

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 familiar 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.

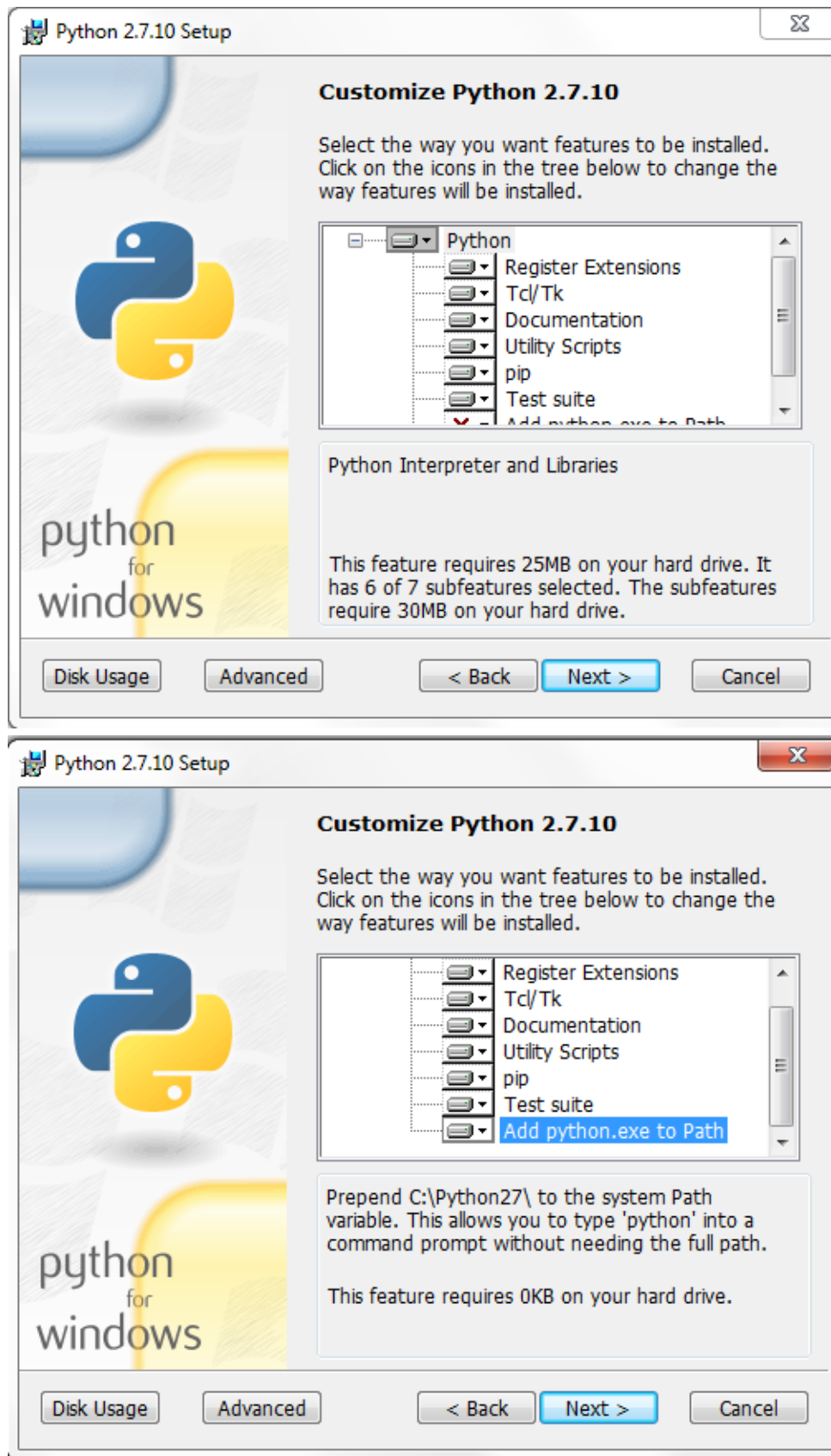
In the output log file: count of hits from BLAST result for each spacer sequence.

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'

**Graphical User Interface:**

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

1-error tolerant hit threshold

Submit Reset Quit

Suggestions:

1. The files 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 swift mode with the default parameters if the sequencing throughput is no less than 135Mbp.
4. For sequencing experiments with throughputs below 135Mbp, please 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.)
5. If you do wish to disable the swift mode for sequencing experiments with throughputs above 1260Mbp, please 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.

Got weird spoligotype prediction?

- **Sequencing throughput is very high** (>1000Mbp, for example): try to disable the swift mode and set the hit thresholds higher (10% of the estimated read depth, for example).

```
1. # Example commad:
2. python SpoTyping.py -s off -m 10 -r 12 read_1.fastq.gz read_2.fastq.gz
```

- **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.

```
1. library(gdata)
2. # pacakge gdata with function read.xls to parse xls files.
3.
4. inXLS <- ""      ## Fill in here the input name
5. outPDF <- ""     ## Fill in here the output name
6.
7. data <- read.xls(inXLS)
8. data <- as.matrix(data)
9.
10. pdf(file=outPDF)
11. for(i in 5:13){
12.   content <- data[(data[,i]!=" " & !is.na(data[,i])),i]
13.   inpie <- table(content)
14.   pie(inpie,main=colnames(data)[i],cex=0.5)
15.   mtext(paste("Number of records:",length(content),sep=""), side=1, line=1)
16. }
17. dev.off()
```