esgemme

Release 1.3.0

Mustafa Tekpinar

May 30, 2023

CONTENTS:

CHAPTER

ONE

USING ESGEMME VIA DOCKER

1.1 Requirements

You need to have docker installed on your machine. You can consult the following page for this: https://docs.docker.com/get-docker/

I am assuming some basic familiarity with Linux/Unix/MacOS terminal commands.

Let's start our favorite terminal app.

You must create a folder called docker-tutorial and go to that empty folder:

```
mkdir docker-tutorial
cd docker-tutorial
```

1.2 Getting the example input data

Let's download the sample data provided in the ESGEMME repository for this exercise. First, we will download the multiple sequence alignment file in fasta format:

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/ESGEMME/blob/master/data/aliBLAT.fasta
```

If you don't have wget, you can try the same command with curl:

```
curl http://gitlab.lcqb.upmc.fr/tekpinar/ESGEMME/blob/master/data/aliBLAT.fasta
```

Please verify that the aliBLAT.fasta file is in the folder.

Now, we will download the PDB (Protein Databank) file for BLAT:

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/ESGEMME/blob/master/data/blat-af2.pdb
```

1.3 Single point mutation calculations

In order to make sure that the docker is installed:

```
sudo docker -h
```

If it shows you a list of options, you are on a good track. On MacOS, you may not need 'sudo' word before the docker command at all.

```
sudo docker run -ti --rm --mount type=bind,source=$PWD,target=/home/tekpinar/research/
↪myexample \
tekpinar/esgemme-docker:v1.3.0
```

You are in the container (your virtual operating system) now. You created a folder called myexample in your container with the previous command. Let's change to that folder.

```
cd ../myexample/
```

When you check the data in that folder with 'ls' command, you are supposed to see aliBLAT.fasta and blat-af2.pdb files. Basically, your docker-tutorial folder on the host system and myexample folder on the docker container are pointing to the same place.

1.3.1 Obtaining the entire single point mutation landscape

In this step, we will use only evolutionary information from an MSA file:

```
python $ESGEMME_PATH/esgemme.py aliBLAT.fasta -r input -f aliBLAT.fasta
```

After a few minutes of calculation, you must see at least two files named BLAT_normPred_evolCombi.txt and BLAT_normPred_evolCombi.png. You have the entire single point mutational landscape of BLAT protein in these files.

If you want to utilize structural information (highly recommended) as well as evolutionary information:

```
python $ESGEMME_PATH/esgemme.py aliBLAT.fasta -r input -f aliBLAT.fasta \
--pdbfile blat-af2.pdb \
--normweightmode sstjetormax
```

1.3.2 Predicting the effect of a subset of single point mutations

If you are interested in only a bunch of single point mutations, you have to prepare a mut file. The format is a simple text file and each line contains a single point mutation such as D26A. . . . Fortunately, we have an example mut in data folder of ESGEMME repository.

```
wget http://gitlab.lcqb.upmc.fr/tekpinar/ESGEMME/raw/master/data/Stiffler_2015_BLAT_
↪ECOLX.mut
```

Similar to the previous step, there are two possible ways to do the calculations: with or without structural information. First, let's do it without structural information:

```
python $ESGEMME_PATH/esgemme.py aliBLAT.fasta -r input -f aliBLAT.fasta \
-m Stiffler_2015_BLAT_ECOLX.mut
```

You can include structural information in the following way: .. code:: bash

```
python $ESGEMME_PATH/esgemme.py aliBLAT.fasta -r input -f aliBLAT.fasta –pdbfile blat-af2.pdb
–normweightmode sstjetormax -m Stiffler_2015_BLAT_ECOLX.mut
```

You will have BLAT_normPred_evolCombi.txt file in your folder. However, the output format is completely different from the entire mutational landscape scanning file. Each line of this file is a mutation and its predicted effect separated by a space. In addition, you won't have a png file like in the previous case.

1.4 Multiple point mutation calculations

Sometimes, we need to see effects of double or triple mutations. ESGEMME can perform calculations if you provide a mut file. In this case, the mut file must have the following format:

```
E26D:Y44R
E56N:A77F:H94V
```

The first line of the text file is impact of a double mutation and the second line is the impact of the triple mutations. As you can see, the mutations are separated by a colon(:) character. The output file will be in a similar format. Each line will contain the multiple mutation and its predicted effect, separated by a space.

1.5 Running several jobs using docker

CHAPTER

TWO

ANALYZING THE ESGEMME OUTPUT

By default, ESGEMME will output the following files: * myProt_normPred_evolEpi.txt * myProt_normPred_evolInd.txt * myProt_normPred_evolCombi.txt

The most important output file is **myProt_normPred_evolCombi.txt**. The values of interest are the normalized predictions (normPred). Each file contains a 20 x n matrix, where n is the number of positions (columns) in the query sequence. If the user provides her/his own list of mutations, then only the global epistatic model will be run and the output file will contain 2 columns, the first one with the mutations, the second one with the normalized predicted effects.

CHAPTER

THREE

PREPARING YOUR OWN INPUT

3.1 Preparing Your Input MSA and PDB with Colabfold

You have a fasta file for your protein of interest and you want to understand impact of (certain) mutations. Before starting, please make sure that your fasta file does not contain a gap. The quickest method to obtain both multiple sequence alignment and a protein structure is to use Colabfold. Let's do this step by step:

1. Let's go the Colabfold web site:

   https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

   Sign in using your gmail account.

2. Click on the 'Connect' button on the top right hand side.

3. Clean 'query_sequence' box and paste your sequence to the 'query_sequence' box. For me, I selected adenylate kinase (AKE) as my example fasta sequence (https://www.rcsb.org/fasta/entry/4AKE/display).

4. Change the 'jobname' to something that makes more sense to you.

5. Go to the menu bar of your 'AlphaFold2.ipynb' notebook, where 'File, Edit, View, Insert, Runtime, Tools, Help' are listed. Click on the Runtime and select 'Run all'.

6. This process make take from a few minutes to a few hours depending on your protein size. It will give you an a3m file and up to 5 PDB models. Put these files in a clean folder and change the directory to that folder in your terminal.

7. Unfortunately, a3m file is not in fasta format and it contains gap columns. We have to clean those gaps. We can do that with a GUI program like Ugene or Jalview. However, it is a labor intensive procedure. Here, I will use a small tool that I developed and added to the ESGEMME docker image that I created.

8. Start the docker image with the following command:

   ```
   sudo docker run -ti --rm --mount type=bind,source=$PWD,target=/home/tekpinar/
   ↪research/myexample \
   tekpinar/esgemme-docker:v1.3.0
   ```

9. Now, change the directory to myexample folder.

   ```
   cd ../myexample/
   ls -l
   ```

   We are supposed to see our a3m and pdb files in this folder.

10. Let's use a small script from hhsuite to convert a3m file to fasta format.

```
reformat.pl a3m fas AKE.a3m AKE.fasta
```

11. Final step and we are there:

```
demust removegaps -i AKE.fasta -o AKE_nogaps.fasta
```

Congratulations! Now, you have all the input files required for ESGEMME: I. An input MSA: AKE_nogaps.fasta II. An input PDB: myprotein.pdb

CHAPTER

FOUR

INSTALLATION

ESGEMME is implemented in Python 3 and R. It has been tested only on Linux. Since ESGEMME has many dependencies, we recommend using our web site or our docker image. If you are a determined user, here comes the steps required to install it from the source.

4.1 Installing the dependencies:

ESGEMME has the following external dependencies:

- Joint Evolutionary Trees: http://www.lcqb.upmc.fr/JET2/ and its dependencies:
  - java
  - naccess: http://www.bioinf.manchester.ac.uk/naccess/
  - muscle: https://www.drive5.com/muscle/
- seqinr R package: https://cran.r-project.org/web/packages/seqinr/index.html
- dssp for secondary structure prediction.

These tools should be installed to be able to use ESGEMME.

4.2 Getting the source code and preparing the environment:

Download the ESGEMME source code from http://gitlab.lcqb.upmc.fr/tekpinar/ESGEMME. Define and export the environment variable ESGEMME_PATH=/path-to-ESGEMME-directory/

```
export ESGEMME_PATH=/path-to-ESGEMME-directory/
```

4.3 Configuring default.conf file

Inside ESGEMME folder, there is an important file called default.conf. This file contains essential parameters of ESGEMME, such as paths of external parts, default internal parameters. etc. You have to correct the Software section of this file according to your system.

CHAPTER FIVE

INDICES AND TABLES

- genindex
- modindex
- search