**Phylogenetic tree inference: a top-down approach to track tumor evolution**

**PTI - Usage Example**

Pin Wu

September 22, 2019

[1 Abstract 3](#_Toc16976)

[2 Getting started 3](#_Toc28635)

[2.1 Minimum requirements 3](#_Toc468)

[2.2 Installation 3](#_Toc2890)

[2.3 Workflow overview 3](#_Toc17203)

[3 Preparing inputs for PTI 4](#_Toc10057)

[3.1 One sample per file 4](#_Toc32133)

[3.2 Binary matrix 5](#_Toc16776)

[4 Output files： 5](#_Toc30410)

[4.1 Tree structure 5](#_Toc4818)

[4.2 auxiliary file 6](#_Toc3143)

# 1 Abstract

Genome instability is a prominent feature of intra-tumor heterogeneity. Using somatic mutations from multiple tumor samples from the same patient to infer the phylogenetic tree can provide insight into the origin of metastasis and dynamic process of tumor progression. Here, we describe a method, named PTI (Phylogenetic Tree Inference), which can exploit somatic mutations obtained from three or more samples of individual cancer patients to infer tree structure with high accuracy, in which the samples are placed as leaf nodes.

This document details a typical analysis of multiple samples of the same patient using PTI.

# 2 Getting started

### 2.1 Minimum requirements

• Software: Python

• Packages: NumPy、Biopython、Matplotlib

• Operating system: Linux, Windows

• Python version: 3.7.4

• NumPy version: 1.17.1

• Biopython version: 1.73

• Matplotlib version: 2.2.3

## 2.2 Installation

In order to install PTI, you can download the source code from https://github.com/bioliyezhang/PTI and put it in your path.

## 2.3 Workflow overview

A typical workflow developed with PTI is structured as follows:

1. Defining the intersection of somatic mutations in all samples from the same patient as the length of the root trunk of the tree structure;
2. Remove the shared mutations, and then find the optimal branch split;
3. Remove the mutation information of the sample that has been the leaf node.

Then, iterate through the above steps until all samples are split into leaf nodes.

1. Each tree structure is weighted by a score.
2. Output the tree structure with the largest weight score.

Understanding the time of occurrence and the distribution of driver mutations in different samples is important to understand the evolution of tumor progression. Therefore, our method also annotates the putative 299 driver genes on the branches of the tree for downstream analysis. In addition, for each patient, only the phylogenetic tree with the largest weight score is optimal, and its structural information will be output. If a patient has two or more tree structures with the largest weight score, then all the optimal tree structures will be output.

# 3 Preparing inputs for PTI

PTI can accept two input file types, one is ‘one sample per file’, the other is ‘binomial matrix’, also known as ‘0-1 matrix’. The number of mutations used to reconstruct the phylogenetic tree has a significant impact on the accuracy of the tree structure. We recommend that the number of mutations used to build a tree for each tumor sample is no less than 30. However, while a large number of mutations increase the accuracy of the tree topology, it takes more runtime. Therefore, in the case of a large number of mutations, you can use the AF parameters (default is 0.1) which is calculated by the count of mutant reads and reference reads to filter the mutations to improve the speed of the PTI.

## 3.1 One sample per file

See: [https://github.com/bioliyezhang/PTI/tree/master/Demo/Test2](https://github.com/Bio-wupin/PTI/tree/master/Demo/Test2)

**Command line:**

**$ python PTI.py --AF 0.1 -i input\_dir -o output\_dir**

Parameters:

--AF: The allele frequency of mutation. (The default is 0.1)

-i, --input: The path of the input folder

-o, --output: The path of the output folder

## 3.2 Binary matrix

Each patient only needs one binary matrix and its elements can only be 0 or 1. The first column of the file records unique mutation id. The middle columns record whether the sample has a certain mutation. If the sample has a certain mutation, it is recorded as 1, otherwise it is recorded as 0. The last column is about the genes involved in the mutation. Since the binomial matrix does not contain the count of mutant reads and reference reads, the PTI command line does not contain restrictions on AF.

See: [https://github.com/bioliyezhang/PTI/tree/master/Demo/Test1](https://github.com/Bio-wupin/PTI/tree/master/Demo/Test1)

**Command line:**

**$ python PTI.py -i input\_dir -o output\_dir**

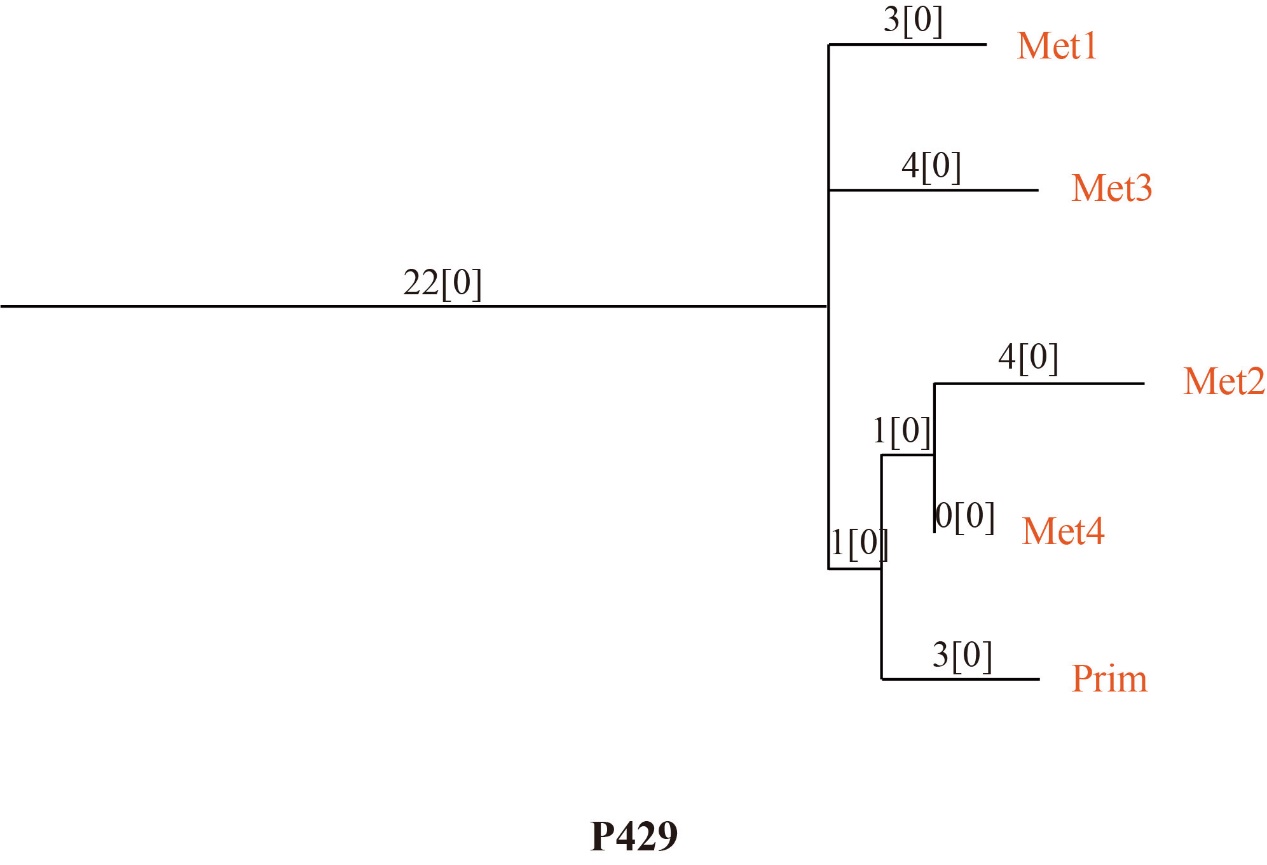
# 4 Output files：

This method will output all phylogenetic trees with the largest weight score, but in most cases each patient has a unique optimal tree structure and corresponding auxiliary file.

## 4.1 Tree structure

This file named “Patient-id\_tree\_0.pdf” contains the patient's optimal phylogenetic tree, which is a rooted tree. In the tree structure, all samples of the patient are placed as leaf nodes. The tree branches are labeled with the branch length, the number of driver genes and gene names involved in somatic mutations. The length of the tree branches represents the number of shared somatic mutations. If a patient has two or more tree structures with greatest weight score, each tree structure file will be named in the format “\*\_tree\_[num].pdf”, .

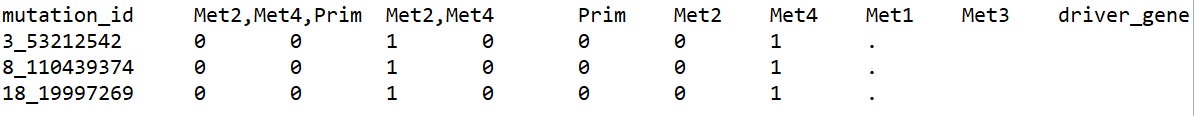
For example: **P429\_tree\_0.txt**



## 4.2 auxiliary file

This file named “Patient-id\_info\_[num].txt”, contains somatic mutations that have occurred in two or more branches other than the root trunk in the tree structure. The middle columns in the file represent all tree branches except the root node. If the mutation occurs in a branch, it will be recorded as 1, otherwise it will be recorded as 0. The last column indicates whether the mutation involves a known driver gene. If the mutation involves a known driver gene, the gene name will be recorded, otherwise it will be empty.

For example: **P429\_info\_0.txt**

****