User guide of the LSD software version 0.2

If you use this software, please cite: "**Fast dating using least-squares criteria and algorithms**", T-H. To, M. Jung, S. Lycett and O. Gascuel" (submitted manuscript).

All executable files and source code can be downloaded at: http://www.atgc-montpellier.fr/LSD/

1 Introduction

LSD, for Least-Squares Dating, is a C++ program that implements the two algorithms LD and QPD. Given a binary tree (rooted or unrooted) with branch lengths expressed in numbers of substitutions per site, and the sampling date of each tip, it estimates the substitution rate and the date of each internal node under the assumption of a strict molecular clock and a normal distribution of the errors affecting branch lengths. This model is robust regarding uncorrelated fluctuations of the substitution rate among branches. It yields a least-squares, likelihood function which has only one unique optimum that can be computed rapidly. LD is a linear time algorithm (i.e. with computing time proportional to the number of tree tips), and thus very fast. For example it takes less than 2 minutes to solve a rooted tree of 24,000 tips on a desktop computer. However, LD does not take into account the temporal constraints that the date of a node is required to be greater than or equal to that of its ancestors. QPD includes this condition by using the active set method to optimize our objective function under linear constraints. Yet it takes about half an hour for a tree of 24,000 tips. The users can choose either weighted or unweighted versions. The weights are the inverses of the error variances. They are used so that long branches, which could have large errors, have low weights, and thus small contributions in the objective function. The program can also estimate the root position for unrooted trees or re-estimate the root position for rooted trees. For rooted trees, the users can choose to re-estimate the root either around the given root, or on all branches. When all branches are required to be explored, the rooting function takes quadratic times under non constrained mode (LD), i.e. proportional to the square of the number of tree tips. For constrained mode (QPD), we developed a fast method that firstly pre-estimates the root without using constraints, then runs the constrained mode to improve the root position around this pre-estimated root. The slow method, that runs the constrained mode on all branches, is also available.

When (re)estimating the root position, the unweighted version is automatically selected (see paper for details and explanations on these choices and procedures). When all leaves are sampled at the same

time, the program estimates the relative dates by giving, by default, the root date = 0 and the leaves date = 1.

2 Installing the Program

The executable files are provided for windows, Linux and mac OS X. The source code is also provided so that you can compile it yourself by typing *./configure* and then *make*. The executable file will be generated in the directory *src*. Standard libraries of C++ are sufficient, no external library is required. If you wish to install the program on your system, type *make install* (or *sudo make install*) after compiled it.

3 Input data

LSD takes as input at least **a text file containing trees in NEWICK format**: this file contains a set of binary rooted or unrooted trees in NEWICK format, which have the same leaf set. The semicolon character ";" must be specified to indicate the end of the tree and a new tree must begin on a new line. For example:

```
(a:0.12,(b:0.02,c:0.32));
((a:0.35,c:0.17),b:0.22);
```

In the case that all leaves are sampled at the same time, then the sampled dates are not required, the program estimates the relative date by giving, by default, the root date = 0, and the leaves date = 1. You can also specify the root date, leaves date. In the other case, **a text file containing the sampling dates** must be provided. The first line must indicate the number of leaves. After that, each line contains the name of the tip following by its sampling date. Note that each taxon that presents in the trees must have one and only one sampling date. For example:

```
3
a 2000
b 1990
c 2012
```

Moreover, depending on the chosen options, the user can provide other input files:

- **A text file containing the name of the outgroup taxa:** This procedure is used to root trees using outgroup. If the input tree(s) contain(s) monophyletic outgroup taxa and the user wants to root the tree(s) by removing the corresponding clade from the tree(s), then a file containing the names of these taxa has to be provided. This file begins with the number of outgroup taxa, and then each line contains the name of a taxon. It should be noticed that the outgroup taxa MUST

be monophyletic and form a subtree separated from the rest of the tree(s); otherwise, it is not possible to remove the outgroup. This procedure inputs unrooted trees with outgroup, and outputs rooted trees without outgroup. An example of a file containing outgroups:

```
2
outgroup1
outgroup2
```

- **A text file containing the rates:** if the users want to estimate the dates of internal nodes by using known substitution rates, they can give these rates through a file in which each line contains the rate of the corresponding tree in the tree file.

4 Output results

LSD provides four different output files:

- The output file with the suffix ".result" contains the estimates of the substitution rates, the root date of each tree, and the value of the objective function in the equation (2) of the paper. It also recalls the options being selected by the user.
- The output file with the suffix ".newick" contains the trees in NEWICK format with estimated branch lengths expressed in substitution per site deduced from the estimated dates and substitution rates.
- The output file with the suffix ".date.newick" contains the trees in NEWICK format with branch lengths expressed in elapsed times between the two branch extremities.
- The output file with the suffix ".nexus" contains the trees with estimated branch lengths (number of substitution per site) as well as estimated dates at each node in nexus format. In this case, the tree can be directly open with FigTree to see it in time-scaled format.

Moreover, if the input trees are unrooted, the program estimates the position of the root and gives the rooted trees to the users. In the case where the input trees contain outgroup and you ask the program to remove these outgroup, then the rooted, ingroup trees will be written in the files with the suffix "_ingroup" in NEWICK format.

5 Running the program

LSD is launched by either double click on the executable file; or from a command-line, type ./lsd from the directory containing the executable file (type *lsd* from anywhere if the program is installed) with or without argument. Sometimes, you have to add executable permission (*chmod +x file*) to the file if it was

not compiled from your computer. If LSD is launched without any argument, we have a **PHYLIP-like interface.** Otherwise, we have a **command-line interface**.

5.1 PHYLIP-like interface

First, you will be asked to enter the names of the input tree file and of the input date file. Then an interface will appear with the options to choose/switch by letters. Choice of options is not case sensitive, for example "q" or "Q" is recognized as the same. All options are the same as in the command-line interface (cf. section 5.2 for further details). For example option `[I]` corresponds to option `–i` in the command-line interface and let you modify the name of the input tree file. By typing the indicated letter, you will either switch its status, or modify its value. Depending on the cases, the value should be a file name, a positive integer or a positive floating point number.

5.2 Command-line interface

Let's take a simple example with an input rooted tree file rooted_`tree.txt` and an input date file `date.txt`. To execute the program with temporal constraints, with weights (variances), sequence length 500, and 2 trees, the user should include all of these options in the command line by using the following syntax from the directory that contains the executable file:

```
$ ./lsd –i rooted_tree.txt –d date.txt –c –v –s 500 –n 2
```

If you want to run the program without constraints, remove option `–c`. If you want to re-estimate the position of the root around the given root, add the `–r l` option. If you want to re-estimate the root position on all branches using fast method, add option –r a. If you want to re-estimate the root position by using constrained mode on all branches (slow method), then add option –r as.

If the tree is not rooted and there is no outgroups in the tree, use option –r a (or –r as) to estimate the root positions, for example:

```
$ ./lsd –i unrooted_tree.txt –d date.txt –c –r a
```

If the tree contains outgroups, use option –g to specify the outgroups file name. The program will remove the outgroups to obtain the root, for example:

```
$ ./lsd –i tree_with_outgroup.txt –d date.txt –g outgroup_file -c
```

If you want to estimate the relative dates, for a rooted tree, using variances under constrained mode:

```
$ ./lsd –i rooted_tree.txt –c –v –a root_date –z tips_date
```

or just

`$ ./lsd –i rooted_tree.txt –c –v` (in this case T[root] = 0 and T[tips] = 1 by default).

All options are explained in details in the following:

- `-a root date:` If the dates of all tips are equal (which is given by option -z), you must use this option to provide the root date. In this case, the input date file can be omitted, and the program estimates only the relative dates based on the given root date and tips date. By default, T[root]=0 and T[tips]=1.

- **`-b`** `positive_real`: it is the positive integer *b* in the formula of variance for branch lengths (see option –v for the formula and paper for details). It is 10 by default.
- **`-c`** `constraints` choose this option to include temporal constraints. Otherwise and by default, the program does not use constraints and is faster (to be used for preliminary analyses, using constraints is highly recommended).
- **`-d`** `dates file`: this option is necessary to load the file containing the dates of each leave of the input trees. The format of this file should be

  n
  TAXON1 DATE1
  TAXON2 DATE2
  …
  TAXONn DATEn

  If this option is omitted, the program will estimate relative dates by giving T[root]=0 and T[tips]=1.
- **`-g`** `outgroup file`: if your data contain one or more outgroup taxa, specify the name of the file containing this outgroup. The program will remove corresponding taxa from the input trees and use the rooted ingroup trees for date and rate estimations. The format of this file should be:

  n
  OUTGROUP1
  OUTGROUP2
  ...
  OUTGROUPn
- –**h**: print the help message.
- **`-i`** `trees file`: this option is used to specify the input trees file in NEWICK format.

- **-n** positive_integer: the number of trees in the input file that you want to analyze.
- **-o** output_file: this option indicates the base name of the files used by the program in which it will write the results. By default the base name is the same as the input file.
- **-r** rooting_method: This option is used to specify the rooting method to estimate the position of the root for unrooted trees, or re-estimate the root for rooted trees. The principle is to search for the position of the root that minimizes the objective function.
    - If the tree is rooted, then either using operand "l" for searching the root around the given root, or using "a" for searching the root on all branches. Moreover, when the constraint mode is chosen (option -c), method "a" firstly estimates the root without using the constraints. After that, it uses the constraints to improve locally the position of the root around this pre-estimated root. To use constraint mode on all branches in this case, please specify "as".
    - If the tree is not rooted, then the program searches the root on all branches. Similarly for the previous case, if the constraint mode is chosen, method "a" uses only constraint mode to improve the root position around the pre-estimated root which is computed without constraints. To use constraints on all branches, use "as".
- **-s** sequence_length: this option is used to specify the length of the sequences in the input trees. It is used to calculate the variances-weights (see option -v). By default it is equal to 1,000.
- **-t** lower_bound_rate: this is the lower bound for the estimated rate. It is 0.00001 by default.
- **-v** variances: variances are used to reduce the weights of estimated errors in the input branch lengths. The formula for the variance is $v_i = b_i + \frac{b}{s}$ where b_i is the length of the branch i, b is a positive real defined by option -b, and s is the sequence length defined by option -s. The weights in the objective function (equation (2) in the paper) are the inverses of the variances.
- **-w** given_rate_file: if you know the substitution rate(s), give the file that contains them here. The program will use these rates to estimate the dates of the internal nodes. This file contains as many substitution rates (one per line) as input trees.
- **-z** tips_date: This option is used to give the date of the tips when they are all equal. It must be used with option -a to give the root date. In this case the input date file can be omitted, and

the program estimates only the relative dates based on the given root date and tips date. By default, T[root]=0 and T[tips]=1.