (using `phylip`)

6.4 Analysing `paml` files

In the directory **`paml`**, three sub-directories can be found:

tree_building_using_phyML: as stated by its author, `paml` is not a very good tool for tree reconstruction. In this example, we choose to reconstruct the phylogenetic tree using the software `phyml` [Guindon and Gascuel, 2003]. The phylogenetic reconstruction step has been performed in this directory

unrooted_absent and unrooted_present: which are similar to ones described for `paup`.

6.4.1 Phylogenetic tree reconstruction using `phyml`

The files necessary for the phylogenetic reconstruction can be found in the directory **tree_building_using_phyML**. The input file for `phyml` is the phylip format file named **trio2.phy**. The options used to run `phyml` are specified in the file **run_phyml**. The other files in the directory are `phyml` output files. In the following, we will only use the file **trio2.phy_phyml_tree.txt** which contains the reconstructed phylogenetic tree.

6.4.2 Association test

The two **association** directories contain the same files:

trio2.phy: the `phylip` format file containing the sequences. It is used as an input file for `paml`

nb_cas_controls.txt: it contains the number of time each haplotype is carried by case and control individuals

trio2.phy_phyml_tree.txt: the output file of `phyml`. It is also used as an input file for `paml`

baseml.ctl: the parameter file used by `paml`

rst: the `paml` output file which will be used by `ALTree`. It contains the apomorphy list and the tree structure

2base.t, lnf, mlb, rst1 and rub: all the other `paml` output files. They are not useful for `ALTree`

1_trio_ML.asso: the `ALTree` output file, result of the association test

run_altree: a bash script containing the two command lines to run respectively `paml` and `ALTree` (association test)

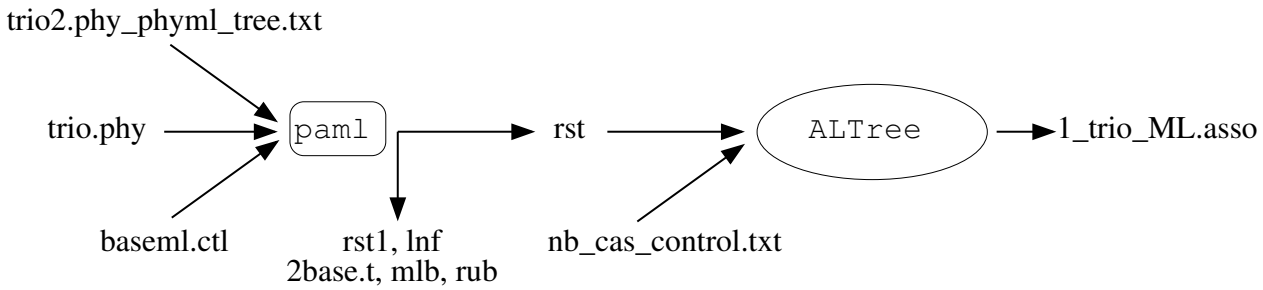


Figure VI: Summary of the different files and programs used for the association test (using `paml`)

6.4.3 Localisation test

With `paml`, only unrooted trees are obtained. These unrooted trees can be directly analysed with `ALTree`, so the question of the presence or absence of an outgroup is irrelevant. Only one localisation directory exists, it is located in the directory **unrooted_present**.

The **localisation** directory contains the following files:

trio2.phy: the `phylip` format file without the character S

et_trio2.phy: the `phylip` format file including the character S. It is one of the input file for `paml`

nb_cas_controls.txt: contains the number of time each haplotype is carried by case and control individuals

trio2.phy_phyml_tree.txt: the output file of `phyml` (tree reconstructed without taking the character S into account). It is also an input file for `paml`

baseml.ctl: the parameter file used by `paml`

rst: the `paml` output file which will be used by `ALTree`. It contains the apomorphy list and the tree structure

2base.t, lnf, mlb, rst1 and rub: all the other `paml` output files. They are not useful for `ALTree`

trio2.loc: the `ALTree` output file, result of the localisation test

run-prog: a bash script containing the three command lines to run respectively `ALTree-add-S`, `paml` and `ALTree` (localisation test)

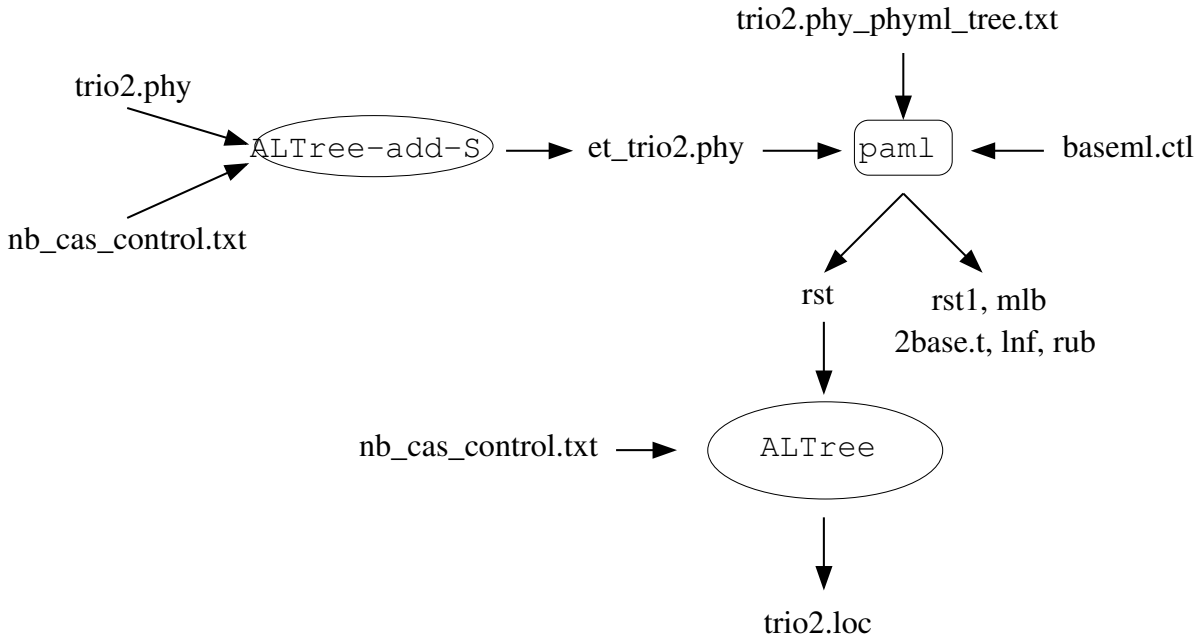


Figure VII: Summary of the different files and programs used for the localisation test (using `paml`)

Chapter 7

URLs where programs can be downloaded

7.1 Haplotype reconstruction programs

FamHap http://www.uni-bonn.de/%7Eumt70e/becker.html
phase http://www.stat.washington.edu/stephens/software.html

7.2 Phylogeny reconstruction programs

paup http://paup.csit.fsu.edu/
phylip http://evolution.genetics.washington.edu/phylip.html
paml http://abacus.gene.ucl.ac.uk/software/paml.html
phyml http://atgc.lirmm.fr/phyml/

Bibliography

C Bardel, V Danjean, J P Hugot, P Darlu, and E Génin. On the use of haplotype phylogeny to detect disease susceptibi lity loci. *BMC Genetics*, 6(24), 2005.

Tim Becker and Michael Knapp. A powerful strategy to account for multiple testing in the context of haplotype analysis. *Am J Hum Genet*, 75(4):561–570, Oct 2004.

J Felsenstein. Phylip (phylogeny inference package) version 3.6. `http://evolution.genetics.washington.edu/phylip.html`, 2004. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.

Y Ge, S Dudoit, and T P Speed. Resampling-based multiple testing for microarray data analysis. *Test*, 12:1–77, 2003.

S Guindon and O Gascuel. A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst biol*, 52:696–704, 2003.

M Stephens and P Donnelly. A comparison of bayesian methods for haplotype reconstruction from population genotype data. *Am J Hum Genet*, 73(5):1162–1169, Nov 2003.

M Stephens, N J Smith, and P Donnelly. A new statistical method for haplotype reconstruction from population data. *Am J Hum Genet*, 68(4):978–989, Apr 2001.

D L Swofford. `paup` phylogenetic analysis using parcimony. version 4.0b10. Sunderland, Massachusetts: Sinauer Associates, 2002.

Z Yang. Phylogenetic analysis by maximum likelihood. http://abacus.gene.ucl.ac.uk/software/paml.html.