PEMA

a pipeline for eDNA metabarcoding analysis

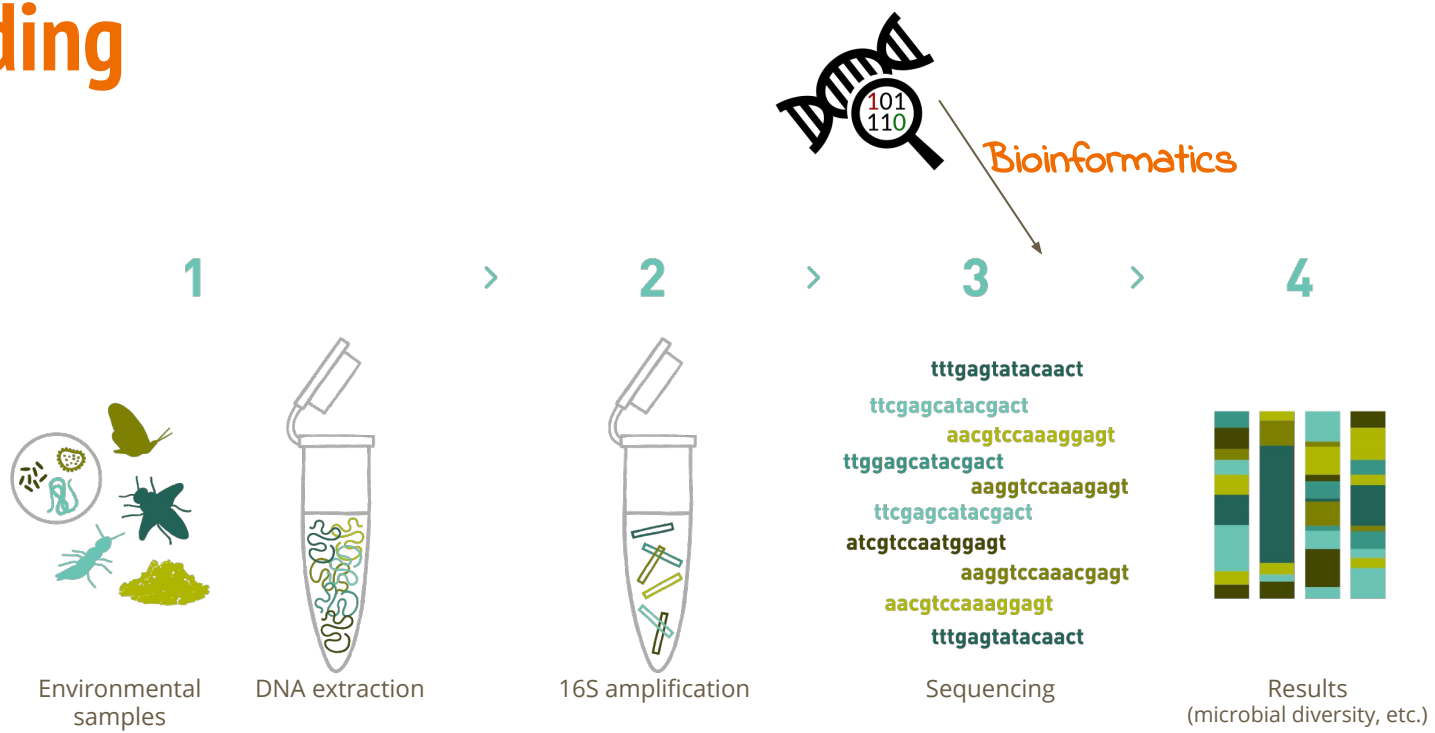
By:
Haris Zafeiropoulos (haris-zaf@hcmr.gr)

ink well by Daria Moskvina from the Noun Project

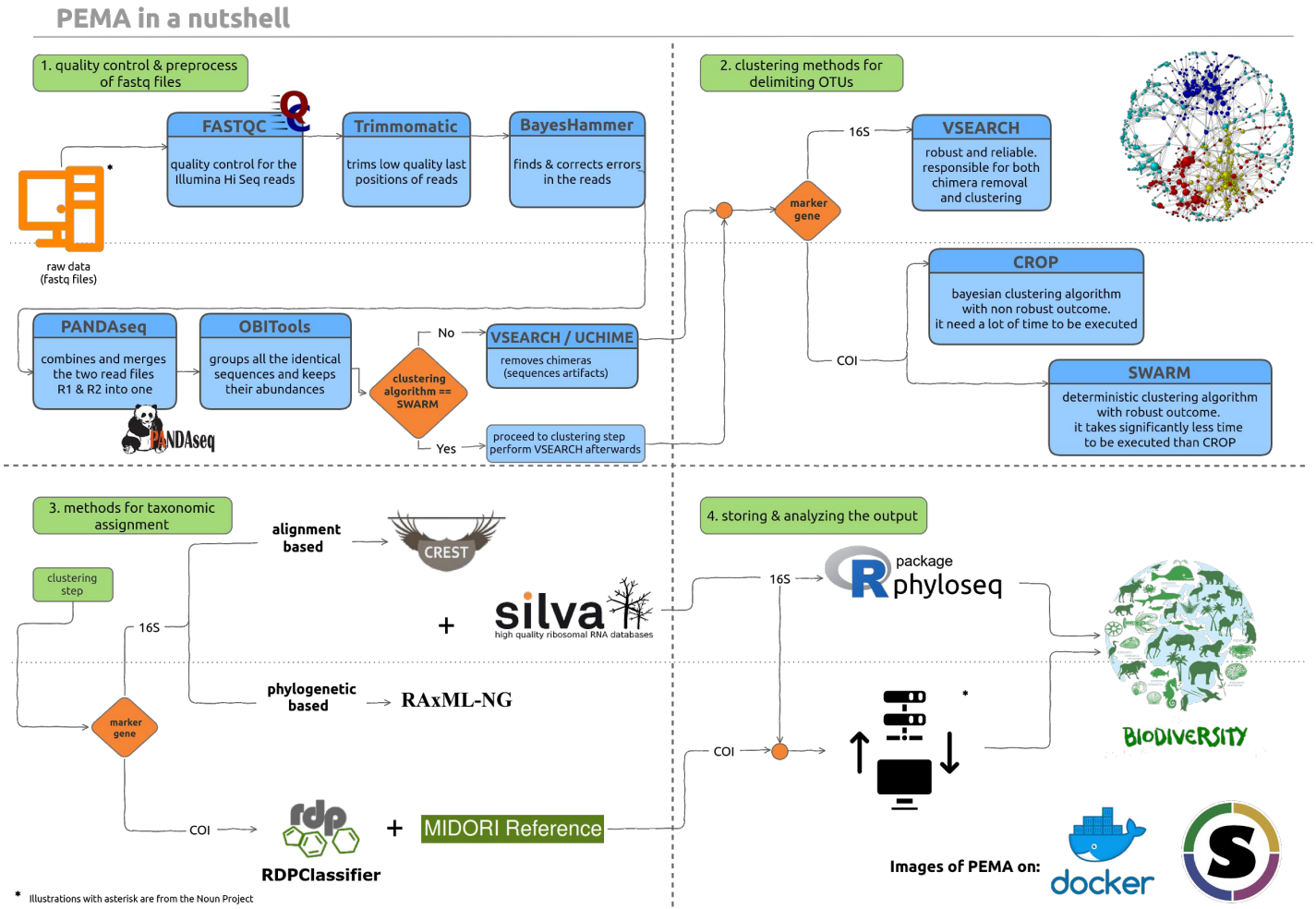
You can find this tutorial as Google Slides here:
https://docs.google.com/presentation/d/1lVH23DPa2NDNBhVvOTRoip8mraw8zfw8VQwbK4vkB1U/edit?usp=sharing

Metabarcoding

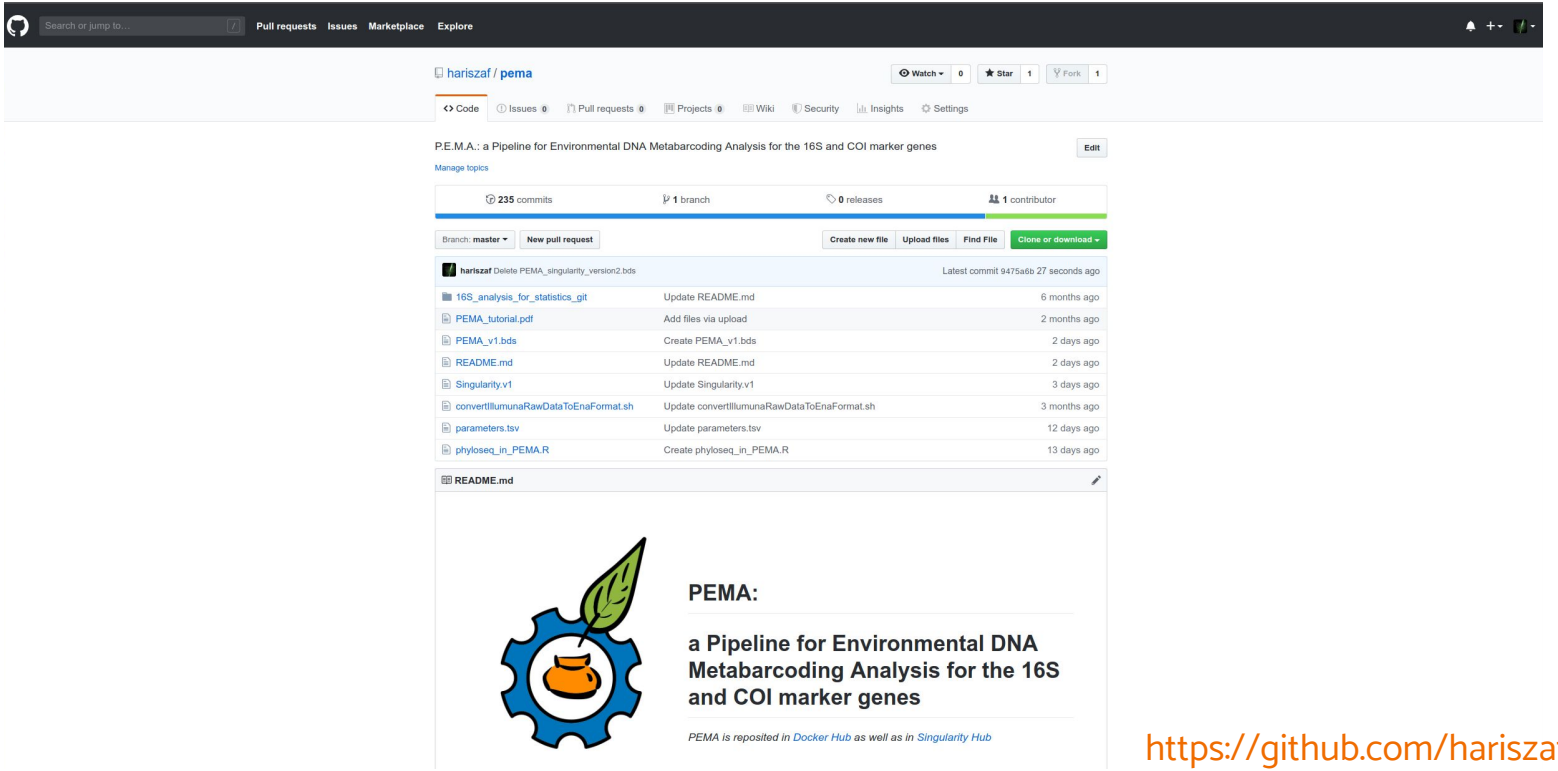
is a relatively recent approach that focuses on the simultaneous detection of a large number of species, that uses universal PCR primers to identify DNA from a mixture of organisms.



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

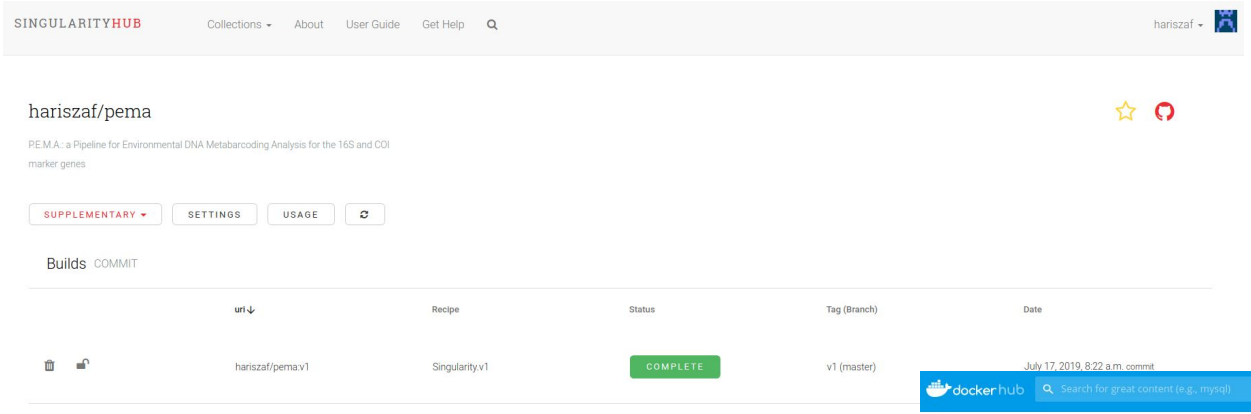


PEMA's home page is its GitHub repository



https://github.com/hariszaf/pema

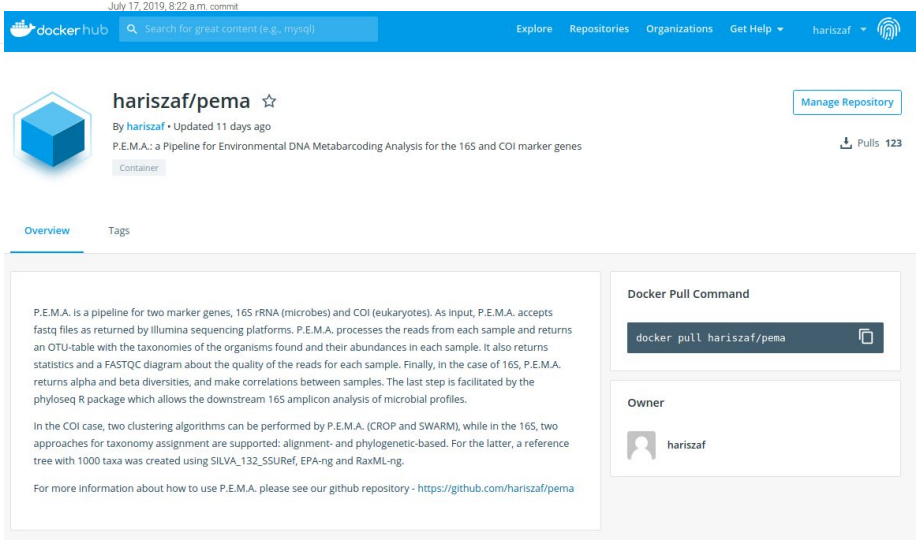
PEMA and its containers



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

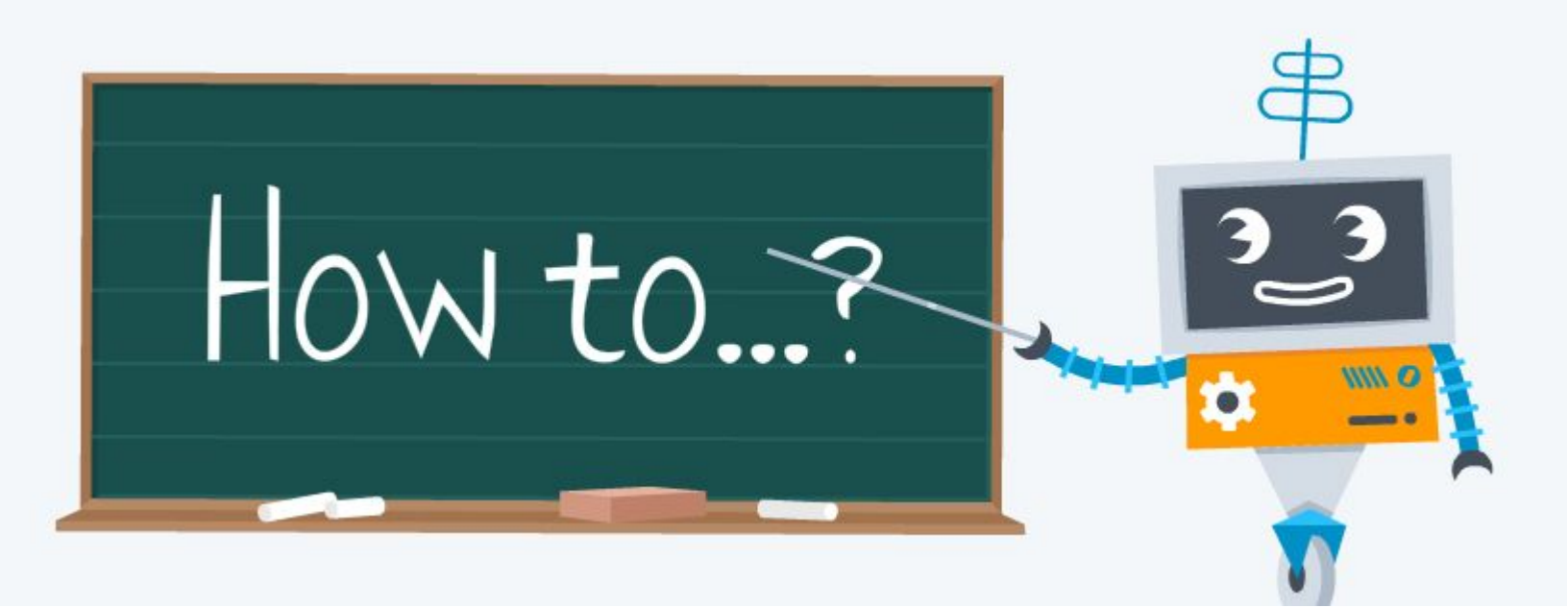
In order to use PEMA, you need to get either **Docker** or **Singularity** on your computer environment.

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



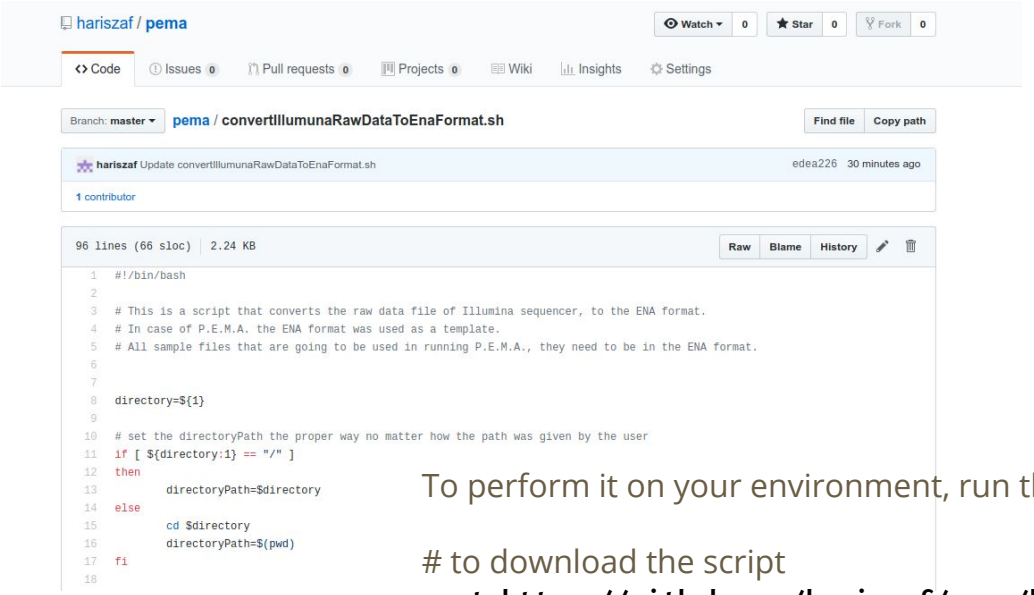
https://hub.docker.com/r/hariszaf/pema

And now..



run P.E.M.A. ?

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



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 on PEMA's repository on GitHub.

ATTENTION! Your raw data files need to look like this:
forward read: "<anything>1.fastq.gz"
reverese read: "<anything>2.fastq.gz"

To perform it on your environment, run the following commands:

```
# to download the script
wget https://github.com/hariszaf/pema/blob/master/convertIllumunaRawDataToEnaFormat.sh

# to make it executable
chmod +x convertIllumunaRawDataToEnaFormat.sh

# to run the script
./convertIllumunaRawDataToEnaFormat.sh /path/to/your/my_data/directory
```

Parameters' file

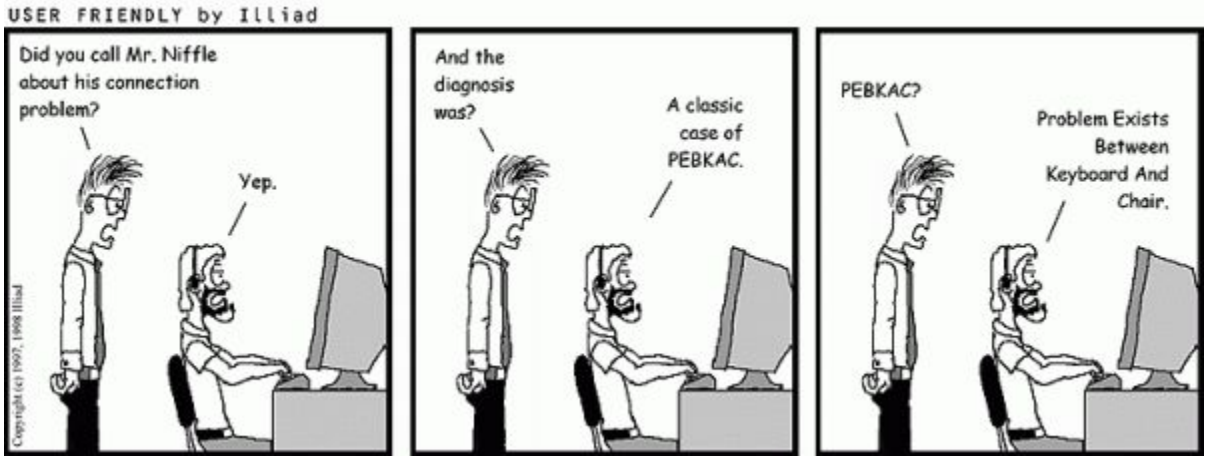
```
#############################################################################################
################################   PEMA 's PARAMETERS   #####################################
#############################################################################################
#
#
# In this file 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 frome here!  ########################
#############################################################################################
#
## give in your each uniq 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

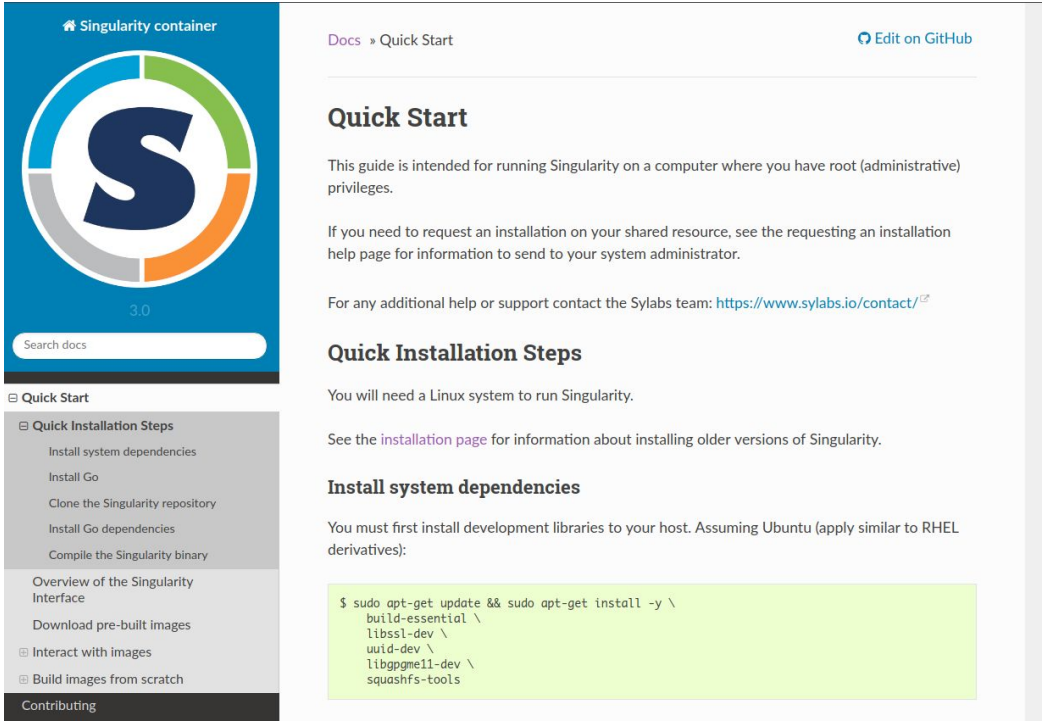
a pipeline for eDNA metabarcoding analysis

... on **HPC**

..as a

container

Some info for your sys admin



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!!

Singularity and PEMA on your computational environmnet

Step 1

Make sure that you have Singularity on your HPC environment:

```
$singularity --version
```

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

As different versions of Singularity handle the "pull" command in a different way, please always check how the container you just pulled has been named.

Get a PEMA container using this command:

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

All after the ":" symbol, is called "tag" and change from version to version.
Check PEMA's Singularity Hub repository for more.

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

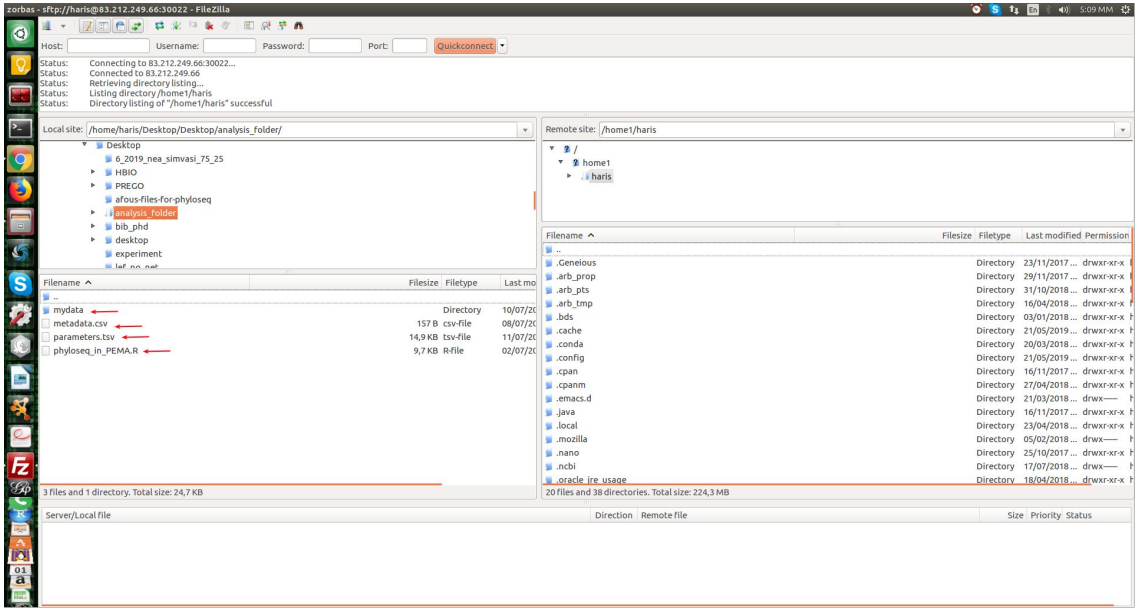
```
pema_v1.sif
```

and you are ready to use it!

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_emai_adress>
#SBATCH --mail-type=ALL
#SBATCH --requeue
```

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.

singularity run -B /<path_to>/analysis_folder/ **:/mnt/analysis**
/<path_to_your_PEMA_container>/ **pema_v1.sif**

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.zafr@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/analysis
/<path_to_your_PEMA_container>/pema_v1.sif
```

That means you can use Singularity **both** on a cluster **and** on a server or even 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!**

Taking advantage of the checkpoints!

Let us assume that you need to re-run your analysis, changing for example the clustering method.

Then, you need only to change the corresponding parameter in the *parameters.tsv* file and run the following command:

```
singularity run -B /<path_of_your_analysis_folder/ :/mnt/analysis -r /<path_to_checkpointFile>/ trimming.chp
```

PEMA

a pipeline for eDNA metabarcoding analysis

... on personal PC

.. yet a container!

A. For Linux

B. For MacOS

C. For Windows

How to get the container-based technologies on my computational environment?

A. For Unix/Linux

In case you do not have Docker on your computer, you can download **Docker Engine for a Linux system** here:
https://docs.docker.com/install/linux/docker-ce/ubuntu/

Likewise, you can get **Singularity** (any version >=2.6) in a straightforward way, following the commands you can find here:
https://sylabs.io/guides/3.0/user-guide/installation.html#install-on-linux

B. For MacOS

In case you do not have Docker on your Mac, you can download and install **Docker Desktop for Mac** from here :
https://docs.docker.com/install/linux/docker-ce/ubuntu/

You can also get **Singularity** (any version >=2.6) but this task is a bit trickier than for Linux, as you actually set up a virtual machine. Follow the commands reffering to Mac that you can find on this link:
https://sylabs.io/guides/3.0/user-guide/installation.html#install-on-windows-or-mac

For MacOS you can also have the **Singularity Desktop** version which is really a peace of cake to get!
https://sylabs.io/singularity-desktop-macos/

How to get the container-based technologies on my computational environment?

C. Windows

In case you do not have Docker on your Windows system, you can download and install **Docker Desktop for Windows** here: https://docs.docker.com/docker-for-windows/install/

Likewise the MacOS case, you can get **Singularity** (any version >=2.6) by building a Virtual Machine environment, following the steps reffering to **Windows**, you can find here: https://sylabs.io/guides/3.0/user-guide/installation.html#install-on-windows-or-mac

We strongly suggest the **Singularity** and **Singularity Desktop** (when you use MacOS) to run PEMA.

Docker and PEMA on your computational environmnet

Step 1

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.

Get a PEMA container using this command:

```
docker pull hariszaf/pema
```

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

```
hariszaf/pema
```

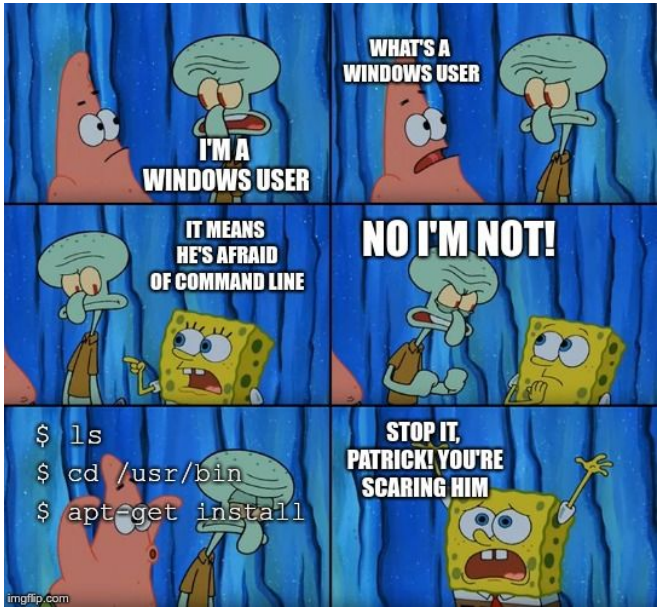
which has the `latest` tag and you are ready to use it!

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

Step 3

First, you need to start a Docker container and mount your analysis directory on it:

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

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

```
root@27fdf0381193:/home# ls
GUniFrac  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!

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.trimmomatic_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   30 Jul 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 217 Jul 4 21:15 checkpoints_for_lakes_235bp_only_d_1

-rw-r--r-- 1 haris users 15K Jul 4 19:14 parameters0f.lakes_235bp_only_d_1.tsv

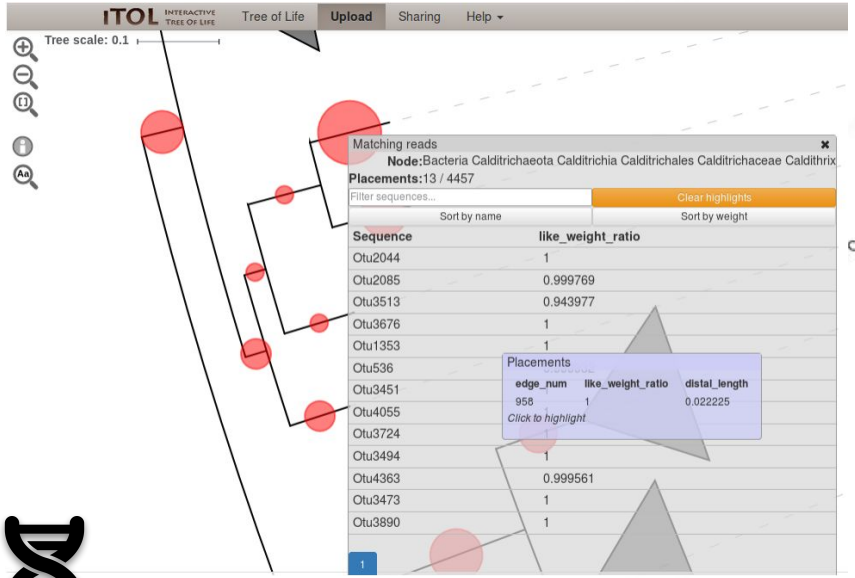
-rw-r--r-- 1 haris users 147M Jul 4 20:35 teliko_all_samples.fasta

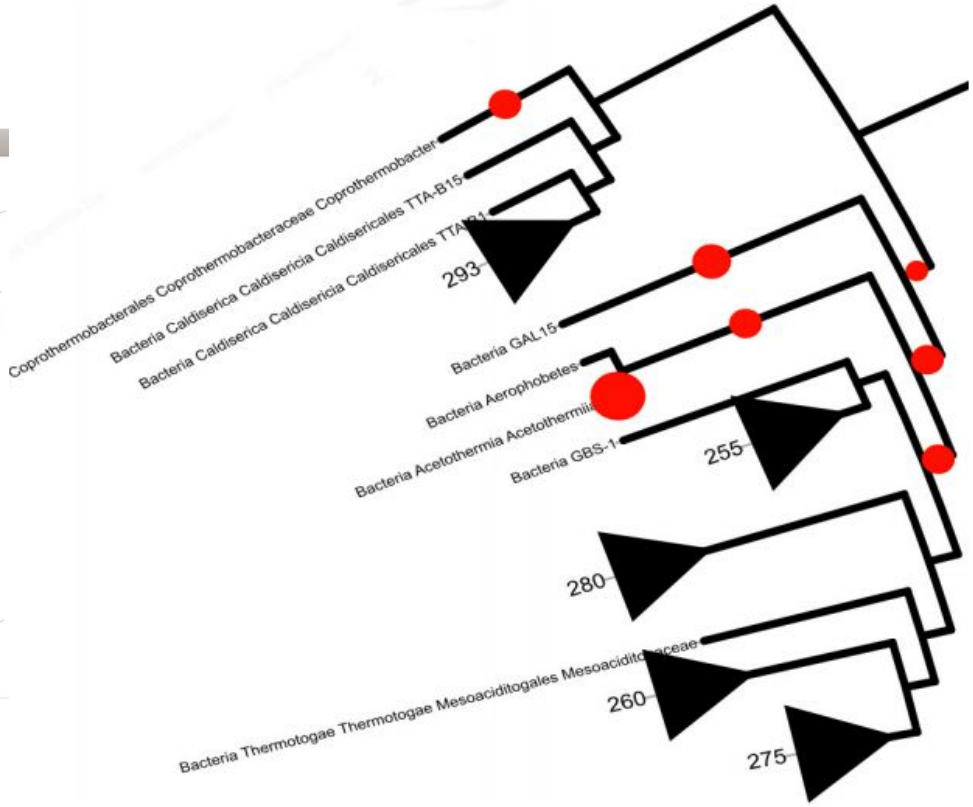
In orange you can see the directories that are made in each step and on the 7th directory you can find the (M)OTU clustering and the taxonomy assignment.

OTU-table for the case of 16S marker gene

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

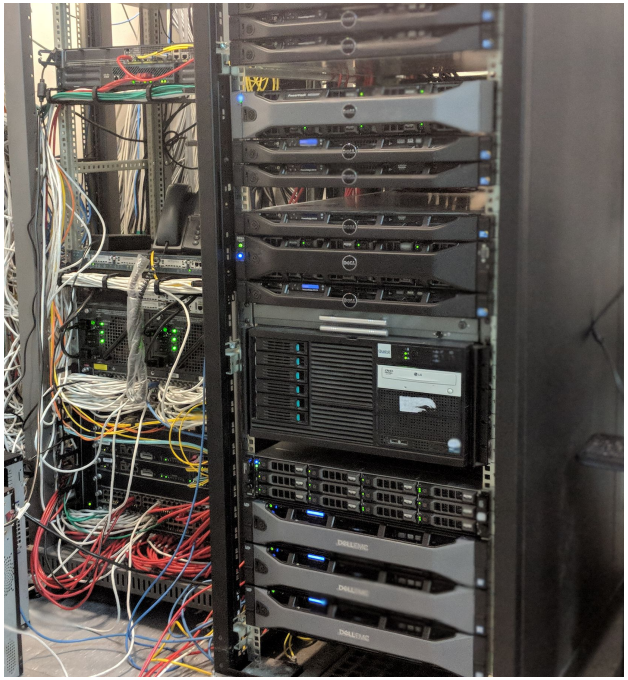


Photo by: Emmanouela Panteri, IMBBC

A great thanks to our HPC IT group

Dimitris Sidirokastritis, Minas Maris, Antonis Potirakis, Stelios Ninidakis

and also Ha Quoc Viet

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 Sidirokastritis and Antonis Potirakis

Thank you for your attention!