Supporting Text S2

PhyloPythiaS+: A Self-Training Method for the Rapid Reconstruction of Low-Ranking Taxonomic Bins from Metagenomes

Installation instructions

This document describes how to install the VM release of the *PhyloPythiaS+* (*PPS+*) version 1.4. The installation under Ubuntu without the VM is described as well.

Mailing list

PhyloPythiaS@uni-duesseldorf.de

Home page

https://github.com/algbioi/ppsp/wiki

Method Overview

PPS+ is software for the automated and accurate taxonomic binning of a metagenome sequence sample to low ranking taxonomic bins. The software employs marker gene classification combined with composition based taxonomic binning method named *PhyloPythiaS*. The software works in two steps. In the first step, the input FASTA file containing DNA sequences is processed employing the marker gene analysis. The marker gene analysis returns a list of taxa that were identified to be present in the sample and respective sample-derived sequences assigned to these taxa. In the second step, the composition based taxonomic binning method *PhyloPythiaS* is used to build models based on the list of taxa, the sample specific data and reference data from the public databases. These models are consequently used to assign taxonomic identifiers to the metagenome sample sequences.

Hardware Requirements

The VM runs on any 64bit operating system on which the 64bit *Oracle VirtualBox* can be installed. It requires at least 4GB RAM (i.e. 2GB for the VM, however 8-16GB RAM are

recommended) and 50-80GB disk space (80-110GB recommended, depending on the reference and sample used). We also strongly recommend using SSD storage.
The VM was tested on the following HW configurations:

- OS X version 10.9; 4GB RAM; Intel i5 2557M 1.7GHz, SSD storage.
- Windows 7 64bit and Ubuntu 12.04 64bit; 4GB RAM; Intel i5 M520 2.4GHz; HDD 7200 rpm
- Windows 8.1; 4GB RAM; Intel i5-3230M 2.60GHz, HDD 7200 rpm

VM Installation - How to set up the virtual machine on your computer

To set up the VM, follow these steps:

1) Install the 64bit *Oracle VirtualBox*, choose the appropriate installation file here:
http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html

2) Download the image of the virtual machine from:
http://algbio.cs.uni-duesseldorf.de/software/ppsp/1_4/ppsp_1_4_vm_64bit.ova
(You can verify that the downloaded file is not corrupted by computing the md5 checksum of the file that must be: "d52377a9b8f4ef4291aaa9718d614936")

3) Import the VM image into the *Oracle VirtualBox* (usually, double click the *ppsp_1_4_vm_64bit.ova* file to import the VM). After the import finishes, the *Oracle VM VirtualBox Manager* displays the imported VM with name *hhu_vm_ppsp_64bit* in the list of all virtual machines. Click on *hhu_vm_ppsp_64bit*, the settings of the virtual machine will be displayed on the right-hand side. Here, you can modify the properties of the VM. Under *System*, you can increase the *Base Memory*, which is the amount of the main memory given to the VM, e.g. you can give 50% of the physical main memory to the VM. Under *Shared folders*, edit the shared folder, s.t. the *Folder Path* is the path to the directory that will later contain all reference data, set the *Folder Name* to *host_shared*. Note that the shared folder may not support all Ubuntu file system operations, e.g. creating of symbolic links on Windows host file systems. The disk that contains this shared folder should have at least 25-55GB free space (depending on the reference used).

4) Start the VM (click on the *hhu_vm_ppsp_64bit* and then on *Start*)

5) Open the Terminal window and press enter
(Note that the password for *user1* is an empty string.)

6) Install the latest version of the system tools, type and press enter:

update -s

7) Install the latest version of the *PPS+* package, type and press enter:
update -t

8) Install the reference data, depending on which version of the reference you want to use, type one of the commands or both) and press enter:
update -r NCBI20121122
update -r NCBI20140513

The update command takes a name of reference data and installs it. After the script finishes, the reference data can be used. Note that this step is time consuming and can take up to several hours depending on your HW and internet connection, it downloads and decompresses a file of several GB.

9) Optional: To verify that the VM is set up correctly, you can run the following test, that runs the whole *PPS+* pipeline with a small dataset, according to the reference data installed, type one of the commands and press enter:
time ppsp -c /apps/pps/tools/config_ppsp_vm_refNCBI20121122_example.cfg -n -g -o s16 mg -t -p c -r -s
or
time ppsp -c /apps/pps/tools/config_ppsp_vm_refNCBI20140513_example.cfg -n -g -o s16 mg -t -p c -r -s

The results will be in /apps/pps/tests/test01/output or /apps/pps/tests/test02/output, respectively; this test will take approx. 2hours, depending on your HW.

License and source code

The additions of the *PhyloPythiaS+* are distributed under GNU GPL v 3 license and the source code can be accessed via:
https://github.com/algbioi/kmer_counting
https://github.com/algbioi/ppsplus

The original *PhyloPythiaS* and the other tools keep their licenses, as described in:
/apps/pps/tools/LICENSE.txt

Disclaimer

This software is distributed as it is without any warranty. Please tell us about any errors you encounter, problems with the installation, or suggestions how to make the release better. The *PhyloPythiaS+* release contains several third-party components that may cause problems that we may not be able to fix ourselves promptly.

The VM directory structure

/apps/pps/tools … contains all the tools and scripts required by *PPS+*

/apps/pps/samples … usually contains all the samples (FASTA files) to be analyzed

/apps/pps/samples/test_3strains … contains a sample FASTA file (*.fna) containing simulated contigs and corresponding labels (*.tax)

/apps/pps/tests … store all the analysis results here

/apps/pps/tools/config_ppsp_vm_refNCBI20121122_example.cfg … sample configuration file when using reference NCBI20121122

/apps/pps/tools/config_ppsp_vm_refNCBI20140513_example.cfg … sample configuration file when using reference NCBI20140513

/mnt/host_shared … is the mounted shared folder that contains the reference data, you can exchange files with the host operating system using this folder

Sample analysis configuration – how to prepare and configure one metagenome sample analysis

For more information, see provided tutorial:
/apps/pps/tools/ppsp_vm_tutorial.pdf

A) Create an empty directory in the /apps/pps/tests folder, that will contain all the analysis results and temporary files (e.g. test01 and test02 are used for the sample dataset)

B) Create a configuration file, e.g. copy and modify file /apps/pps/tools/config_ppsp_vm_refNCBI20121122_example.cfg or file /apps/pps/tools/config_ppsp_vm_refNCBI20140513_example.cfg depending on the reference you want to use. Note that all the paths in the configuration file must be absolute paths.

First, set the *pipelineDir* to the directory created in step A.

Section INPUT FILES contains all the input files, where only the entry inputFastaFile is mandatory; it contains the sequences you want to classify. However, it is recommended to specify all the input files if available.

Section REFERENCE contains paths to the reference data; you can use this setting for all the samples you classify. The reference data is stored in the shared folder.

Section TOOLS contains paths to the tools and scripts required by *PPS*+, you do not have to change these paths.

In the BASIC SETTINGS section, you can set the following: down to which taxonomic rank you want to assign the sequences and the maximum number of clades that you want to model. Note that it is recommended to consider only sequences of at least 1000bp length; and it is not recommended to change anything in the ADVANCED SETTINGS.

Run *PPS*+, the simple way

Here, you can find a list of basic commands that can be run from a command line of the VM.

```
# To list all the options of the main script, type:
ppsp -h

# To run the whole pipeline employing the marker gene analysis (if scaffolds are available, instead of "-p c" type "-p c s v")
ppsp -c CONFIGURATION_FILE -n -g -o s16 mg -t -p c -r -s

# To get predictions only based on the marker gene analysis (this step does not run PhyloPythiaS), check the configuration parameter minSeqLen!
ppsp -c CONFIGURATION_FILE -n -g -o s16 mg -s

# To run the whole pipeline using the general model for the most maxLeafClades abundant clades, at a given taxonomic rank, in the reference (if scaffolds are available, instead of "-p c" type "-p c s v")
ppsp -c CONFIGURATION_FILE -o general -t -p c -r -s
```

Run *PPS+*, the harder way

This approach enables to modify the list of clades (taxa) or to use expert sample-derived (sample-specific) data to train *PhyloPythiaS* (note that this option has not been tested extensively).

To run the marker gene analysis and prepare the output for *PhyloPythiaS*, run:
ppsp -c CONFIGURATION_FILE -n -g -o s16 mg

Now, file pipelineDir/working/ncbids.txt (where pipelineDir is the directory set in the configuration file) contains all the leaf level clades (taxa) to be modeled by *PhyloPythiaS*. You can delete some of the clades (taxa) or add new ones, but make sure that the file contains only leaf level clades (taxa).

You can also change which sample-derived data will be used to build the models. The directory pipelineDir/working/sampleSpecificDir contains all the sample-derived data found via the marker gene analysis. Each FASTA file contains sample specific data for NCBI taxon id denoted by the prefix of the file name (e.g. a file named 126.1.fna contains sample-derived data for NCBI taxon id 126). If you want to add more (expert) sample-derived (sample-specific) data, create a FASTA file with an appropriate name in that directory. Note, that it is possible to have more than one FASTA file containing reference sequences for one NCBI taxon id, e.g. you can name such files as: 126.1.fna, 126.2.fna, 126.3.fna, etc.)

Note, that the list of clades (ncbids.txt) and the content of the sample-derived directory (sampleSpecificDir) must be consistent. Also note, that it is sufficient to have at least 100 kb of sample-derived data to model a clade (taxon).

Run the rest of the pipeline with the modified list of clades or sample-derived data. (In the case scaffolds are available, instead of "-p c" type "-p c s v".) Run:
ppsp -c CONFIGURATION_FILE -t -p c -r –s

Test Datasets

Two real and two simulated datasets are provided for testing under:
https://github.com/algbioi/datasets

Output files

pipelineDir/output … contains all important output files

pipelineDir/output/inputFastaFile.fna.pOUT … tab separated file containing assigned sequences, first column ~ sequence name, second column ~ NCBI taxon id

pipelineDir/output/inputFastaFile.fna.PP.pOUT … file containing assignments in the *PhyloPythia* format

pipelineDir/output/summary_train.txt … a list of clades generated by the marker gene analysis that contains all the clades that will be modeled, 1st column ~ sample-derived data in terms of bp, 2nd column ~ sample-derived data in terms of the number of sequences, 3rd column ~ scientific name (NCBI taxon id) of a clade at rank superkingdom, 4th column ~ corresponding phylum rank, etc. (Note that this file is a result of the marker gene analysis and does not reflect any change that you did manually to the list of clades or the sample-derived data)

pipelineDir/output/summary_all.txt … a list of all clades for which some sample-derived data was found (not all clades from this list are modeled)

pipelineDir/output/inputFastaFile.fna.cons … the scaffold contig consistency file (the consistency is computed based on the scaffoldsToContigsMapFile from the configuration file)

pipelineDir/output/precision_recall.csv … precision and recall computed at different taxonomic ranks considering referencePlacementFileOut set in the configuration file as the true assignments of the sequences (see also entries recallMinFracClade and precisionMinFracPred from the configuration file)

pipelineDir/output/precision_recall_no_ssd.csv … precision and recall computed only based on the input sequences different from the sample-derived data

pipelineDir/output/no_ssd.fas … contains sequences from the inputFastaFile without the sample-derived data

pipelineDir/output/cmp_ref … contains comparison tables for different taxonomic ranks, where rows correspond to the true assignments (referencePlacementFileOut from the configuration file) and columns correspond to the assignments by *PPS+* (i.e. ideally, all the data lie on the diagonal)

pipelineDir/output/train_accuracy … contains precision and recall (as well as comparison tables) for different training data types where the training data are considered

as true assignments (this data is available after you run the pipeline after training using option "-a")

pipelineDir/output/log … contains log files for different subroutines (e.g. *PPS* train, *PPS* predict)

pipelineDir/output/contigs_vs_scaff … comparison of the contig and scaffold assignments (if run with "-p c s v" and inputFastaScaffoldsFile and scaffoldsToContigsMapFile were set in the configuration file; assignments of the scaffolds correspond to the rows and assignments of the contigs correspond to the columns. (In the case the models are good, the data lie on the diagonal.)

Working files

pipelineDir/working … contains working/temporary files

pipelineDir/working/projectDir … is the *PhyloPythiaS* (*PPS*) working directory

pipelineDir/working/projectDir/models … after *PPS* training phase finishes, the models are stored in this directory and can be reused

pipelineDir/working/projectDir/sampled_fasta … FASTA files used to train *PPS* (this directory can be removed after *PPS* training phase finishes or after training accuracy is computed)

pipelineDir/working/projectDir/train_data … feature vectors generated from the sampled_fasta files used to train *PPS* (this directory can be removed after *PPS* training phase finishes)

pipelineDir/working/ncbids.txt … list of clades that is used to train *PPS* (i.e. to build *PPS* models)

pipelineDir/working/sampleSpecificDir … contains sample-derived data for *PPS*

pipelineDir/working/PPS_config_generated.txt … generated *PPS* configuration file

pipelineDir/working/train_accuracy … contains temporary files used to compute the training accuracy (this directory can be removed after the training accuracy is computed)

pipelineDir/working/crossVal … contains temporary files used to compare predictions of scaffolds vs predictions of contigs

*.sl … large files containing temporary data (feature vectors) that can be removed after the whole pipeline finishes

pipelineDir/working/*.ids … working input FASTA files where the sequence names were given working (temporary) sequence ids, where sequence id pattern [0-9]+_[0-9]+ corresponds to scaffoldID_contigID where the corresponding mapping (map: contigName -> contigID) is stored in file *.cToIds and the scaffold mapping (map: scaffoldName -> scaffoldID) is stored in file *.sToIds. The mapping (map: scaffoldId -> scaffoldId_contigId) is stored in file *.mapSCIds.

pipelineDir/working/*.ids.out … tab separated prediction file generated by *PPS*, first column contains sequence ids (the sequence ids correspond to the ids in file *.ids), last column contains NCBI taxon ids.

pipelineDir/working/*.ids.PP.out … prediction file in the *PhyloPythia* format generated by *PPS* (the sequence ids correspond to the ids in file *.ids)

pipelineDir/working/*.ids.ssd_cross … shows how the sample-derived data (i.e. data found by the marker gene analysis) were assigned for different clades (all computations are in terms of the number of sequences, not bp)

Frequently Asked Questions

Q: Command **update** returns an error message.
A: It is very likely that you are having network problems. You should try to run the command again after you resolve the network issue. Alternatively, you can download the sources manually.
For the **reference data**, download:
http://algbio.cs.uni-duesseldorf.de/software/ppsp/1_4/reference_NCBI20121122.tar.xz
or
http://algbio.cs.uni-duesseldorf.de/software/ppsp/1_4/reference_NCBI20140513.tar.xz

Copy the downloaded file to the shared directory host_shared, and decompress it there (e.g. under linux systems using command: tar -xJf REFERENCE.tar.xz), s.t. there will be the following paths to the reference resources afterwards:
host_shared/REFERENCE/mg3
host_shared/REFERENCE/sequences

host_shared/REFERENCE/silva111
host_shared/REFERENCE/taxonomy

For the **tools**, download:
http://algbio.cs.uni-duesseldorf.de/software/ppsp/1_4/tools.tar.xz
Copy it to folder: /apps/pps
Decompress it using command: tar -xJf tools.tar.xz

Q: How can I compute the **md5 checksum** of a file?
A: Linux: md5sum fileName
OS X: md5 fileName
Windows: E.g. use this program: www.winmd5.com

Q: When importing the VM, I am getting: "**VirtualBox - Error**, VT-x disabled in the BIOS".
A: Enable the Intel(R) - Virtualization Technology in the BIOS.

Q: The marker gene analysis **did not find all clades** that were expected in a sample.
A: You can try to lower the bootstrap cutoff of the Naive Bayes classifier, which is by default 80% (see configuration parameter: mothurClassifyParamOther; change "cutoff=80" e.g. to "cutoff=70" or "cutoff=60"). Note that by lowering the cutoff, the number of false assignments can increase.

Q: What is the **maximum length** of a DNA sequence in an input file?
A: The sequences should be shorter than 1 Mbp.

Ubuntu *PPS*+ installation without VM

This section describes, how to install *PPS*+ directly under Ubuntu 12.04 LTS, i.e. without the distributed virtual machine. Note that these instructions can differ under other Linux distributions.

Make sure that your installed RAM and swap space add up to at least 20 GB. If not, create an additional swap space to meet this requirement. This is mainly due to the requirements of the *mothur* software when running the pipeline with parameter "-n".

Create a directory under /mnt/host_shared. All reference data will be stored in this folder, e.g. create a symbolic link (using command: ln –s) with this name pointing to an arbitrary folder. This folder should be located on a large disk (~100 GB).

Create a directory /apps/pps/ in the same way as in the previous step. The folder should be located on a fast disk (~50 GB).

Create a symbolic link using command: sudo ln –s /apps /mnt/apps

Install the following packages (using command: sudo apt-get install PACKAGE_NAME):
python-biopython
ruby1.8
sqlite3
libsqlite3-dev
rubygems
ruby-sqlite3
libsqlite3-ruby1.8
p7zip-full
vim

Download the distribution package from:
https://github.com/algbioi/ppsp
Copy it to directory: /apps/pps
Decompress it, e.g. using command: 7za x distribution_file.7z
Copy the directories from the decompressed folder to /apps/pps, s.t. there are the following paths:
/apps/pps/samples
/apps/pps/sys
/apps/pps/tests
/apps/pps/tools
/apps/pps/ppsp_vm_tutorial.pdf

Setup PATH and PYTHONPATH:
export PATH=/apps/pps/sys:/apps/pps/tools:$PATH
export PYTHONPATH=/apps/pps/sys:/apps/pps/tools/ppsplus:$PYTHONPATH

Continue the installation instructions for the VM starting with step 6.
(Note that steps 6 and 7 can be skipped, but it is highly recommended to follow them to get the latest version of the software.)

Change log

2014/06/19 – version 1.4 released

2014/07/21 – FAQ updated
2014/10/01 – document refined, installation instructions for Ubuntu 12.04 added.