**CSC Bioweek. 11.3. 2019. Exercises part 1.**

# *1. Downloading data to Taito*

Log in to Taito server (taito-shell.csc.fi) and move to your $WRKDIR directory (Replace the *XX* with your training account number in all the commands in this exercise booklet).

```
ssh taito-shell.csc.fi -l training0XX
cd $WRKDIR
```

Create a new directory called *p_aeruginosa_XX* were *XX* is replaced by the number of your training account. Go to the new directory

```
mkdir p_aeruginosa_XX
ls -l
cd p_aeruginosa_XX
```

Use ensEMBL bacteria service (http://bacteria.ensembl.org/index.html). Locate the genome of *Pseudomonas aeruginosa PAO1*:

1. First use *Search for a genome* field to list the pseudomonas aeruginosa genomes.
2. Choose the first one from the list by clicking the name in the species column.
3. Continue to "Download DNA sequence"
4. The file we want to download is:Pseudomonas_aeruginosa_pao1.ASM676v1.dna.toplevel.fa.gz. Don't download the file to your local computer. Instead just copy the URL.

Now you can download the genome directly to Taito using *wget* command and the URL in Taito-shell:

```
wget ftp://ftp.ensemblgenomes.org/pub/bacteria/release-42/fasta/bacteria_13_collection/pseudomonas_aeruginosa_pao1/dna/Pseudomonas_aeruginosa_pao1.ASM676v1.dna.toplevel.fa.gz
```

Check what you have got with command:

```
ls -l
```

Unpack and rename the sequence with commands below. Hint: use *Tabulator* to use autocomplete feature when you are typing the long file names.

```
gunzip Pseudomonas_aeruginosa.PUPa3_1.0.dna.toplevel.fa.gz
mv Pseudomonas_aeruginosa_pao1.ASM676v1.dna.toplevel.fa p_aeruginosa.fna
```

Download the query sequence set R.fasta this directory:

```
wget https://object.pouta.csc.fi/materials/R.fasta
```

Check what you have downloaded with command:

```
ls -ltrh
```

Study the fasta files with commands

```
head p_aeruginosa.fna
tail -20 p_aeruginosa.fna

head -20 R.fasta

module load biokit/4.9.3

infoseq p_aeruginosa.fna
infoseq R.fasta
infoseq_summary R.fasta
```

## 2. Working with data columns

Submit a BLAST search using the Taito specific BLAST command line interface: *pb blast*

```
pb blastn -query R.fasta -dbnuc p_aeruginosa.fna -out pb_blast_results -outfmt 7
```

( If Taito is very loaded then it may take too long to wait that the job starts. If the job has not finished in 5 minutes, stop the process by pressing Ctrl+C. Then download the pre-calculated results with command:

wget http://materials.object.pouta.csc.fi/pb_blast_results )

Study the first rows of the blast result file by running command:

```
head –50 pb_blast_results
```

You can see that the result file contains columns, where the query sequences are in the first column and the hit sequences in the second column. You can now check, how many hits each query sequence got with command:

```
awk '{print $1}' pb_blast_results | grep –v "#" | sort | uniq –c | sort –k1n
```

Study how this command works by executing it step by step ( use command history browsing with arrow keys when adding new steps to your command):

```
awk '{print $1}' pb_blast_results
awk '{print $1}' pb_blast_results | grep –v "#"
awk '{print $1}' pb_blast_results | grep –v "#" | sort
awk '{print $1}' pb_blast_results | grep –v "#" | sort | uniq –c
awk '{print $1}' pb_blast_results | grep –v "#" | sort | \
uniq –c | sort –k1n
```

Finally, make a table file *pb_blast_results.tsv* file that contains only the data rows and not the comment lines. This can be done by using *grep* to select only those lines that do not contain “#” character:

grep -v “#” pb_blast_results > pb_blast_results.tsv

You can check the number of query sequences in the input file with command:

```
grep -c ">" R.fasta
```

Alternative ways to count number of sequences with EMBOSS:

```
seqcount R.fasta
infoseq_summary R.fasta
```

If you compare that to the number query sequences in the result file:

```
awk '{print $1}' pb_blast_results | grep -v "#" | sort | uniq -c | wc -l
```

You can notice that the commands used above show no information for those query sequences that did not get any matches.

**Extra task:**

Use linux commands to sort out those query sequences that did not produce any hits. There is no single right way to do this task. You probably need several commands and temporary files to get the result. You can also try linux scripting.

# 3. Using object storage

## 3.1. Command line in Taito

Re-run the previous BLAST run, but this time, save the results to file *pb_blast_results.html* as a html formatted traditional BLAST report (the default option). This can be done with command:

```
pb blastn –query R.fasta –dbnuc p_aeruginosa.fna \
–out pb_blast_results.html –html
```

If the pb blast run does not start immediately, then open a second terminal session to Taito and move to the *p_aeruginosa* directory:

```
cd $WRKDIR/p_aeruginosa
ls –l
```

The connection to Pouta Object Storage has already been configured for your training *account* (with your own account you must run configuration command *poutaos_configure* before you start using Pouta Object Storage for the first time)

Check what data buckets your Object Storage has with commands:

```
s3cmd ls
```

Create a new bucket with command (replace *XX* with your account number):

```
s3cmd mb –P s3://trng*XX*_test
```

Then upload the table formatted blast result files to this object storage bucket.

```
s3cmd put pb_blast_results s3://trng*XX*_test/pb_blast_results.tsv
s3cmd ls –l s3://trng*XX*_test/
```

Check that the *pb blast* run submitted earlier has already finished. Then upload the html formatted BLAST result file to the same bucket, but this time add option **-P** to the upload command to make the file publicly available:

```
s3cmd put –P pb_blast_results.html s3://trngXX_test/pb_blast_results.html
```

When the upload is finished, the command prints out an html address for the uploaded file. Change the *http* to *https* in this address and check if you can access this url with your browser.

Then try to access the table formatted blast result file by replacing the ".html" in the end of the URL with ".tsv".

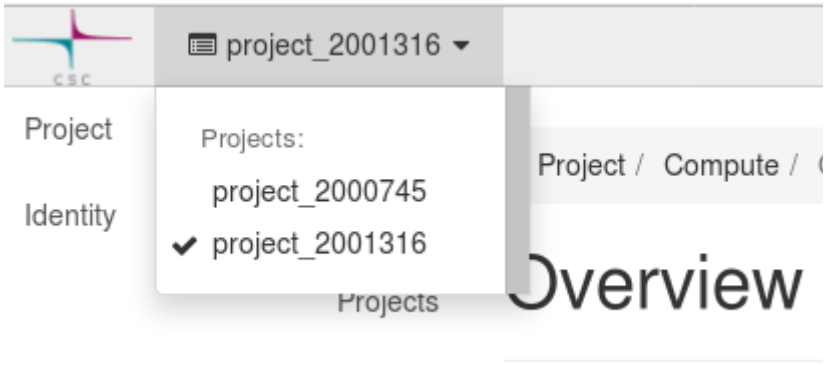
### 3.2. Pouta WWW interface

Next we use the cPouta interface to access Object Storage form your local machine.

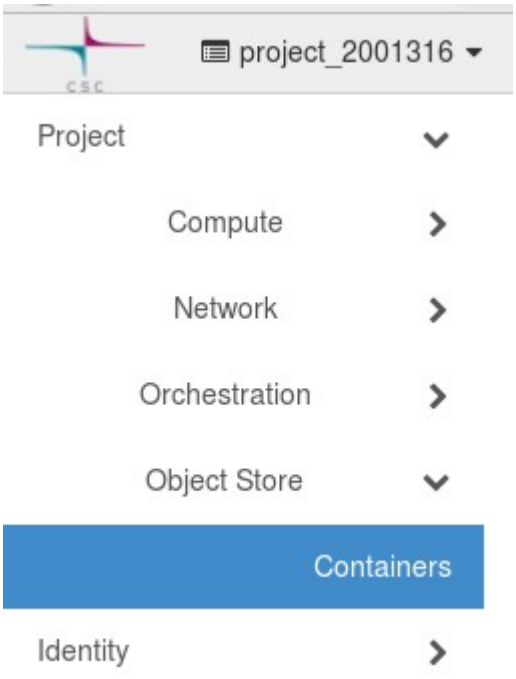
Use your WWW account to login to cPouta WWW interface in:

https://pouta.csc.fi

Once logged in, check that you are using project: 2001316




Then open the container list by selecting: Project -> Object Store -> Containers




You should now be able to see the container that you just created using the *s3cmd mb* command in Taito. Open your bucket and download the table formatted blast result file to your local computer and test if you can open it in a spread sheet program (excel, LibreOffice,...).

The use the Upload option of Pouta Object Storage 

to upload some file from your local computer to your bucket in  Pouta Object Storage. (Any file will do).

Then switch back to the Taito-shell session and use s3cmd commands to copy the file you uploaded to your bucket to your $WRKDIR directory.

## 3.3. Cleaning up

As a final step, remove the files you uploaded to object storage and the bucket you created:

```
s3cmd rm s3://trngXX_test/*
s3cmd rb s3://trngXX_test
```