2021_SAGE_II_group9_PhageDefence

Vincent Somerville

13/04/2021

Recap Wally server

It has been some time since you entered the wally server. Here, are some refresher commands:

```
##------LOGIN to wally
ssh vsomervi@wally-front1.unil.ch
#now enter your passwrod

##------copy something to wally
scp file.txt vsomervi@wally-front1.unil.ch:/users/vsomervi/scratch/vsomervi/ # a single file
scp -r directory/ vsomervi@wally-front1.unil.ch:/users/vsomervi/scratch/vsomervi/ # a complete director

##------navigate to SAGE II directory
cd /users/vsomervi/scratch/common_files/SAGE_II

##------look what is in there
ls
ls -l

##------open README.txt (and close by pressing q)
less README.txt
```

Introduction rMarkdown

This is an rMarkdown! rMarkdown is a coding format including both written text blocks combined with code chunks and also programming output. All of this is formatted by a soft latex style. Interestingly, you can combined not only r code but also bash, python or whatever programming language you prefer. To find out more see here or the cheat sheet.

Introduction git/github

Git is a version control tool. You can think of it as if you are continuous back-upping at regular intervals and you can return to certain stages of a coding project if needed. Github is the online platform that you can use to share and publish the code. You can sign up for free and download and create your own content. I will save, update and distribute my code and presentations in the my SAGE2 repository. Let's try to organize the project with it.

Reminder Wally submission

Here is the complete script that you need to run. Don't forget to change names and direcotries

```
#!/bin/bash

####------------------------------------
##SLURM options
####------------------------------------
#SBATCH --job-name test_wally
#SBATCH --partition wally
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 2
#SBATCH --mem 2G
#SBATCH --time 03:00:00
#SBATCH --output /users/vsomervi/scratch/vsomervi/logs/%x_%j.out
#SBATCH --error /users/vsomervi/scratch/vsomervi/logs/%x_%j.err

####------------------------------------
##preparation
##set you bash variables in order to quickly call them in the script
####------------------------------------
username=vsomervi

genesss=$(echo "type I restriction") # gene name
genome=Lapis_ESL0263 # genome name

personal_home_directory=/users/${username}/scratch/${username}
Overall_output_directory=${personal_home_directory}/SAGE_II

####--------------------
##modules
####------------------------------------

module load HPC/Software
#module load UHTS/Analysis/prokka/1.13
#module load UHTS/Analysis/barrnap/0.8
module avail
module load SequenceAnalysis/HMM-Profile/hmmer/3.1b2

####------------------------------------
##start of script
####------------------------------------

start=$SECONDS
echo "Step: Test"

echo "============================================================"
date +"START : %a %b %e %Y %H:%M:%S "
echo -e "Hello " ${username}

duration=$(( SECONDS - start ))
echo -e "The script ran for "${duration} "seconds"
```

Let's do biology

Technical aspects aside let's look at the biology. If we want to find a specific mechanisms or pathway in our genomes what can we do?

look at annotations

The first step to find out if a gene is present or absent is look at the annotations. A useful function to scan annotations if you remember is grep. With grep you can scan your annotation .gff files of the Lapis_ESL0263 genome file as follows:

```
grep "type I restriction" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/Lapis_ESL0263/*gff
##------------OR-----------------------------------------------
genesss=$(echo "type I restriction") # gene name
grep "${genesss}" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/Lapis_ESL0263/*gff
##------------OR-----------------------------------------------
genesss=$(echo "type I restriction") # gene name
genome=Lapis_ESL0263 # genome name
grep "${genesss}" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/${genome}/*gff
```

How can we also output our results appropriately? In the end we want to know if the gene if present or not? (and if present is it multicopy?) We can do this for example with if else statements. Have a look here how it goes.

```
genesss=$(echo "type I restriction")
genome=Lapis_ESL0263
grep "${genesss}" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/${genome}/*gff
geneCount=$(grep -c "${genesss}" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/${genome}/*gff
mkdir -p /users/vsomervi/scratch/vsomervi/SAGE_II/01_greps_gff/ # create directory
echo -e "genome\tgene\tcount" > /users/vsomervi/scratch/vsomervi/SAGE_II/01_greps_gff/output_grep.txt #

if [ "$geneCount" = "0"  ] ##first if statement
        then
        echo "NO gene identified"
        echo -e ${genome}"\t"${genesss}"\t"${geneCount} >> /users/vsomervi/scratch/vsomervi/SAGE_II/01_
elif [ "$geneCount" = "1"  ]  ##second if statement
        then
        echo "one gene identified"
        echo -e ${genome}"\t"${genesss}"\t"${geneCount} >> /users/vsomervi/scratch/vsomervi/SAGE_II/01_
else  ##remaining cases
        echo "multiple genes identified"
        echo -e ${genome}"\t"${genesss}"\t"${geneCount} >> /users/vsomervi/scratch/vsomervi/SAGE_II/01_
fi
```

This is a simple grep but how can be grep multiple names in a loop? Here is an example how you can grep multiple gene names in a loop.

```
for genes in $(echo "lacZ lacR lacS lacA")
do
echo ${genes}
grep "${genes}" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/Lapis_ESL0263/*gff
echo "======================"
done
```

or if you have create a list of your genes of interest you can also cat the list:

```
for genes in $(cat /users/vsomervi/scratch/common_files/gene.list)
do
echo ${genes}
grep "${genes}" -i /users/vsomervi/scratch/common_files/SAGE_II/Genomes/Lapis_ESL0263/*gff
echo "======================"
done
```

finally, you do not only want to query a single genome but multiple genomes. therefore I suggest you loop through the genomes. Try it out.

What is next? If we find a gene in our annotations this is usually a good sign that it is present (true positives). However if it is not there it could mean two things.

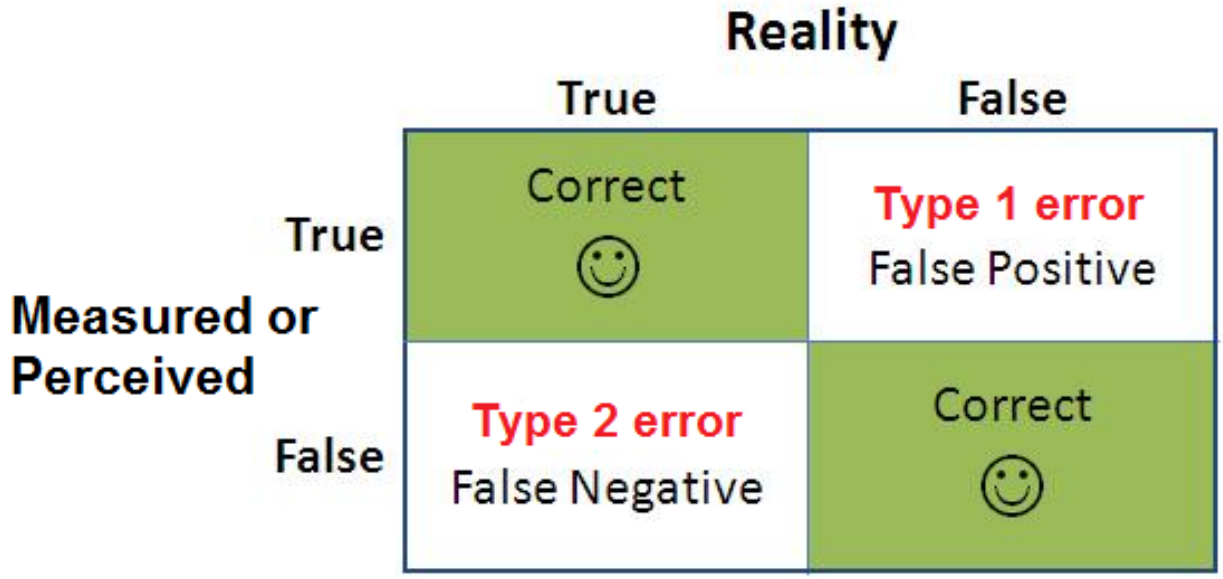

Figure 1: What could our results look like?

Let's see if we can further distinguish between true and false negatives.

Hiden Markov Models (HMMs)

Some genes are very specific for a species or have not been previously annotated, therefore we need to create specific Hiden markov model (hmm) profiles of the genes of interest.

1. Find the genes

In a first step you have to identify and find the genes of interest on NCBI. It is crucial that you choose the right gene. You do the following:

1. Ideally you refer to the paper and see if the name an accession number.
2. If you take this accession number and query it on NCBI in **all databases** mode.
3. Look at the search and study the domains in the **Protein family models** section.
4. Look at **Identical Protein Groups* section

5. Look at the **gene** section and identify the gene of interest. Download the nucleotide and protein fasta file and rename it with your gene Name.

Find your way around NCBI. It has many different *help* or *about* pages.

What you need to take from this section is one nucleotide and protein fasta file for every gene of interest. Remember to also fill in the google.sheets with the information.

2. Create a mutisequence alignement file

In order to create a HMM model we need to find multiple closely related forms of this gene so called gene family. From this we create a multisequence alignement (MSA) file for every gene-of-interest. With such a multisequence alignement file we can see conserved regions (often domains) and variable regions within a gene. We can also see all different variable sites that the genes from different species can have. There are many different ways how to find a gene-family and create MSA files. You will do either one of the two follow two:

2.1 OMA The best way to identify gene families of your genes-of-interest is to find them on a orthologous gene repository. One of these orthologous gene-family repositories is OMA

2.2 Blast However many of our genes are maybe not yet in the OMA browser so what we can do is the following:

1. blastn you genes-of-interest. Let's do it on protein level (=blastP)
2. Take the non-redudant (nr) protein database.
3. from the blast results do an alignment (online)
4. Download and label .msa or .aln file according to the gene-of-interest name
5. **Also download a genome that contains the gene! (positive control)**

Once you have created a folder with a .msa file for every gene-of-interest you can continue to build a hmm model. You should move this to the wally to the appropriate location. If you cannot remember how, have a look here:

```
##------copy something to wally
scp file.txt vsomervi@wally-front1.unil.ch:/users/vsomervi/scratch/vsomervi/ # a single file
scp -r directory/ vsomervi@wally-front1.unil.ch:/users/vsomervi/scratch/vsomervi/ # a complete director
```

3. Build HMM

In the next step we take the multisequencealignement file (msa) and creat the actual model. We therefore use a tool called hmmer. Remember on wally you need to load the tools before you can use them.

```
####---------------------------------------
##modules
####---------------------------------------

module load HPC/Software
#module load UHTS/Analysis/prokka/1.13
#module load UHTS/Analysis/barrnap/0.8
module avail
module load SequenceAnalysis/HMM-Profile/hmmer/3.1b2
```

Within the hmmer toolbox we will use the following two functions. See if you can understand what to do:

```
hmmbuild -h
hmmsearch -h
```

Here, you have some input for the hmmbuild part. Remember you should write a script and submit it to wally and not run it on the login node.

```
####--------------------------------------
##hmmbuild
####--------------------------------------
hmmbuild -h
hmmbuild ${Overall_output_directory}/capV.hmm /data/projects/p539_crisprscope/defense_mechanisms/MSA_OM
```

Here, you have some input for the hmmsearch part.

```
####--------------------------------------
##hmmsearch
####--------------------------------------
hmmbuild -h
hmmsearch  -o /data/projects/p539_crisprscope/defense_mechanisms/detection/CBASS/${speciesss}_${genomes
```