Part 1. Prepare the working directory.

1. Create a working directory "/workdir/$USER". Copy all data files for this exercise from "/shared_data/alignment2020/project1/" into the working directory.

```
mkdir /workdir/$USER
cd /workdir/$USER

cp /shared_data/alignment2020/project1/* ./

ls -1
```

Part 2. Identify genes homologous to human SKCa1 from the honeybee genome.

In this exercise, you are provided with the following files:

- Honeybee genome sequence: bee.fna.gz
- Annotated proteins in the honeybee genome: bee.faa.gz
- Human SKCa1 gene in fasta format: skca1.fasta

You will use BLAST, HMMER and HH-Suite to identify genes homologous to skca1.

Manual for BLAST package: https://www.ncbi.nlm.nih.gov/books/NBK279684/.

Manual for HMMER: http://eddylab.org/software/hmmer/Userguide.pdf

1. **Run blastp program (protein against protein).**

   a. First using "zcat" command to de-compress the .gz file: bee.faa.gz. This file has all annotated bee proteins in FASTA format;

   b. Using "makeblastdb" command to make a protein blast database from the decompressed bee.faa file;

   c. Run "blastp", using human gene fasta file "skca1.fasta" as the query;

   d. Using "less" command to examine the content of the result file "bee_skca1.txt".

   (When running "less", press "space" bar to advance to the next page, or press "q" key to exit)

```
zcat bee.faa.gz > bee.faa
makeblastdb -in bee.faa -dbtype prot
blastp -db bee.faa -query skca1.fasta -out bee_skca1.txt
less bee_skca1.txt
```

2. **Testing different parameters of blastp.**

   Examine output with either "less" or "less -S" command. The "less -S" command truncate the lines to fit the page, which makes it easy to read.

a. First testing num_threads and "-outfmt 6". The result file "bee_skca1_tab2.txt" is a table that you can open in Excel. Each output row is an HSP, the 12 columns are 'qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore'.

```
blastp -num_threads 2 -db bee.faa -query skca1.fasta -outfmt 6 -out
bee_skca1_tab1.txt
less bee_skca1_tab1.txt
less -S bee_skca1_tab1.txt
```

b. Add stitle column to "-outfmt 6", also testing evalue cutoff and max_target_seqs.

```
blastp -num_threads 2 -db bee.faa -query skca1.fasta -outfmt "6 std stitle" -
evalue 1e-20 -out bee_skca1_tab2.txt
less bee_skca1_tab2.txt

blastp -num_threads 2 -db bee.faa -query skca1.fasta -outfmt "6 sseqid evalue
stitle" -max_target_seqs 5 -out bee_skca1_tab3.txt
less bee_skca1_tab3.txt
```

3. **Using the tblastn program (protein against nucleotide).**

a. First using "zcat" command to de-compress the genome sequence file bee.fna.gz;

b. Using "makeblastdb" command to make a nucleotide blast database from the bee.fna file;

c. Run "blastp", using the human gene fasta file "skca1.fasta" as the query;

d. Using "cut -f2,9,10,11" command to examine the columns 2,9,10,11 of the output file. Column 2 is the genome contig name, and column 9 and 10 are start and end positions of the HSPs on the genome contigs. You probably notice that some of the HSPs have the start position larger than the end position. In these HSPs, the query matches the reverse strand of the target.

```
zcat bee.fna.gz > bee.fna
makeblastdb -in bee.fna -dbtype nucl
tblastn -db bee.fna -query skca1.fasta -outfmt 6 -out bee_skca1_blastn.txt
cut -f2,9,10,11 bee_skca1_blastn.txt
```

4. **Using the HMMER package**

a. Identify what domains are on the *skca1* protein.

Open a web browser on your laptop, go to the site https://pfam.xfam.org/, and click "SEQUENCE SEARCH"

Copy-paste the protein sequence *skca1* into the search form, then click "GO". You will find out that top matched domain is "*SK_channel* (PF03530)"

(The protein sequence is in the file "skca1.fasta")

```
less skca1.fasta
```

b. Download the PFAM model file of "PF03530". You can use the web site to download the model. Alternatively, you can using the Linux command "wget" to download the model file.

```
wget -O PF03530.hmm https://pfam.xfam.org/family/PF03530/hmm
ls -l
```

c. Run hmmsearch program to identify honey bee proteins that contains the domain "PF03530".

Using the "export PATH=/programs/hmmer/bin:$PATH" command to add hmmer executables to the Linux PATH. After this step, you will be able to execute the "hmmsearch" command without specifying its full path.

The "hmmsearch -h" command is to print the help menu of "hmmsearch".

When running hmmsearch, if the "--tblout" parameter is used, you will get an output file in tab-delimited text file. This is useful if you have large number of output. The "hmm.tab.txt" is the output file we want. This is a table that can open in Excel, which show all proteins with the SK_channel domain, and evalue for each match.

```
export PATH=/programs/hmmer/bin:$PATH
hmmsearch -h
hmmsearch -o hmm.output.txt --tblout hmm.tab.txt PF03530.hmm bee.faa
less hmm.output.txt
less hmm.tab.txt
```

Part 3. Multiple sequence alignment

EBI has a web site (https://www.ebi.ac.uk/Tools/msa/) where you can run MSA online. But sometimes, especially if you need to align a large number of sequences, you have to do it on a local computer using a command line tool.

1. **Run MAFFT**

We will be using MAFFT, one of the most popular software to generate MSA. The input file is a fasta file. The MAFFT software has an option to output results either in aligned fasta format, or Clustal format.

You can run MAFFT through an on

You might want to read the MAFFT manual https://mafft.cbrc.jp/alignment/software/manual/manual.html , which listed commands to use in different scenarios. In this example, we will be using "MAFFT L-INS-i" (probably the most accurate; recommended for <200 sequences; iterative refinement method incorporating local pairwise alignment information).

You will run the commands twice, generate two different output formats, Clustal and FASTA.

```
export PATH=/programs/mafft/bin:$PATH

mafft --thread 2 --amino --localpair --maxiterate 1000 --clustalout
skcal_species.fasta  > aligned.clustal

less aligned.clustal

mafft --thread 2 --amino --localpair --maxiterate 1000 skcal_species.fasta  >
aligned.fasta

less aligned.fasta
```

2. **Run gblocks to remove regions of unreliable alignment**

   This trimming step is recommended before you use the MSA results for phylogenetics analysis. Gblocks is one of the software developed for this purpose.

```
export PATH=/programs/Gblocks_0.91b:$PATH
Gblocks aligned.fasta -t=p -b5=h
less aligned.fasta-gb
```

After Gblocks, you will find a new file aligned.fasta-gb, which is a trimmed MSA file.

Gblocks parameters (For details, read http://molevol.cmima.csic.es/castresana/Gblocks/Gblocks_documentation.html):

- -t: p, t or c, which represent protein, DNA or codons. The "c" option is to make sure no frame shift after trimming if your input sequences are cDNA.
- -b5: n h or a. n: no gaps allowed at each position; h: half of the positions can be gap; a: all gaps allows

3. **(Optional) Visualize or editing MSA result on your laptop**

AliView (https://ormbunkar.se/aliview/) is one of the software developed for visualizing MSA file. First you need to download the software to your laptop. Windows users can download this jar file: https://ormbunkar.se/aliview/downloads/windows/windows-version-1.26/without_installer_version/aliview.jar Then you can open the software by clicking this file. If you do not have Java installed on your laptop, you will need to install JAVA first. https://www.java.com/en/download/help/windows_manual_download.xml

Mac users can follow instructions here: https://ormbunkar.se/aliview/#DOWNLOAD

Using Filezilla to download the file aligned.fasta to your laptop and open the file in AliView.

Part 4. Parallel computing

When we work with large data files, quite often, a job could take hours or even longer time to finish. For jobs like this:

1. Run jobs in "screen", (https://biohpc.cornell.edu/lab/doc/Linux_exercise_part2.pdf).
2. Run jobs in parallel if possible.

In this exercise, we will be using GNU parallel to run BLAST.

Instructions for run BLAST in parallel can be found at:

https://bioinformaticsworkbook.org/dataAnalysis/blast/running-blast-jobs-in-parallel.html#gsc.tab=0

First start "screen", we will run all commands in the session created by screen. The advantage of running commands in "screen" session is that even if your laptop gets disconnected, the job will still be running on the server. You can re-attach back to the session at any time.

```
screen
```

In this case, we will run blastp using the same file "bee_partial.faa" both as query and as target database.

With this command, the fasta file "bee_partial.faa" is broken down into many 5kb-size blocks (specified by "--block 5k"). These blocks of sequences will be piped into 2 simultaneously running blast processes (as specified by "-j 2"). The "--recstart '>' " parameter is to make sure that a single sequence record would not be separated into two different blocks.

During the hands-on session of this workshop, you will be sharing this 8-core computer with three other people, that is why you can only set "-j 2". When you do real data analysis, you need to set "-j" (jobs) much bigger, but make sure that the value of "number-of-jobs x num-of-threads" not to exceed the total number of CPU cores of the computer you are running.

```
zcat bee_partial.faa.gz > bee_partial.faa

makeblastdb -in bee_partial.faa -dbtype prot

cat bee_partial.faa | \
parallel -j 2 \
--block 5k \
--recstart '>' \
--pipe blastp \
-num_threads 1 \
-evalue 0.01 \
-outfmt 6 \
-db bee_partial.faa \
-query - > combined_results.txt
```

While the job is running, you can safely detach from the "screen" session by pressing "ctrl-a" followed by "d".

After detaching from "screen" session, you can use the "top -u $USER" command to monitor the progress of the job. After the job is finished, using "less" to examine the output file combined_results.txt.

To re-attach back to the screen session, use the command "screen -r".