SpoTyping - GUI

1. SpoTyping is a software for predicting spoligotype from sequencing reads, complete genomic sequences and assembled contigs.
2. The GUI version makes use of the Python package Tkinter, which comes along with Python installation.
3. Linux and MAC and windows users who are farlimiar with the command lines are suggested to use the command line version.
4. This manual will be focused on using SpoTyping on windows without touching the command line.

Part I. Spoligotype prediction and SITVIT database query.

Prerequisites:

- Python2.7
- BLAST

Input:

1. Fastq file or pair-end fastq files
2. Fasta file of a complete genomic sequence or assembled contigs of an isolate

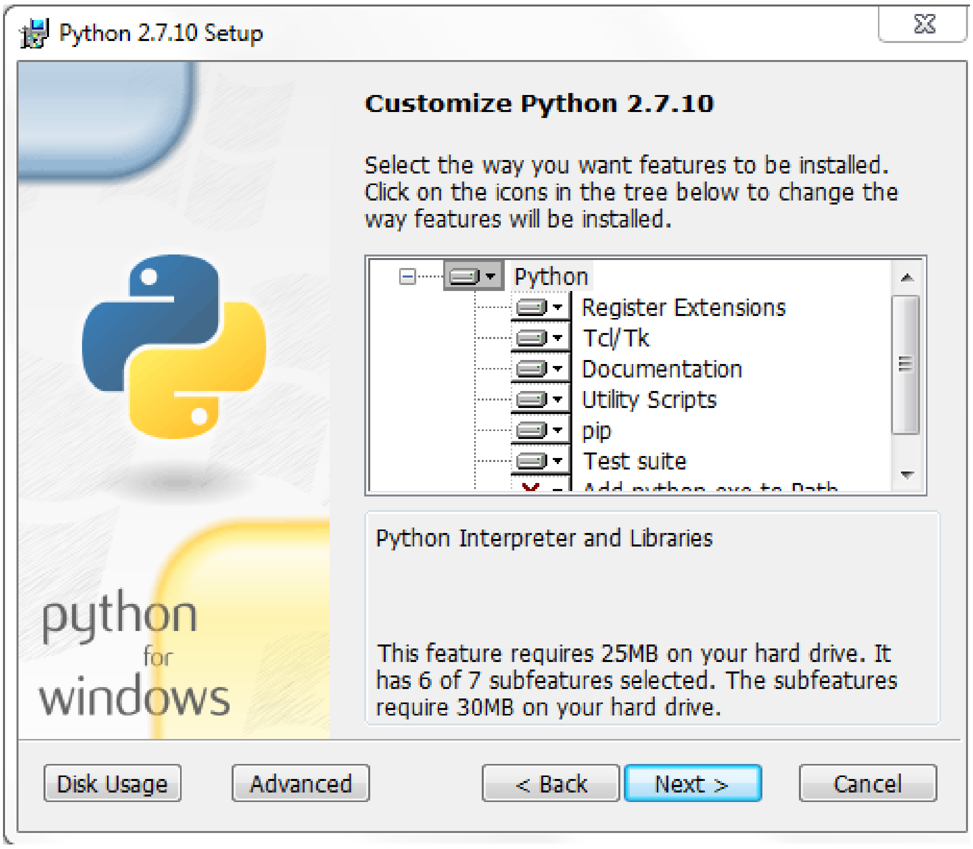
Output:

In the output file specified: predicted spoligotype in the format of binary code and octal code.
In the output log file: count of hits from BLAST result for each spacer sequence.
In the xls excel file: spoligotype query result downloaded from SITVIT WEB.

> Note: if the same spoligotype has been queried before and have an xls file in the output directory, it will not be queried again.

Installation of Python2.7 on windows

1. Download Python2.7 from the website: https://www.python.org/downloads/
2. During installation, enable the function of **‘Add python.exe to Path’** in ‘Customize Python 2.7.10’



Python 2.7.10 Setup

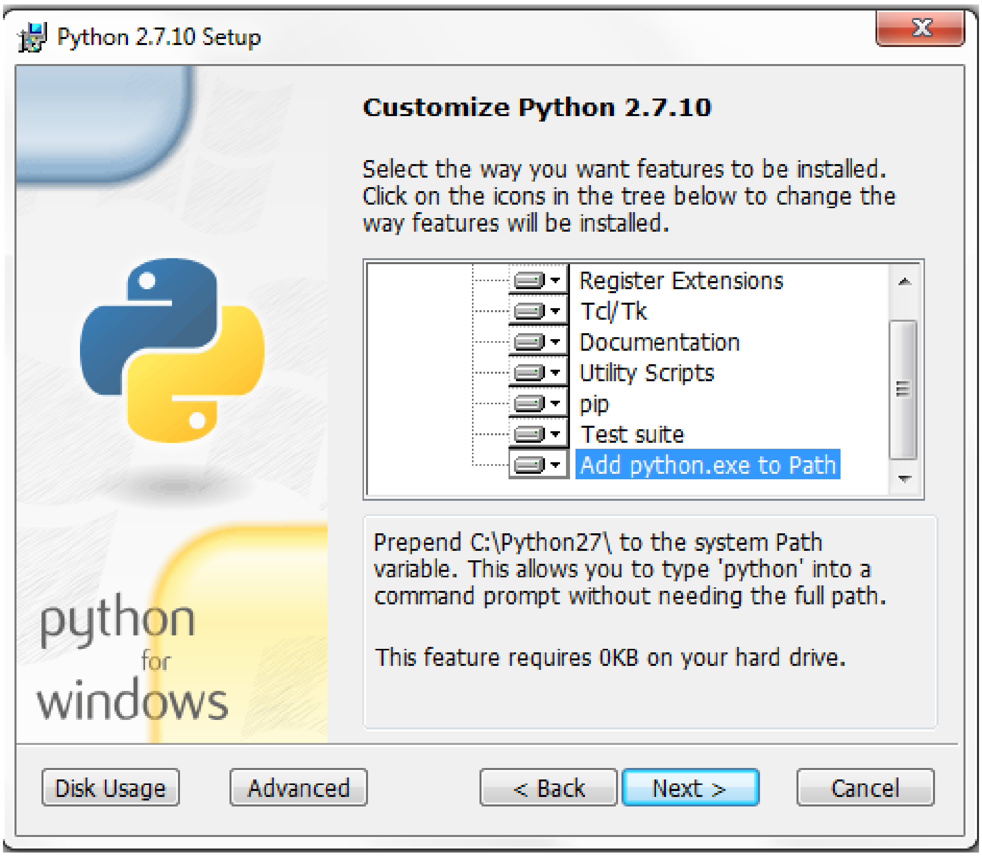
Customize Python 2.7.10

Select the way you want features to be installed. Click on the icons in the tree below to change the way features will be installed.

- Python
  - Register Extensions
  - Tcl/Tk
  - Documentation
  - Utility Scripts
  - pip
  - Test suite
  - Add python.exe to Path

Python Interpreter and Libraries

This feature requires 25MB on your hard drive. It has 6 of 7 subfeatures selected. The subfeatures require 30MB on your hard drive.



Python 2.7.10 Setup

Customize Python 2.7.10

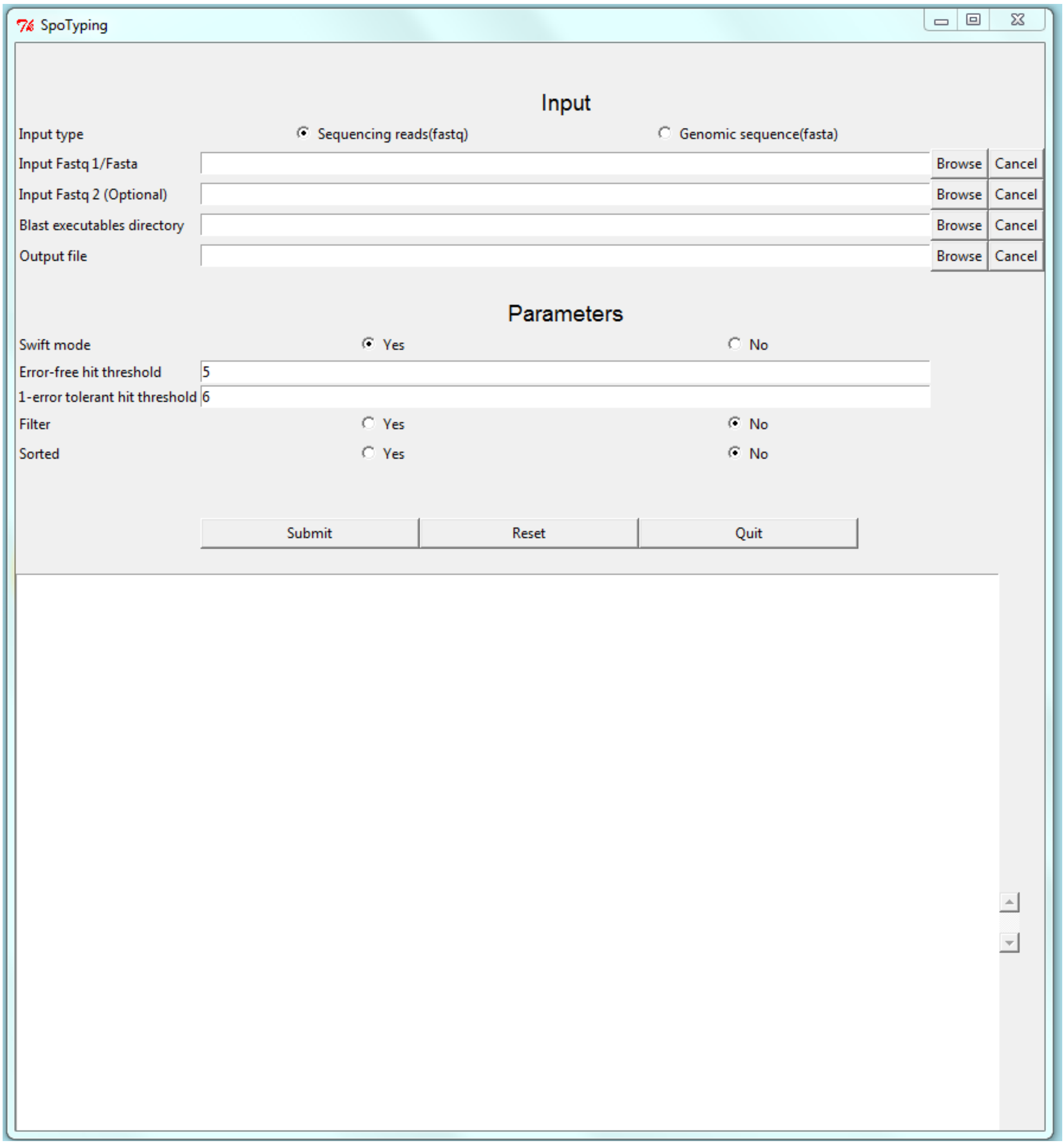
Select the way you want features to be installed. Click on the icons in the tree below to change the way features will be installed.

- Register Extensions
- Tcl/Tk
- Documentation
- Utility Scripts
- pip
- Test suite
- Add python.exe to Path

Prepend C:\Python27\ to the system Path variable. This allows you to type 'python' into a command prompt without needing the full path.

This feature requires 0KB on your hard drive.

Graphical User Interface:



SpoTyping

Input

Input type ● Sequencing reads(fastq) ○ Genomic sequence(fasta)

Input Fastq 1/Fasta Browse Cancel

Input Fastq 2 (Optional) Browse Cancel

Blast executables directory Browse Cancel

Output file Browse Cancel

Parameters

Swift mode ● Yes ○ No

Error-free hit threshold 5

1-error tolerant hit threshold 6

Filter ○ Yes ● No

Sorted ○ Yes ● No

Suggestions:

1. The fileds under the ‘Input’ section could be set based on the need.
2. There is no need to change the parameters if input is genomic sequence(fasta).
3. If input is sequencing reads(fastq), it’s highly suggested to use the default settings (including the swift mode).
4. Do wish to change the hit thresholds? Can adjust the thresholds to be 0.0180 to 0.1486 times the estimated read depth for error-free hits and 0.0180 to 0.1488 times the estimated read depth for 1-error-tolerant hits. (The read depth is estimated by dividing the sequencing throughput by 4,500,000, which is the estimated *Mtb* genome length.) The default setting already has this optimized.
5. For low quality sequence reads (reads with many ‘N’s or long homopolymers), please select

'Yes' for **Filter**.

6. If the reads are sorted against a reference genome (extracted form sorted bam files, for example), please select 'Yes' for **Sorted**.

SpoTyping seems slow? (Not finished in 5 mins, for example)

- **Low quality of sequence reads?** (reads with many 'N's or long homopolymers): try toselect 'Yes' for **Filter**.

Got weird spoligotype prediction?

- **Reads are sorted against a reference genome?**: try toselect 'Yes' for **Sorted**.
- **Sequencing throughput is very low?** (<40Mbp, for example): SpoTyping may not be able to give accurate prediction due to the relatively low read depth.

Part II. Summary pie chart plot from the downloaded xls files.

Prerequisites:

- R
- R package: gdata

Input:

The xls file downloaded from SITVIT WEB.

Output:

A pdf file with the information in the xls file summarized with pie charts.

Usage:

Use the following functions in R.

```
library(gdata)
# pacakge gdata with function read.xls to parse xls files.

inXLS <- ""    ## Fill in here the input name
outPDF <- ""    ## Fill in here the output name

data <- read.xls(inXLS)
data <- as.matrix(data)

```

```
pdf(file=outPDF)
for(i in 5:13){
    content <- data[(data[,i]!="" & !is.na(data[,i])),i]
    inpie <- table(content)
    pie(inpie,main=colnames(data)[i],cex=0.5)
    mtext(paste("Number of records:",length(content),sep=""), side=1, line=1)
}
dev.off()
```