Taxonomic Assignments from Metagenomes

MetaPhlAn

Now that you have created a beautiful file structure in your working directory it is time to get to work ! One of the most fundamental questions in the field of microbial ecology is what sorts of things are in the community I am working in? Another equally important question is how do the things in environment A differ from environment B? In the microbiome world we frequently ask these questions. Here we will use publicly available data and open source tools to answer this very question for three microenvironments: the stool, the oral cavity, and the skin.

Objectives

1. Investigate the compositional microbiome structure at three unique body sites
2. Gain a working knowledge of how taxonomy is assigned to metagenomic datasets
3. Obtain experience working with third party software and publicly available data
4. Learn how to use simple `for` loops in a bash environment

Protocol

1. First lets set some paths so that our server knows where our software current lives. The default shell used by the CGRB is tcsh, however, these labs are written for a bash environment. So before you do anything else get into bash

```
$ bash
$ source
/nfs1/Teaching/CGRB/525_f15/local/bin/.metaphlan
```

   Here we enter a bash unix environment using the command bash. Next we 'source' the .metaphlan file that is in /nfs1/Teaching/CGRB/525_f15/local/bin/. The source command reads and executes the commands in the file that you used for input (/nfs1/Teaching/CGRB/525_f15/local/bin/.metaphlan). In this file there is a series of export commands which will set certain variables in our bash environment. In this case these variables are actually file paths that allow use to reduce typing and point the computer to certain directories that contain software we want to be able to access. Setting paths is an incredibly important feature of operating systems it allows you to do a great deal of customization, maintain multiple versions of software, and generally makes you life a lot easier. Don't worry if these seems a little abstract right now it will make sense soon.

2. Let's take a look at what has happened here.

```
$ vim
/nfs1/Teaching/CGRB/525_f15/local/bin/.metaphlan
```

Here we see that the we are assigning a path to the variable METAPHLAN. Now we know what's going on and we can exit vim. To exit type ':' the q! as below.

```
: q!
```

3. One last thing before we start. We need to make a directory for the results.

```
$ cd ~/metaphlan
$ mkdir metaphlan_out
```

4. The data for this experiment have been already downloaded for you and should be in your home directory in the raw_metagenome directory, but where did we find those data and how did we get them? We got these from the human microbiome project (HMP) data repository at http://hmpdacc.org. Let's try to download a file from the HMP website so you know how it's done. To do this we will use the wget command.

```
$ cd
$ mkdir checksum
$ cd checksum
$ wget
http://downloads.hmpdacc.org/data/Illumina/posterior_
fornix/SRS016111.tar.bz2
```

5. Something important (and somewhat common) occurred when I downloaded these files, one of them was corrupt. This can happen many ways so it is important that we check each file to ensure that it is not corrupt and is what we think it is before we process it. But, how can we tell if a file is corrupt? Luckily, for us the HMP has include checksums for all its files. A checksum is a string of characters that represent the correct number of digits in a file. Most unix based systems will have software to check checksums. On our system we have `md5sums`. Let's check the checksum for the file we just downloaded.

```
$ md5sum SRS016111.tar.bz2
#Should be 697001eaa1dab38debf563b7061a58f9
```

6. Now lets have a look at a corrupt file.

```
$ md5sum
/nfs1/Teaching/CGRB/525_f15/data/bad_md5/SRS011405.ta
r.bz2
#Should be d764dd5c06ea433c45075f970500edf5 but is
it???
```

You should always check the checksums before you unzip files (note we have done this for you) if the database provides you with checksums. Aside from data corruption what might also be a cause of getting a bad checksum? If you are interested the you can see all the checksums at http://hmpdacc.org/HMASM/.

Now onto the actual software!!

MetaPhlAn works by aligning short reads to a set of markers genes that are specific for individual clades of bacteria. In essence it aligns reads to this database of marker genes, assigns taxonomy and then counts up all the hits. This provides information about what taxa are present in our metagenome samples. As this assignment is based on a set of marker genes it is generally considered more reliable than 16S rRNA amplicon data.

7. We will run each of the publicly available metagenomes through metaphlan and then compare the results. Here your input.fastq file can be any file in your raw_metagenome folder.

```
$ cd ~/metaphlan
$ python $metaphlan_path <input.fastq> --bowtie2db
$METAPHLAN_DB --bt2_ps sensitive --nproc 1 --
bowtie2out metaphlan_out/<output.outfmt6.txt> -o
metaphlan_out/<output.profile>
```

Lets have a closer look at this command.

- `python`: This is the programming language that MetaPhlAn was written in and we can tell the computer to use a python interpreter to run this program. The interpreter will read and execute the command sequentially as they appear in the software.
- `$metaphlan_path`: This is a variable set in the .metaphlan file, and it holds the path to the metaphlan software.
- `<input.fastq>`: The file used as input into the software
- `--bowtie2db $METAPHLAN_DB`:The --bowtie2db flag sets the path to the BowTie2 database file and $METAPHLAN_DB is the path to the BowTie2

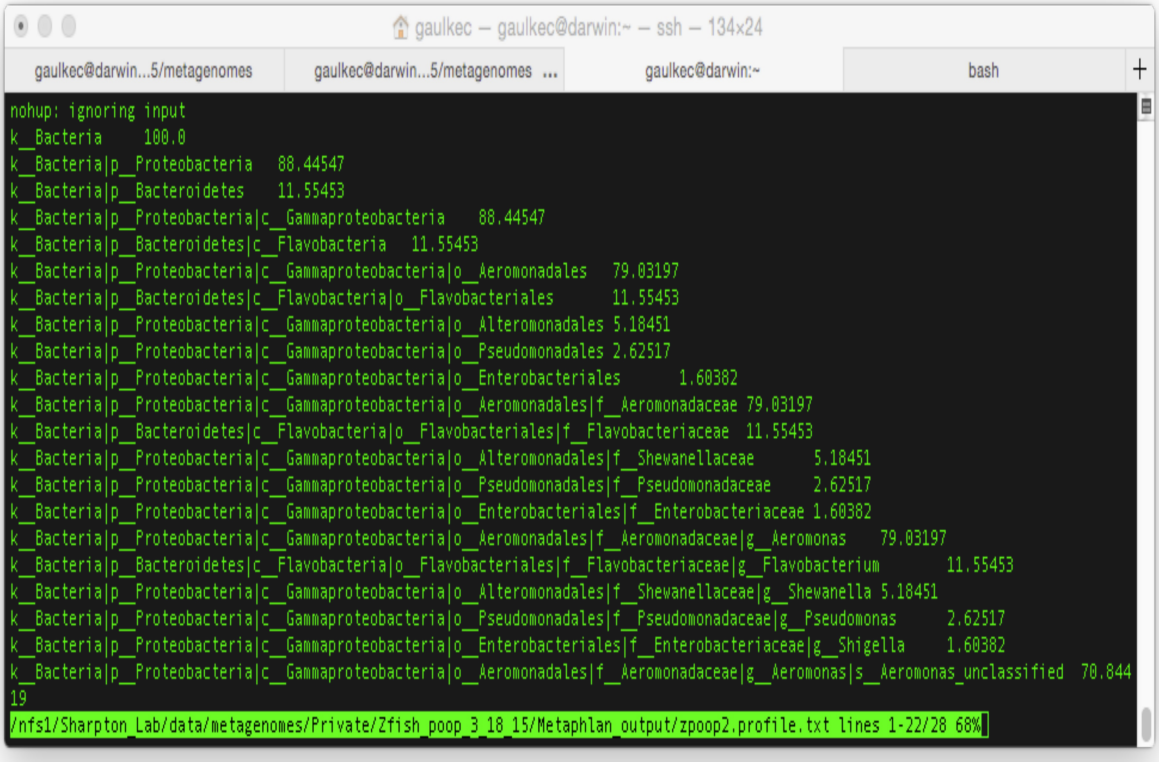
database file of the MetaPhlAn database

- `--bt2_ps`: This flag is used to indicate the BowTie2 presets, sensitive being the least sensitive and the fastest. The input to this option will control how sensitive and fast the alignments are.
- `--nproc`: Number of processors to be used for the command (never assume it will be 1, look into infernal's default processor allocation if you don't believe me)
- `--bowtie2out`: The file for saving the output of BowTie2
- `-o`: The output file that will contain taxonomic abundance information

8. This will take a few minutes to run on samples. After the job finishes take a look at the output.

```
$ less metaphlan_out/<output.profile>
```

The output should look something like this.



Here there are multiple levels of taxonomic assignment from phyla to species. Each level is split by the pipe character ('|'). While have all levels of taxonomy is informative it is possible that we just want to focus on one taxonomic level, perhaps pyhla.

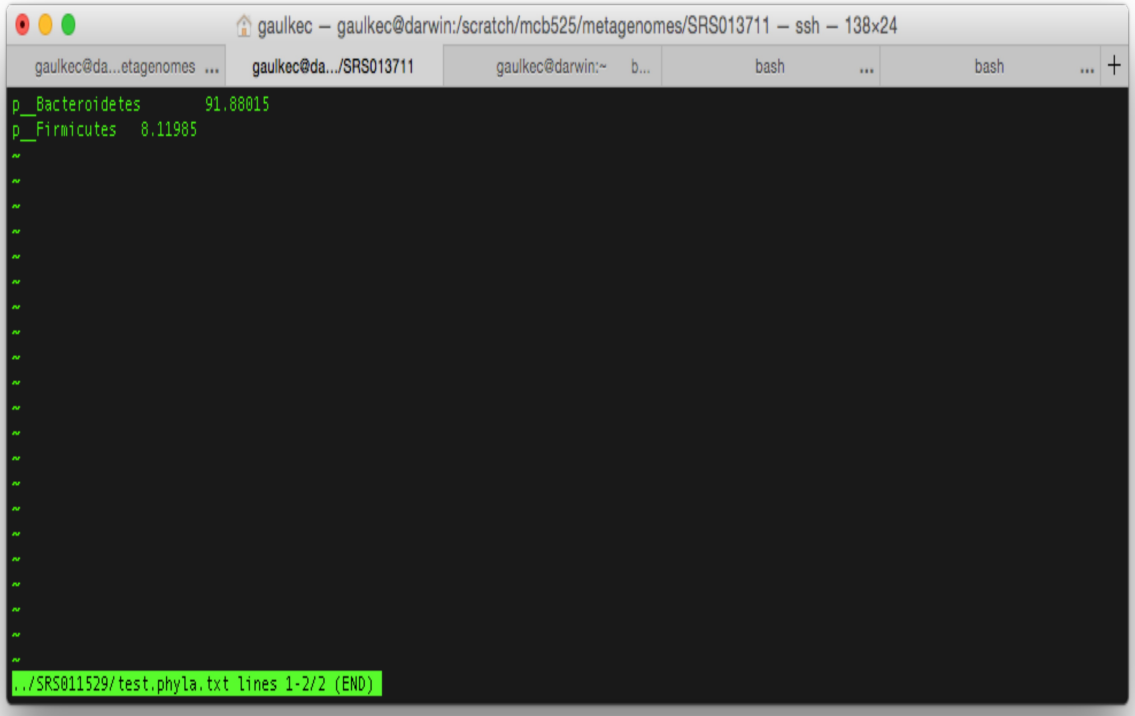
9. Now that we have run this through once we can rapidly reprocess the data at any taxonomic level that we want.

```
$ cd metaphlan_out
$ python $metaphlan_path --input_type bowtie2out
<input.outfmt6.txt> --tax_lev 'p' -o <out.phyla>
```

10. Let's have a look at this file.

```
$ less <out.phyla>
```

It should look something like this.



11. Now we only have phyla in the file which is useful for downstream analysis especially if you are only interested in difference at one level or another.

12. Let's finish running all of the files through MetaPhlAn. We could enter the command 15 times and it wouldn't really be that bad, but what if we had 1,500 files ? Luckily the command line is our friend here! First navigate to the directory containing all the raw metagenome files.

```
$ cd ~/raw_metagenomes
```

13. Gather a list of file names by assigning a variable.

```
$ my_files=$(ls *.fastq)
```

14. Here the variable $my_files gets a list of the contents of the directory. This allows us to run a simple `for loop`. Note here the '>' at the beginning of each line should not be typed.

```
$ cd ~/metaphlan/metaphlan_out
$ for f in ${my_files}
> do
> python $metaphlan_path
/nfs1/Teaching/home/<your_user_name>/raw_metagenome/$
{f} --bowtie2db $METAPHLAN_DB --bt2_ps sensitive --
nproc 1 --bowtie2out ${f}.outfmt6.txt -o
${f}.profile.txt
> done &
```

This might take a few minutes to run.

15. Now that we have run all these files through MetaPhlAn we can use MetaPhlAn to generate a heatmap of abundances.

```
$ cd ~/metaphlan
$ mkdir metaphlan_merged metaphlan_image
```

16. First we need to merge the output files from MetaPhlAn. This simply means that we will combine all of the different .profile files into one large table.

```
$ python
/local/cluster/metaphlan/utils/merge_metaphlan_tables
.py ~/metaphlan/metaphlan_out/*.profile.txt >
~/metaphlan/metaphlan_merged/merged_abundance_table.t
ab
```

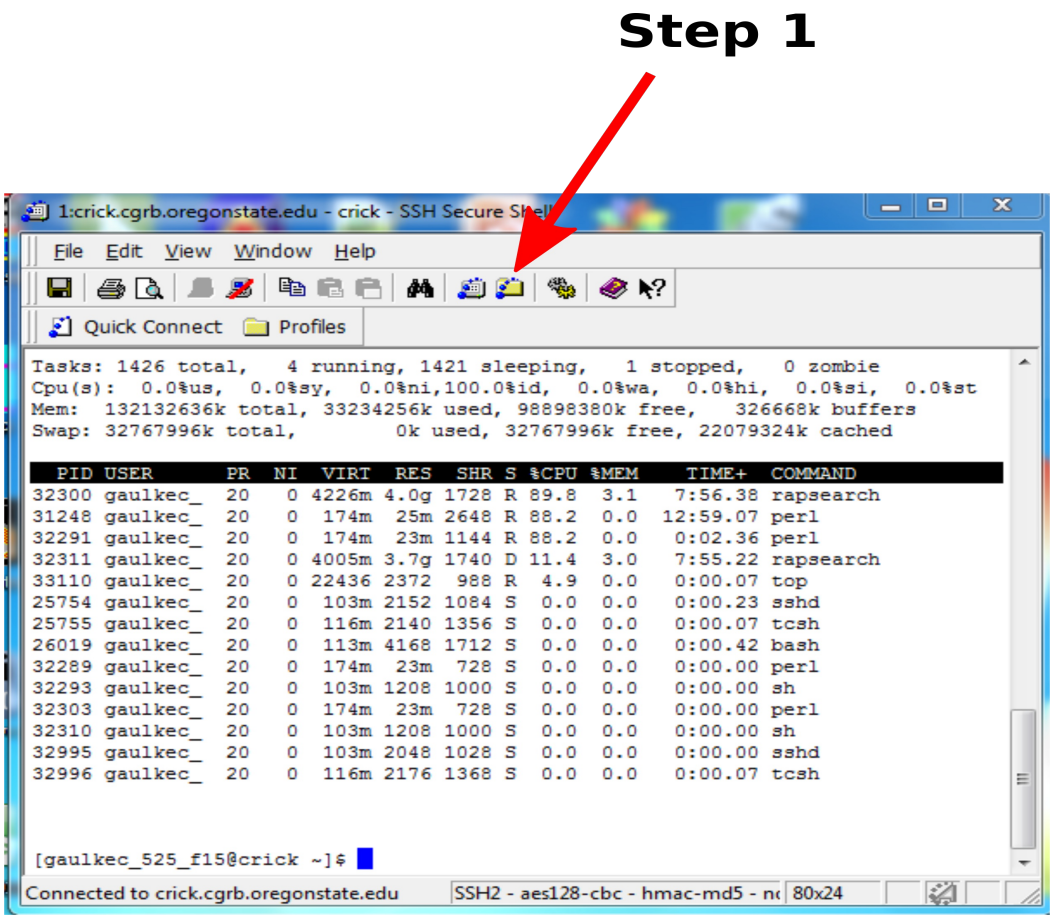
17. With the files all merged lets use the MetaPhlAn heatmap function to make the heatmap.

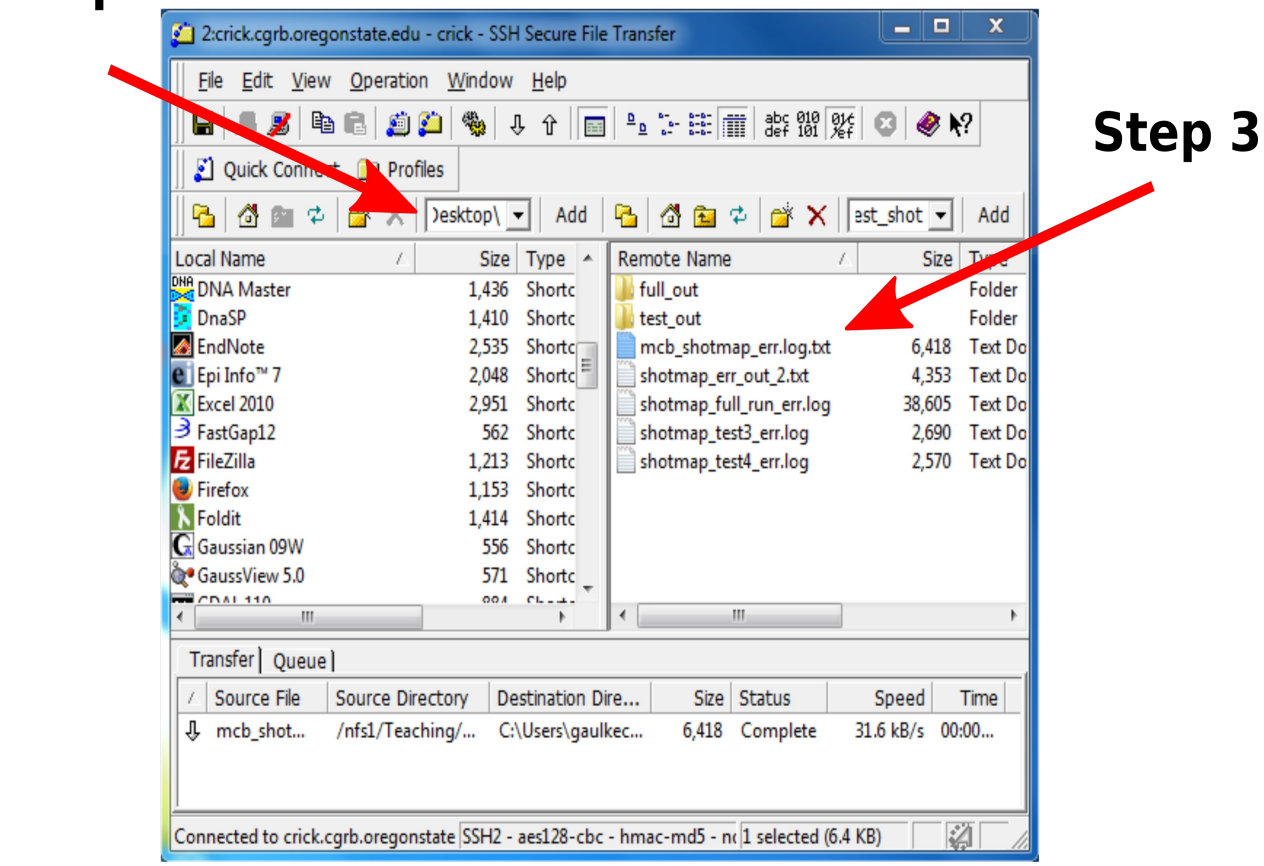
```
$ python
/local/cluster/metaphlan/plotting_scripts/metaphlan_h
clust_heatmap.py -c bbcry --top 25 --minv 0.1 -s log
--in
~/metaphlan/metaphlan_merged/merged_abundance_table.t
ab --out
~/metaphlan/metaphlan_image/abundance_heatmap.png
```

Let's have a look at some of these options

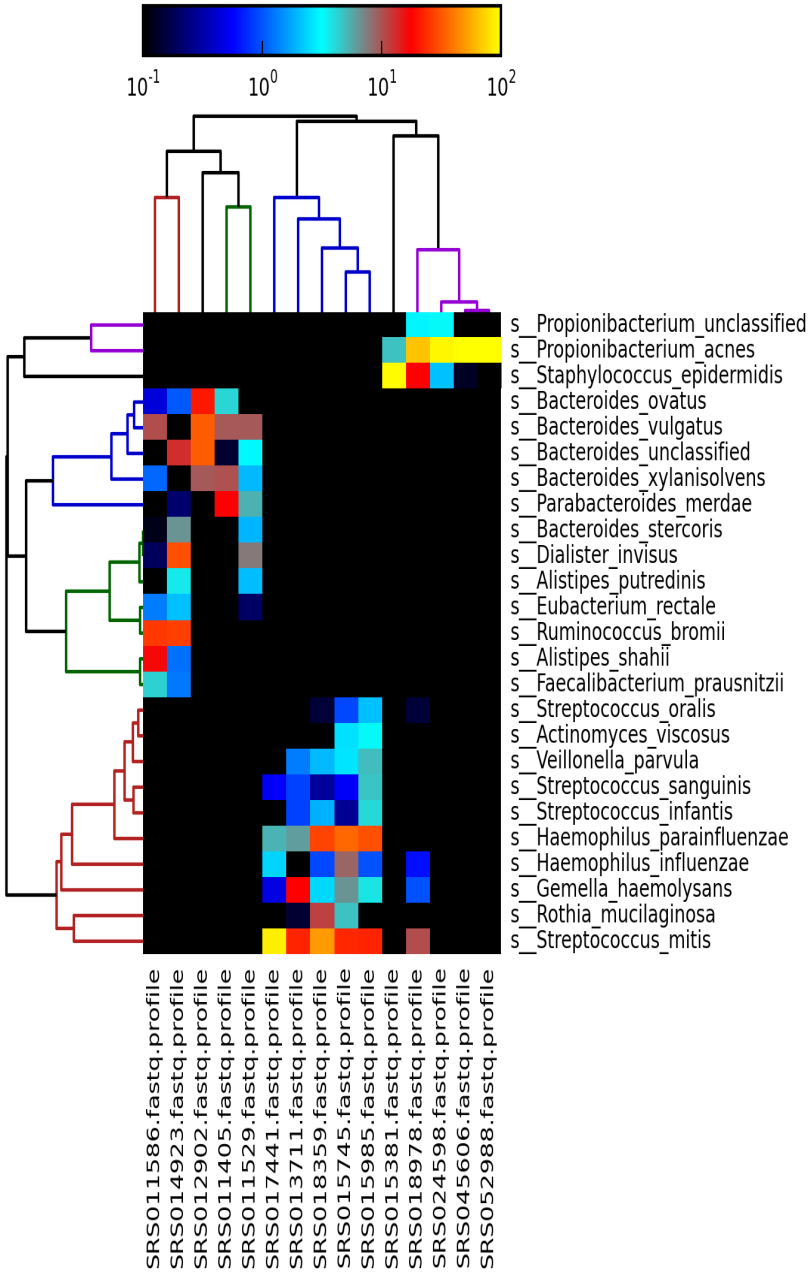
- `python`: This is the programming language that MetaPhlAn was written in and we can tell the computer to use a python interpreter to run this program. The interpreter will read and execute the command sequentially as they appear in the software.
- `/local/cluster/metaphlan/plotting_scripts/metaphlan_hclust_heatmap.py`: Path to the software.
- `-c`: The colors used for heatmap.
- `--top`: Tells the software to only use the top x number of taxa in the heatmap.
- `--minv`: The minimum abundance value to display.
- `-s`: The scale.
- `--in`: The file after this option is used as input into the software
- `--out`: The output file (heatmap).

18. Lets take a look at this image. To do this we will need to use our file transfer client. Follow the steps below to initiate the transfer
    - Step 1: Click on the open new file transfer window icon.
    - Step 2: Set the location you want to download the file.
    - Step 3: Select the file you want to download, right click and select download file.





Your heatmap should look similar (but perhaps not identical) to this.



These data suggest that the taxonomic structure of the microbiome varies across body site. This, as it turns out, is exactly what the human microbiome project found too. But, as it turns out it is not always easy to assign meaningful function to these differences, as taxonomy does not necessarily equal function. We will talk about this more in the functional metagenomics module.