



P.E.M.A.

a pipeline for eDNA metabarcoding analysis

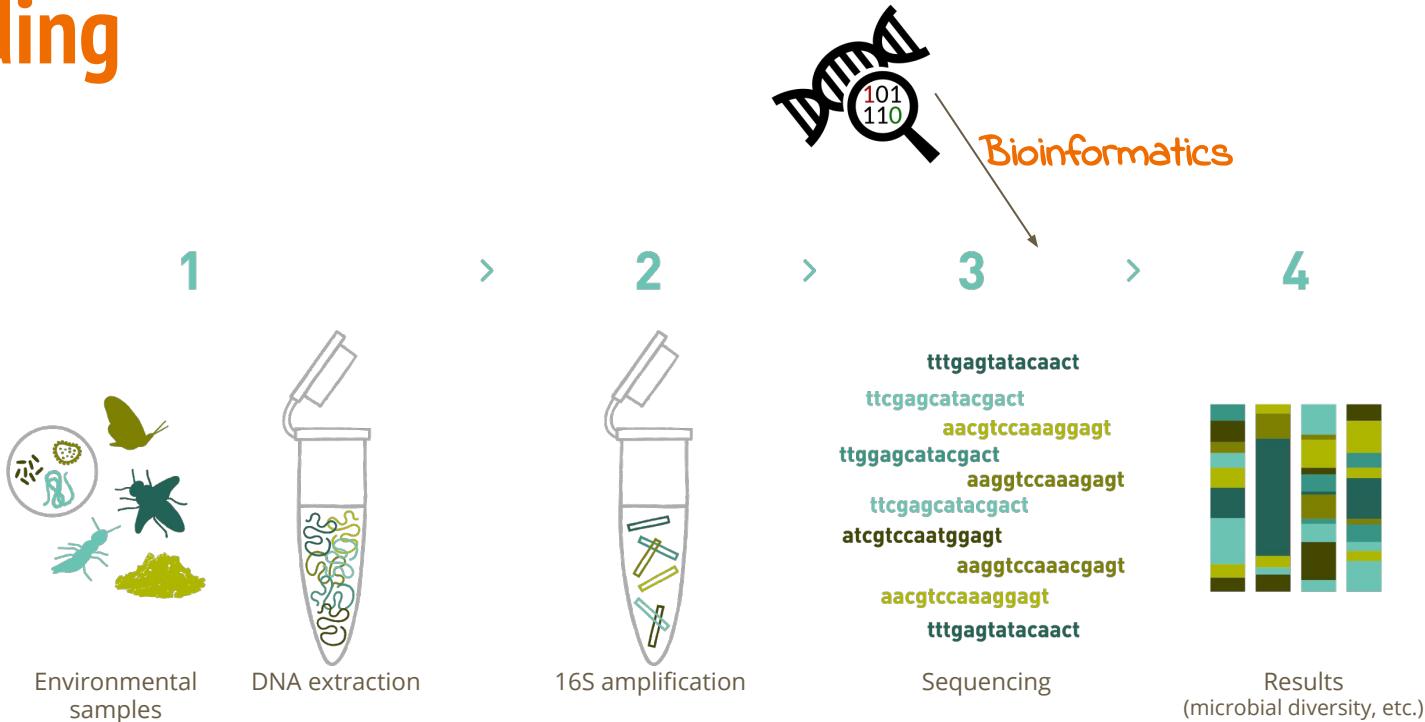
By:

Haris Zafeiropoulos

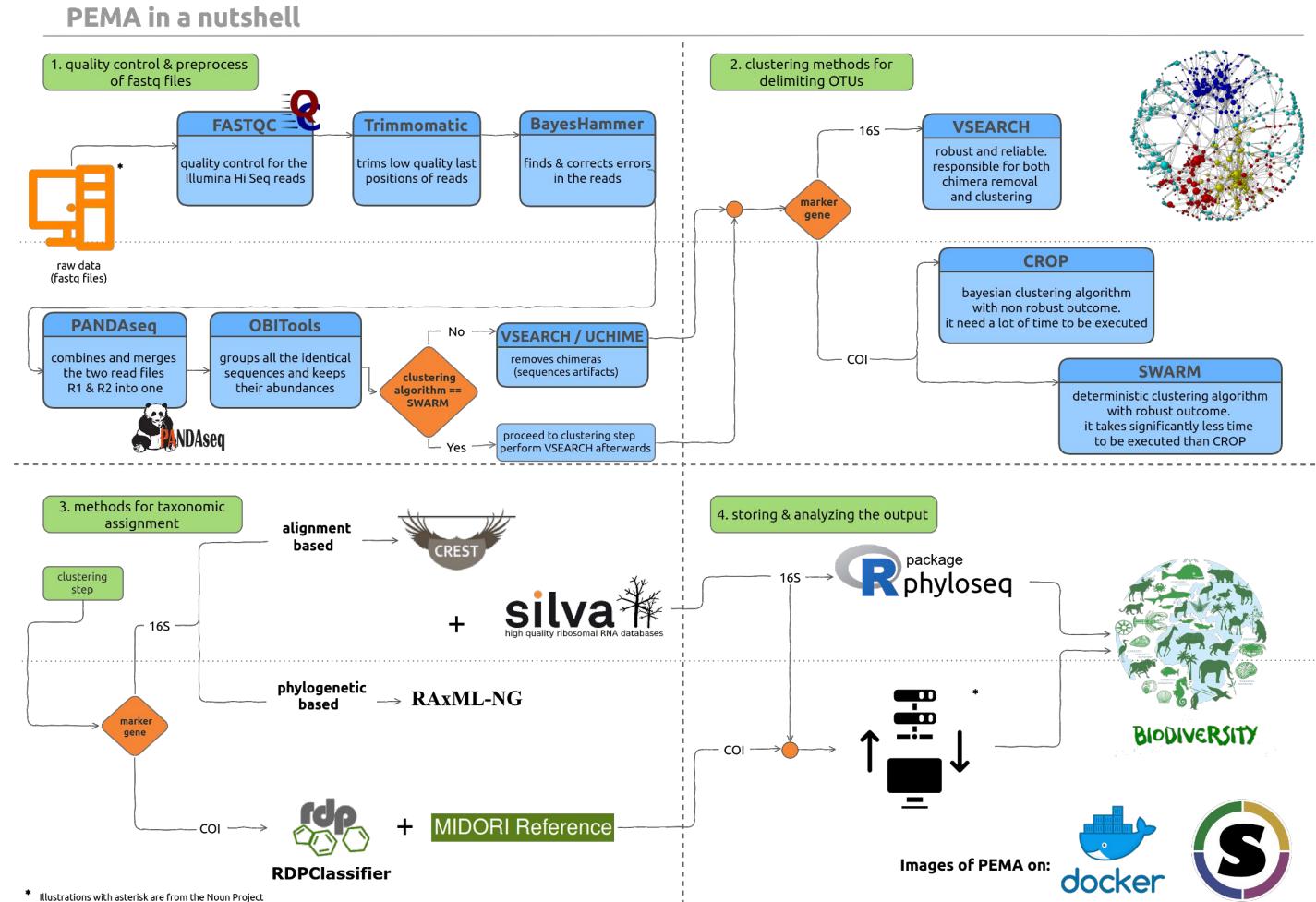
ink well by Daria Moskvina from the Noun Project

Metabarcoding

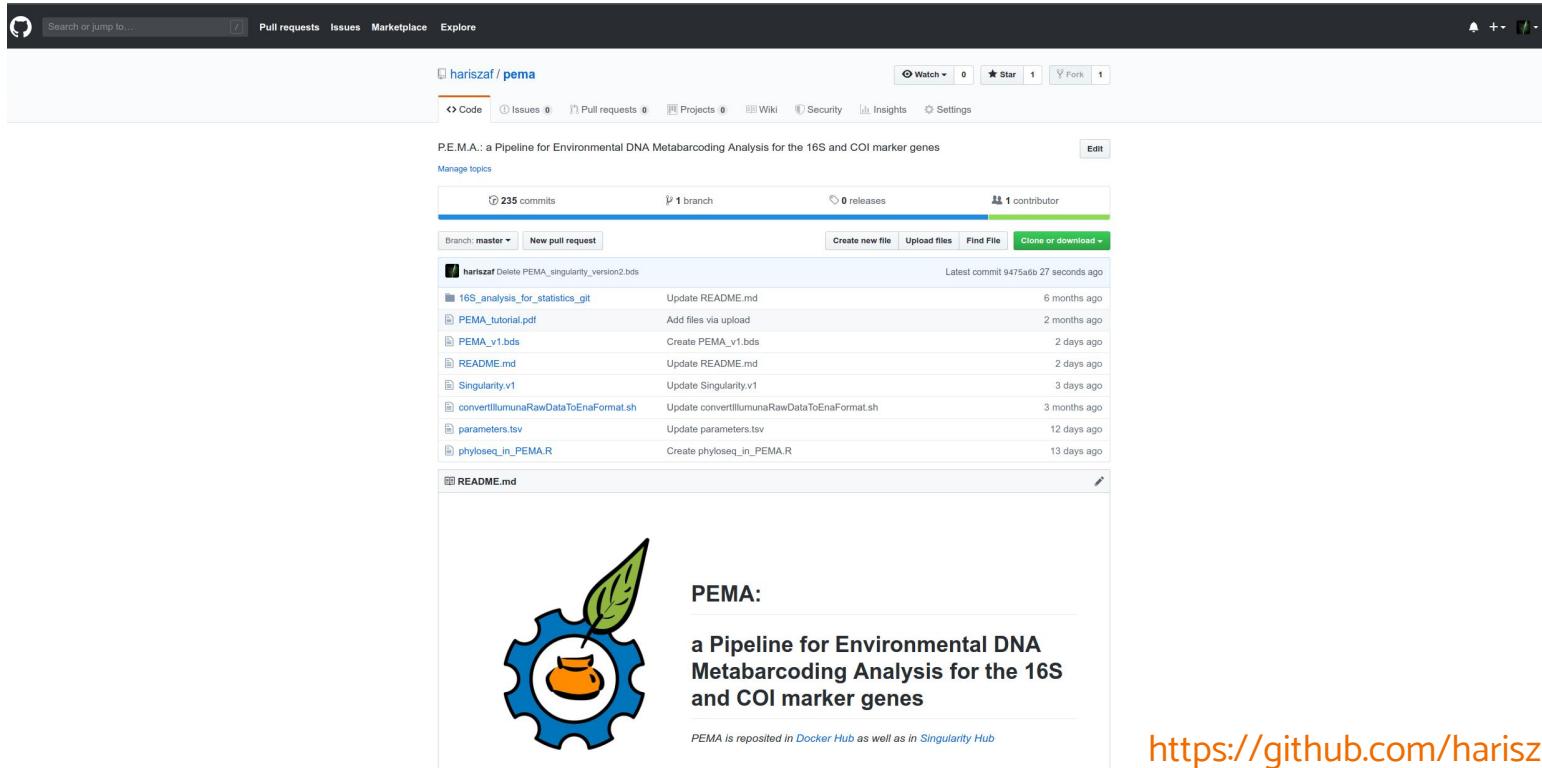
is a relatively recent approach that focuses on the simultaneous detection of a large number of species, that uses universal PCR primers to identify DNA from a mixture of organisms.



Millions of reads: what could be done with so many of them?



PEMA's home page is its GitHub repository



The screenshot shows the GitHub repository page for 'hariszaf / pema'. The repository name is 'P.E.M.A.: a Pipeline for Environmental DNA Metabarcoding Analysis for the 16S and COI marker genes'. It has 235 commits, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted. The file list includes '16S_analysis_for_statistics_git', 'PEMA_tutorial.pdf', 'PEMA_v1.bds', 'README.md', 'Singularity.v1', 'convertIlluminaRawDataToEnaFormat.sh', 'parameters.tsv', 'phyloseq_in_PEMA.R', and 'README.md'. A logo featuring a blue gear with a green leaf and an orange flower is displayed. The text 'PEMA: a Pipeline for Environmental DNA Metabarcoding Analysis for the 16S and COI marker genes' is present, along with a note that 'PEMA is reposited in Docker Hub as well as in Singularity Hub'. The URL <https://github.com/hariszaf/pema> is shown at the bottom.

PEMA:
a Pipeline for Environmental DNA
Metabarcoding Analysis for the 16S
and COI marker genes

PEMA is reposited in Docker Hub as well as in Singularity Hub

<https://github.com/hariszaf/pema>

PEMA and its containers

The screenshot shows the Singularity Hub interface for the PEMA pipeline. At the top, there are navigation links: Collections, About, User Guide, Get Help, and a search bar. The user is signed in as hariszaf. Below the header, the pipeline name is hariszaf/pema. A brief description states: "PE.M.A.: a Pipeline for Environmental DNA Metabarcoding Analysis for the 16S and COI marker genes". There are tabs for SUPPLEMENTARY, SETTINGS, USAGE, and a copy icon. Under the "Builds" section, there is one entry: "Builds COMMIT". The build details show: URL: hariszaf/pema:v1, Recipe: Singularity v1, Status: COMPLETE, Tag (Branch): v1 (master), Date: July 17, 2019, 8:22 a.m. commit.

<https://singularity-hub.org/collections/2295>

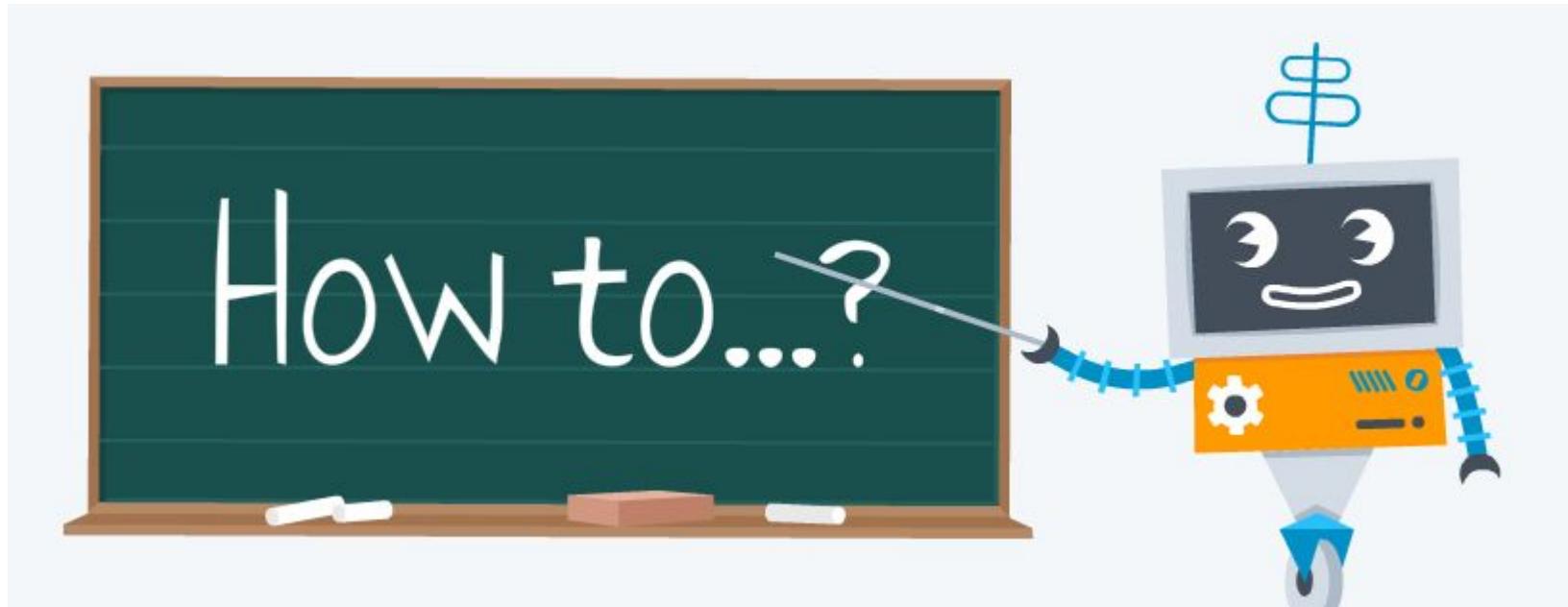
<https://hub.docker.com/r/hariszaf/pema>

In order to use PEMA, you need to get either Docker or Singularity on your computer environment.

If you are working on server/cluster, you will need to ask your sys-admin to do that for you.

The screenshot shows the Docker Hub page for the hariszaf/pema repository. It features a blue hexagonal icon. The repository name is hariszaf/pema with a star icon. It was created by hariszaf and updated 11 days ago. The description is: "PE.M.A.: a Pipeline for Environmental DNA Metabarcoding Analysis for the 16S and COI marker genes". The repository type is Container. Below the repository info, there are tabs for Overview and Tags. The Overview tab contains a detailed description of PEMA's functionality and a note about COI clustering algorithms. The Tags tab is currently empty. On the right side, there is a "Docker Pull Command" section with the command "docker pull hariszaf/pema" and a copy icon, and an "Owner" section showing the user hariszaf with a profile picture. The page number 5 is at the bottom right.

And now..



run P.E.M.A. ?

P.E.M.A. and input format!

hariszaf / pema

Watch 0 Star 0 Fork 0

Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master pema / convertillumunaRawDataToEnaFormat.sh Find file Copy path

hariszaf Update convertillumunaRawDataToEnaFormat.sh edea226 30 minutes ago

1 contributor

96 lines (66 sloc) | 2.24 KB Raw Blame History

```
1 #!/bin/bash
2
3 # This is a script that converts the raw data file of Illumina sequencer, to the ENA format.
4 # In case of P.E.M.A. the ENA format was used as a template.
5 # All sample files that are going to be used in running P.E.M.A., they need to be in the ENA format.
6
7
8 directory=${1}
9
10 # set the directoryPath the proper way no matter how the path was given by the user
11 if [ ${directory:1} == "/" ]
12 then
13     directoryPath=$directory
14 else
15     cd $directory
16     directoryPath=$(pwd)
17 fi
```

If your raw data are not in ENA format, then you should use a script we have made for this task. You can find it in “pema” repository on GitHub.

Parameters' file

```
#####
##### PEMA 's PARAMETERS #####
#####

#
# In this file there are all the parameters that need to be assigned every time PEMA is about to run!
# The parameters we have here, are not the only parameters of the tools invoked by PEMA.
# Hence, we encourage you the most to study the manual of each tool and make them as good as possible for your specific
# experiment.In the link next to each tool, you can find further information about its parameters.
#
# ATTENTION!
# From each variable you have to leave EXACTLY ONE (1) TAB and then fill the parameter as you wish.
#
#
#####
##### The parameter setting starts from here! #####
#####

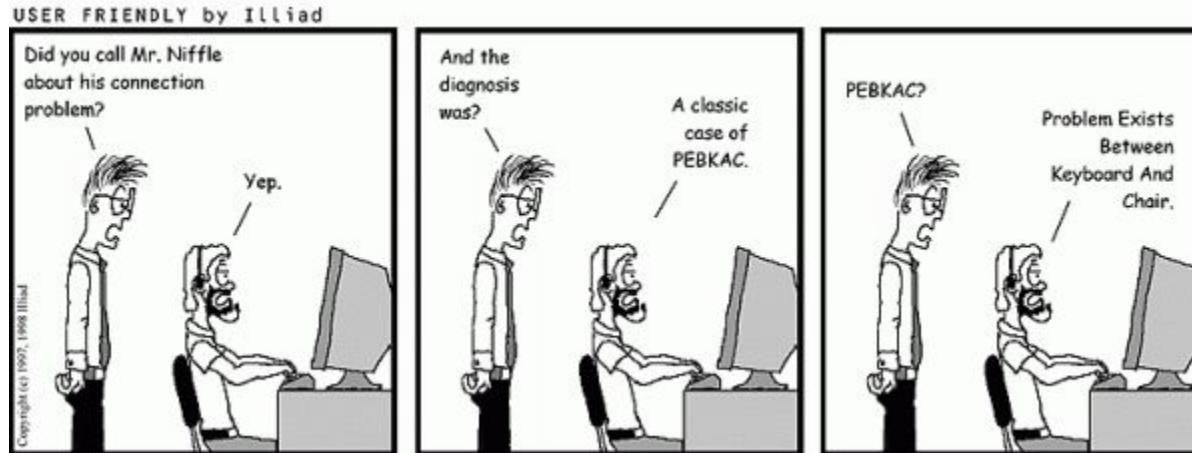
#
## give in your each unique experiment a NAME, so a single output file will be created for each of them
#
outputFolderName testingSingularity
#
#
```

You need to set the parameters' file **every time** you want to run PEMA.

We strongly suggest that you give a different "outputFolderName" every time you run it.

Find more about each parameter in the *parameters.tsv* file.

Three commands (!?) and a few minutes away!



PEMA as a..



container

Docker and PEMA on your computational environment

By running the command:

```
docker version
```

You can check that Docker is on your computer and that is ready to go.

If you get an error, then you need to install Docker on your machine.

Step 1

Get a PEMA container using this command:

```
docker pull hariszaf/pema:v1
```

You now have the Docker image of PEMA on your environment, called:

hariszaf/pema

which has the **v1** tag and you are ready to use it!

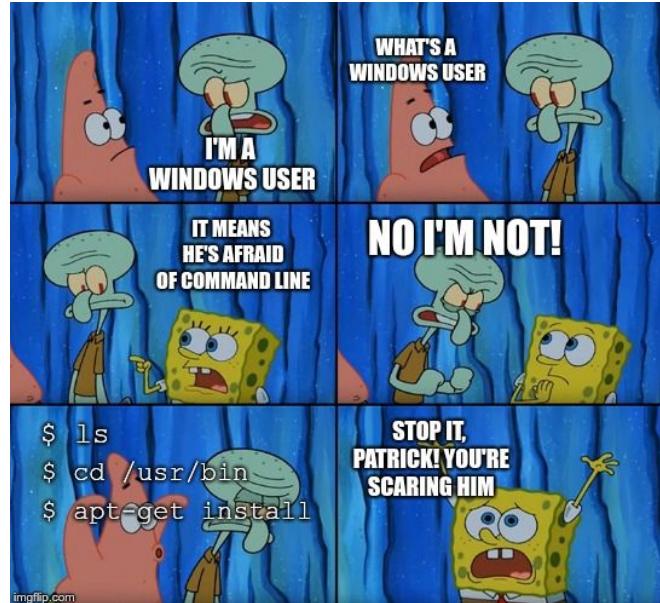
Preparing PEMA to run.. with no command line at all!

Step 2

You can make all things that PEMA needs to run, **without using the command line at all!**

- You just create the “**analysis_folder**” anywhere on your computer you want.
- You download the ***parameters.tsv*** file from PEMA’s GitHub repository and add a copy of it in the “**analysis_folder**”.
- You create a subdirectory called “**mydata**” inside the “**analysis_folder**” where you drag and drop your raw data.
- You edit the ***parameters.tsv*** file the way you want by using a text editor.

If you are about to use the ***phyloseq*** R package, then you also need to download and add on your analysis folder the ***phyloseq_in_PEMA.R*** from PEMA’s GitHub repository, too.
You can also add your metadata file which needs to be called “***metadata.csv***”.



Attention!
Everything coloured like **this**,
needs to be called **exactly** as
it is on this slide!

Or.. in the traditional way!

Step 2

- Create a folder for your analysis. Let's call it “**analysis folder**”.

```
mkdir analysis_folder
```

- Download the **parameters.tsv** file from GitHub and then make a copy of that on your analysis folder using this command:

```
cp /<path_on_your_pc>/parameters.tsv /<path_to_your_>/analysis_folder
```

Otherwise, you can download it from github directly on the server with a command equivalent to wget and then adjust:

```
wget https://raw.githubusercontent.com/hariszaf/pema/master/parameters.tsv
```

- Add your raw data in a subfolder of the “analysis folder” which it has to be called “**mydata**”.

```
cp /<path_to_rawdata/* /<path_to_your_>/analysis_folder/ mydata
```

In the end, you need to have something like this:

```
haris@zorba:~/analysis_folder$ ls  
mydata  parameters.tsv
```

mydata is a
subdirectory containing
only your raw data and
it needs **always** to be
called like that

Now let us run PEMA

```
user@XPS-13-9343:~/Desktop$ docker run -it -v docker run -it -v  
/<path_to>/analysis_folder:/mnt/analysis hariszaf/pema:v1
```

Step 3

Then, **you are inside a container** which looks like this:

```
root@27fdf0381193:/home# ls  
GUUniFrac  PEMA_v1.bds  scripts  tools
```

And the only thing left to do is to run PEMA:

```
root@27fdf0381193:/home# ./PEMA_v1.bds
```

Again, everything coloured like **this**, needs to be set exactly as it is on this slide!

PEMA as a..



container

Singularity and PEMA on your computational environment

Make sure that you have Singularity on your HPC environment:

```
$ singularity --version
```



Step 1

All version ≥ 2.6 are ok. In case you don't have Singularity, ask your IT group to do this for you.

Get a PEMA container using this command:

```
singularity pull shub://hariszaf/pema:v1
```

You now have a Singularity container of PEMA on your environment, called:

```
pema_v1.sif
```

and you are ready to use it!

Some info for your sys admin

The screenshot shows the 'Quick Start' page of the Singularity 3.0 documentation. At the top left is a large logo with a stylized 'S' inside a circle, divided into four quadrants of different colors (blue, green, grey, orange). To the right of the logo, the text 'Singularity container' and '3.0' are visible. Below the logo is a search bar labeled 'Search docs'. On the far left, there's a sidebar with a tree icon and a list of navigation items: 'Quick Start', 'Quick Installation Steps' (which is expanded, showing 'Install system dependencies', 'Install Go', 'Clone the Singularity repository', 'Install Go dependencies', and 'Compile the Singularity binary'), 'Overview of the Singularity Interface', 'Download pre-built images', 'Interact with images', 'Build images from scratch', and 'Contributing'. The main content area has a header 'Quick Start' with a 'Docs » Quick Start' link and an 'Edit on GitHub' button. A sub-header 'Quick Start' is followed by a paragraph about root privileges. It then says: 'If you need to request an installation on your shared resource, see the requesting an installation help page for information to send to your system administrator.' Below that is a link: 'For any additional help or support contact the Sylabs team: <https://www.sylabs.io/contact/>'. The next section is 'Quick Installation Steps' with a sub-header 'Install system dependencies'. It says: 'You will need a Linux system to run Singularity. See the [installation page](#) for information about installing older versions of Singularity.' Finally, there's a code block with a green background containing the command: '\$ sudo apt-get update && sudo apt-get install -y \ build-essential \
libssl-dev \
uuid-dev \
libgpgme11-dev \
squashfs-tools'



You can get Singularity 3.0 from [here](#).

Singularity allows container - based technology to be merged with HPC.

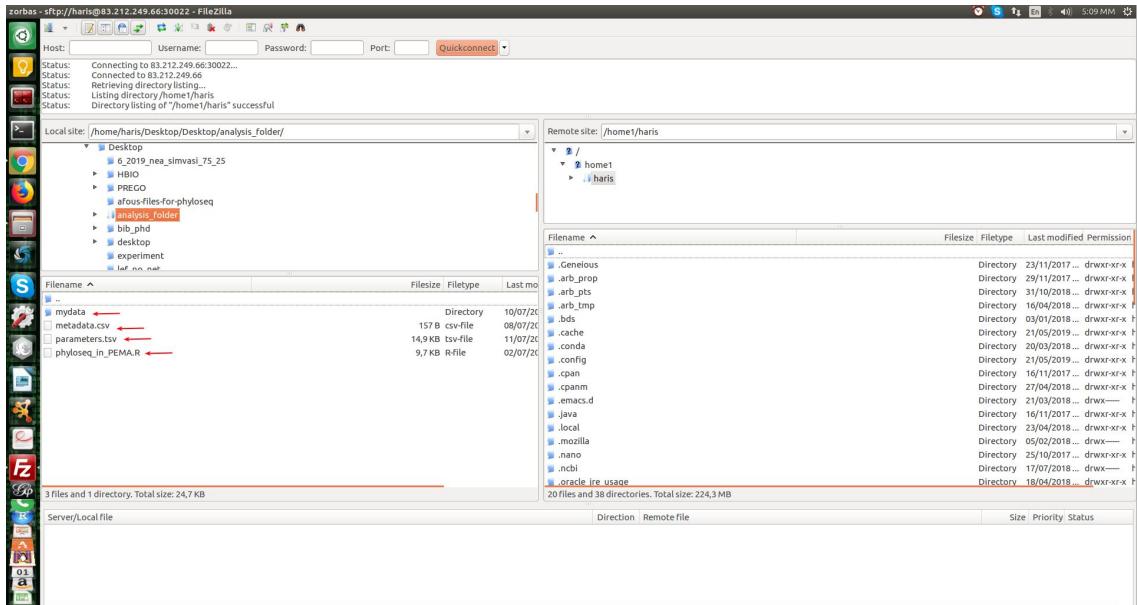
However, it can also be used for servers or any other computer.

Please sys - admin, do not panic!
Singularity is just an app!!

Preparing PEMA to run.. with no command line at all!

Step 2

Likewise in the case of Docker, you can still prepare everything PEMA needs in order to run, on your Desktop and then use something like [FileZilla](#) in order to move your “analysis_folder” on your cluster/server environment.



Or.. in the traditional way!

Step 2

- Create a folder for your analysis. Let's call it “**analysis folder**”.

```
mkdir analysis_folder
```

- Download the **parameters.tsv** file from github and then make a copy of that on your analysis folder using this command:

```
scp -P <port_number> /<path_on_your_pc>parameters.tsv <user>@<cluster_ip>://<path_to>analysis_folder
```

Otherwise, you can download it from github directly on the server with a command equivalent to wget and then adjust:

```
wget https://raw.githubusercontent.com/hariszaf/pema/master/parameters.tsv
```

- Add your raw data in a subfolder of the “analysis folder” which it has to be called “**mydata**”.

```
scp -P <port_number> /<path_to_rawdata/* <user>@<cluster_ip>://<path_to>/analysis_folder/ mydata
```

In the end, you need to have something like this:

```
haris@zorba:~/metabar_pipeline/PEMASingularity/testFile$ ls  
mydata parameters.tsv
```

mydata is a subfolder containing **only** your raw data and it needs always to be called like that

Step 3

Create your sbatch file: **touch pema_job.sh**
and edit it using any editor of your choice

```
#!/bin/bash

#SBATCH --partition=<specify_the_cluster_partition_where_your_job_will_be_submitted>
#SBATCH --nodes=<number_of_nodes_your_job_will_allocate>
#SBATCH --nodelist=<optional_specify_the_nodes_that_will_be_allocated>
#SBATCH --ntasks-per-node=<maximum_ntasks_be_invoked_on_each_core>
#SBATCH --mem=<optional_memory_per_node_specification>
#SBATCH --job-name="<give_a_name_to_your_job>"
#SBATCH --output=myanalysis.output
#SBATCH --mail-user=<put_here_your_email_adress>
#SBATCH --mail-type=ALL
#SBATCH --requeue

singularity run -B /<path_of_your_analysis_folder/ :/mnt
/<path_to_your_PEMA_container>/ pema_v1.sif
```

This is your **sbatch file**, in case that your HPC uses **SLURM**, that allows you to submit a job to a cluster. If you are using another cluster management and job scheduling system, your sbatch will have a different format.



Here is an example.

```
#!/bin/bash

#SBATCH --partition=batch
#SBATCH --nodes=1
#SBATCH --nodelist=node-12
#SBATCH --ntasks-per-node=20
#SBATCH --mem=
#SBATCH --job-name="PEMA_tutorial"
#SBATCH --output=myPEMAanalysis.output
#SBATCH --mail-user=haris.zafra@gmail.com
#SBATCH --mail-type=ALL
#SBATCH --requeue
```



```
singularity run -B /home1/haris/mikroviokosmos/analysis_folder/:/mnt
/home1/haris/mikroviokosmos/pema_v1.sif
```

And what if there is no cluster?

For Singularity there is no need for a cluster!

It is going to run exactly in the same way as before, but this time there is no need of the “sbatch file”.

A simple .sh script will do the job!

```
singularity run -B /<path_of_your_analysis_folder/:/mnt  
/<path_to_your_PEMA_container>/pema_v1.sif
```

That means you can use Singularity both on a cluster and on your personal computer.

We strongly suggest Singularity over Docker for running PEMA.



Photo by: Emmanouela Panteri, IMBBC

Step 3

Run this command:

`sbatch pema_job.sh`

and your job has now be submitted.

The **`sbatch`** command is again a **SLURM** command and if your HPC uses something else you need to do submit your job the way it does so.



Then run:

`squeue`

to see that your analysis is **up and running!**



Let us see some..



PEMA's output directory looks like this!

```
drwxr-xr-x 66 haris users 8.0K Jul 4 21:15 1.quality_control  
drwxr-xr-x 2 haris users 12K Jul 4 19:20 2.trimomatic_output  
drwxr-xr-x 34 haris users 4.0K Jul 4 20:05 3.correct_by_BayesHammer  
drwxr-xr-x 3 haris users 4.0K Jul 4 20:34 4.merged_by_PANDAseq  
drwxr-xr-x 2 haris users 4.0K Jul 4 20:35 5.dereplicate_by_obiuniq  
drwxr-xr-x 2 haris users 4.0K Jul 4 20:41 6.linearized_files  
drwxr-xr-x 3 haris users 30Jul 4 19:14 7.gene_dependent  
drwxr-xr-x 34 haris users 4.0K Jul 4 21:13 8.output_per_sample  
-rw-r--r-- 1 haris users 147M Jul 4 20:35 all_samples.fasta  
drwxr-xr-x 2 haris users 217Jul 4 21:15 checkpoints_for_lakes_235bp_only_d_1  
-rw-r--r-- 1 haris users 15KJul 4 19:14 parametersOf.lakes_235bp_only_d_1.tsv  
-rw-r--r-- 1 haris users 147M Jul 4 20:35 teliko_all_samples.fasta
```

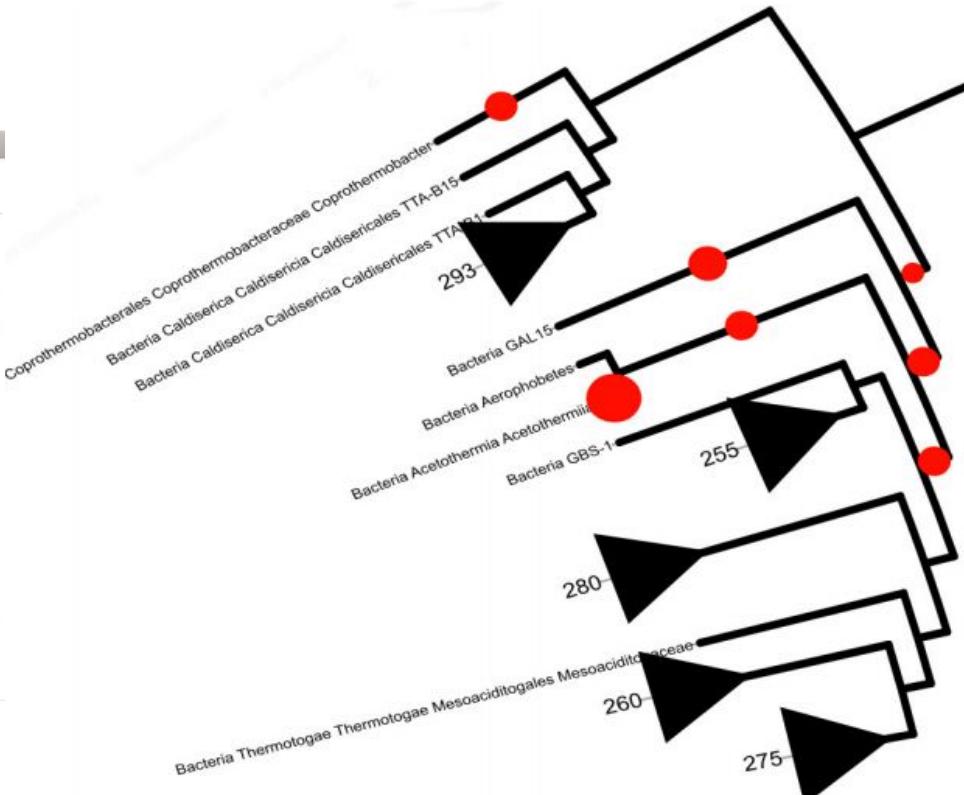
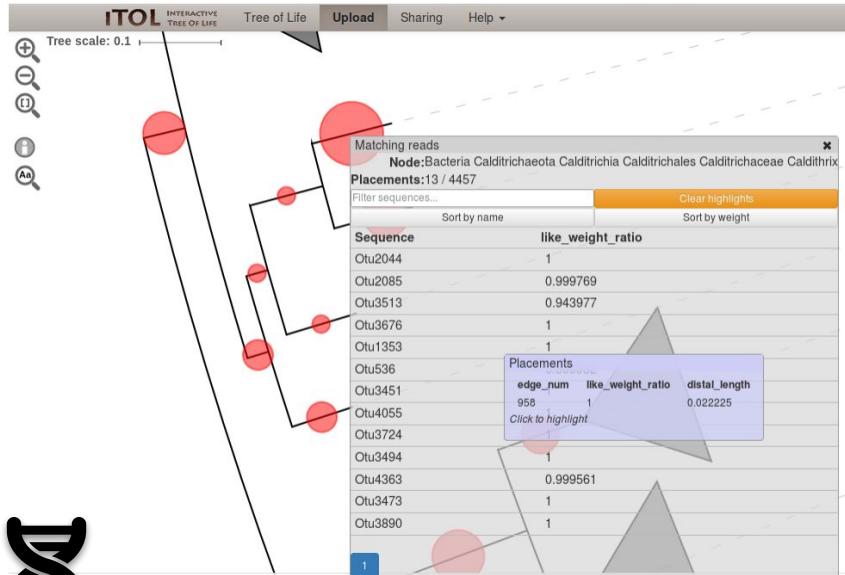
In orange you can see the directories that are made in each step and on the 7th directory you can find the (M)OTU clustering and the taxonomy assignment.

OTU-table for the case of 16S marker gene

OTU	ERR1906855	ERR1906853	ERR1906863	ERR1906856	ERR1906857	ERR1906859	ERR1906858	ERR1906870	ERR1906854	ERR1906867	E
								classification			
RR1906866		ERR1906865	ERR1906861	ERR1906862	ERR1906864	ERR1906869	ERR1906860	ERR1906868			
Otu4056	8	4	0	8	3	6	0	3	0	2	0
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Gammaproteobacteria;Vibrionales;Vibrionaceae;Vibrio											Main genome;
Otu4057	23	5	1	6	8	0	13	0	2	0	0
Bacteria;Bacteria (superkingdom);FCB group;Caldithrix phylum incertae sedis;Caldithrix class incertae sedis;Caldithrix order incertae sedis;Caldithrix family incertae sedis;Unknown Caldithrix family incertae sedis genus 4											Main genome;
Otu4054	0	0	0	1	2	0	12	0	0	0	0
Bacteria;Bacteria (superkingdom);Fusobacteria (superphylum);Fusobacteria;Fusobacteriia;Fusobacterales											Main genome;
Otu4055	0	1	0	4	9	0	9	0	1	0	0
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Deltaproteobacteria;Myxococcales;Unknown Myxococcales family 16											Main genome;
Otu4052	1	3	2	2	1	0	5	0	1	0	0
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Deltaproteobacteria;Myxococcales;VHS-B4-70											Main genome;
Otu4053	0	0	0	1	4	0	11	0	0	13	0
Bacteria;Bacteria (superkingdom);CPR;Ca. Parcubacteria;Candidatus Falkowbacteria											Main genome;
Otu4050	5	3	1	5	2	0	9	0	3	0	0
Bacteria;Bacteria (superkingdom);FCB group;Bacteroidetes;Sphingobacteriia;Sphingobacterales;Unknown Sphingobacterales family 25											Main genome;
Otu4051	0	0	0	5	6	0	14	0	0	1	0
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Deltaproteobacteria;Desulfobacterales;Desulfobulbaceae;Desulfobulbus											Main genome;
Otu4058	2	6	0	15	12	12	9	3	4	11	8
											2
											Main genome;



MOTUs placements after phylogeny based taxonomic assignment



High Performance Computing in a modern marine research landscape

IMBBC - HCMR HPC Infrastructure

- Bioinformatics
- Biodiversity informatics
- Genomics
- Ecology
- Optimized for I/O-intensive applications
- Optimized for *de novo* sequence assembly
- Data- and function-based parallelized pipelines

A great thanks to our HPC IT group

Dimitris Sidirokastritis, Minas Maris, Antonis Potirakis, Stelios Ninidakis
and also Ha Quoc Viet

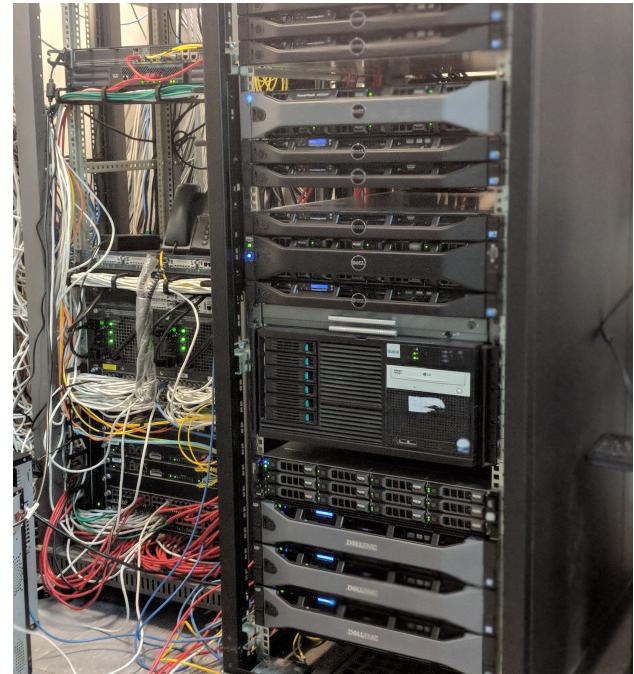


Photo by: Emmanouela Panteri, IMBBC



Lab42 Open

IMBBC
INSTITUTE OF MARINE BIOLOGY,
BIOTECHNOLOGY AND AQUACULTURE



Special thanks to:

Evangelos Pafilis (for being the brain)

IMBBC HPC group (for being the muscle)

Christina Pavloudi (for being the brain and
the muscle)

Ha Quoc Viet (for the patience)

Office No 40 (for being the best and for the
coffee!) and George the.. Photoshop-er

Also, Dimitris Sidiropoulos and Antonis
Potirakis

**Thank you for
your attention!**