

ScisTree2: A Program for Large-Scale Cell Lineage Tree Inference and Genotype Calling from Noisy Single Cell Data Using Efficient Local Search

User Manual

Version 2.2.1.0

July 30, 2025

Yufeng Wu

School of Computing, University of Connecticut Storrs, CT 06269, U.S.A.

Email: yufeng.wu@uconn.edu

©2018-2025 by Yufeng Wu. This software is provided “as is without warranty of any kind. In no event shall the author be held responsible for any damage resulting from the use of this software. The program package, including source codes, executables, and this documentation, is distributed free of charge. If you use the ScisTree2 program in a publication, please cite the following reference:

Haotian Zhang, Yiming Zhang, Teng Gao and Yufeng Wu, Large-scale Inference of Cell Lineage Trees and Genotype Calling from Noisy Single-Cell Data Using Efficient Local Search, under review, 2025.

1 Getting Started with ScisTree2

1.1 Program availability

ScisTree is written in C++. Executables for popular platforms such as Linux 32 bits or 64 bits and MacOS are downloadable from GitHub:

<https://github.com/yufengwudcs/ScisTree2>.

Download

I would recommend to download ScisTree2 from the above GitHub page by (i) first clicking the green button “Code”, and (ii) clicking “Download ZIP”. This would save a file in your file-downloading folder.

After downloading the softwares, to compile the code, first put the zip file in the directory you'd like and unzip it: use "unzip" and tar commands such as:

▷ unzip <ScisTree2-main.zip>

Move to the source code folder:

▷ cd ScisTree2-main>

▷ cd ScisTree2-source-code>

Then type:

▷ make at the prompt. This creates an executable called stells, which can be run by typing

▷ ./scistree at the prompt. You should see the following output:

*** SCISTREE ver. 2.2.0.0, October 24, 2024 ***

Usage: ./scistree [options] [input file]

Options:

-v Turn on verbose mode

-T [nthread] Set number of threads to nthread

-n Only build simple neighbor joining tree (may be useful for very large data)

-e Output mutation tree (may not be binary tree) from called genotypes branch labels.

-e0 Output mutation tree but don't output labels (for visualizing large trees).

-q Use NNI local tree search (NNI is faster but less accurate)

To make ScisTree2 do some useful work, you will need to specify some input options - see below.

1.1.1 The original ScisTree

First, where does the name ScisTree come from? It stands for {S}ingle {c}ell {i}nfinite {s}ites {T}ree.

The original ScisTree was published in:

Yufeng Wu, Accurate and Efficient Cell Lineage Tree Inference from Noisy Single Cell Data: the Maximum Likelihood Perfect Phylogeny Approach, Bioinformatics, Volume 36, Issue 3, Pages 742-750, 2020.

ScisTree was designed to infer cell lineage tree (CLT, the evolutionary tree of single cells) and call genotypes from noisy single cell data. ScisTree takes uncertain genotypes in the form of genotype probability. This is because genotypes called from single cell sequence data tend to be very noisy. It is usually difficult to call a fixed genotype at a position. Different from several existing methods for cell tree inference, ScisTree works with uncertain genotypes with **individualized** genotype probabilities. That is, each genotype (at a site and a cell) can have its own probability, which specifies how likely this genotype has a particular genotype state. This can better utilize information contained in single cell data: often some genotypes in the single cell data can be almost fully determined while others have much larger uncertainty.

ScisTree infers cell tree and call genotypes simultaneously. It can deal with single cell technological noises such as doublets. One key advantage of ScisTree over some existing

methods is that ScisTree is very efficient: it works for data with hundreds of cells and thousands of single nucleotide variants (SNVs). My tests show that ScisTree can be 100 times or more faster than existing methods such as SCITE.

1.2 How does ScisTree work?

ScisTree assumes the infinite sites model. This allows a very simple algorithm for finding the optimal CLT with *maximum posterior probability* of the cell tree and genotypes that maximize the probability of the data under the infinite sites model. Refer to the paper for more details on the methodology of ScisTree.

1.3 What is new in ScisTree2?

While the original ScisTree appears to work well for small or medium sized data, it becomes slow when the data size increases. When there are more than 1,000 cells, ScisTree starts to take much longer to run. As larger single cell data starts to become available, it is useful to make CLT inference practical for larger data.

ScisTree2 is designed to support CLT inference for very large data. It implements a much more efficient tree local search approach than that in the original ScisTree. This new local search uses subtree prune and regraft (SPR), while the original ScisTree only implements nearest neighbor interchange (NNI) local search. The SPR local search explores a much *larger* tree space than the NNI local search. The key advantages of ScisTree2 are:

1. ScisTree2 is more accurate than the original ScisTree because it searches larger tree space and can find better CLT with higher probability than ScisTree.
2. ScisTree2 runs faster than ScisTree even when exploring a large tree space. This is due to a novel local search algorithm that is orders of magnitude faster than brute force.

2 Functionalities and Usage of ScisTree2

2.1 Preparing inputs

ScisTree2 takes the same input as the original ScisTree. To run ScisTree2, the user needs to provide the genotype probabilities for the SNVs of the cells under study. The genotype probability is specified in a matrix. Genotypes can be either binary or ternary. Here is a simple example of the input.

This is an example.

```
HAPLOID 5 4 c1 c2 c3 c4
s1 0.8 0.02 0.8 0.8
s2 0.02 0.02 0.02 0.8
s3 0.8 0.02 0.02 0.8
s4 0.02 0.8 0.8 0.8
s5 0.8 0.02 0.8 0.02
```

ScisTree2 ignores lines starting with #, which are considered to be comments. The first (non-comments) line should have: `< FORMAT >< num – sites >< num – cells >< cell – name – 1 >< cell – name – 2 > ...`. Here, “FORMAT” can be “Haploid”. Haploid format refers to binary genotypes. That is, each allele in the genotype matrix is either 0 or 1. Note: the original ScisTree also supports “Ternary” format. As of now, ScisTree2 doesn’t support Ternary format.

The user needs to specify the number of (SNV) sites and the number of cells. There is a line for the probabilities for genotypes of each site. The line starts with a string for cell’s name (in the above example, s_1, \dots, s_5). Then, the line contains probability for each cell at this site. For each genotype at a site, one specifies the genotype probability sequentially: for binary genotype, use a single value to specify the probability of genotype 0. In the above example, at the first site s_1 , genotype probabilities given mean that the four cells have probability of 0.8, 0.02, 0.8 and 0.8 of being genotype 0. Note that the probability of being genotype 1 is not specified since the probabilities of being 0 and 1 add up to 1.

2.2 Usage

To run ScisTree2, you must provide an input file with the single cell genotype probability (using the format as specified above).

▷ `./scistree <genotype-probability-file>`

By default, ScisTree2 outputs the inferred cell tree (in the Newick format) only. In order to output the called genotypes, you should specify the “-v” option:

▷ `./scistree -v <genotype-probability-file>`

ScisTree2 outputs the imputed genotypes from genotype probability. For reference, it first outputs the maximal probability genotypes (taking the most probable genotype at each position in the matrix). Then ScisTree2 shows the genotypes that are changed from the maximal probability genotypes. The called genotypes are shown below “Imputed genotypes.”. ScisTree infers a cell tree without branch length.

ScisTree has also implemented several additional functionalities. See the following for more details.

2.2.1 How to calculate genotype probabilities from input?

There are two basic scenarios on the preparation of genotype probabilities input.

1. Sometimes, there are called genotypes (say 0/1) where there are some uncertainties about genotypes. Here, one only has an estimate of global error rates: $p_{0 \rightarrow 1}$ (probability of mistaking a 0 to 1) and $p_{1 \rightarrow 0}$ (probability of mistaking a 1 to 0). Then, the probability of genotypes can be computed based on the genotype and the corresponding error rate. That is, if the called genotype is 0, then its genotype probability is $1 - p_{1 \rightarrow 0}$; if the called genotype is 1, then its genotype probability is $p_{0 \rightarrow 1}$. This uniform error rate scenario is similar to those considered by the program SCITE and SiFit.

2. The main motivation of ScisTree is handling non-uniform error rate scenario. The most common situation is that we have sequence reads from single cell DNA sequencing. In this case, probability of individual genotypes is determined by the sequence reads. There are various ways of obtaining genotype probabilities from sequence reads. (i) One may use the program Monovar, which calls single cell genotypes from sequence reads. Monovar can output the genotype probabilities in the VCF format, and (ii) alternatively, you can use a customized program written by myself that converts the single cell read counts to genotype probabilities. More details on converting sequence data to probabilities are given in the Appendix.

2.3 Command line options

For ease of reference, I now provide the list of (optional) command line options.

1. -T $\langle k \rangle$: use k threads for multithreading (parallelization). This can significantly speed up the running of ScisTree2 by using more than one thread. Note: multithreading was not supported in the original ScisTree.
2. -v: output more information about the results by ScisTree2. It outputs the called genotypes, along with genotypes called by simple single site maximal probability genotypes, difference between the two set of genotypes, and some additional information.
3. -n: only output the cell tree constructed by simple neighbor joining. In this case, neighbor joining is run with the maximal probable genotypes from single positions. This can be useful when one wants to find a quick cell tree for very large data.
4. -e: output a tree that is called mutation tree. Briefly, this tree may not be binary. Mutation tree is implied by the imputed genotypes. Mutation tree is meant to specify the ancestral relationships among mutations (i.e., site labels). If the mutation tree is very large, you may use “-e0” to output a mutation tree without mutation labels. This may help to visualize the tree better for large trees.
5. -q: perform NNI local search. By default, ScisTree2 uses SPR local search, which is more accurate than the NNI local search in our experiments. The user can choose to use NNI local search using this option.
6. -s $\langle m \rangle$: set the maximum number of iterations m . By default, the maximum number of iterations is 1,000. You can specify a smaller integer (e.g., 5) to reduce the run time. However, please note that there is a trade-off between accuracy and run time.

3 Revision History

1. July 30, 2025: Release of v2.2.1.0. Add a small option for bounding the number of iterations.
2. 10/22/2024: Release of v2.2.0.0. This is the base version for ScisTree2.