# Exercise 1 – Multiple Alignment using NCBI BLAST online

**Objectives:**

- to become familiar with the NCBI BLAST online system
- to make use of some very basic multiple alignments options there.
- to learn to download the aligned proteins for future analysis.

**Our Test Protein:**

This protein sequence comes from a 1000-year old skeleton, found at a monastery in Germany. The protein sequence is the translation of a short open reading frame found on a piece of DNA from that skeleton. More precisely, the DNA was from calcified dental calculus ("tooth-plaque"). We'll work on this short protein for the next couple of exercises. You can use 'copy-and-paste' to copy it from here:

```
GFFGDRVGRKFIIWFSILGTAPFALWLPYADADTTAILVILIGFIISSAFASILVYSQELLPKKIGMISGV
FYGFAFGMGGLASALLGKLIDLTDITFVYKVCSFLPLMGLIAYFLPNLRKVKMKE
```

**1) submit the protein for a BLAST search**

when confronted with an unknown protein, a good idea is to first search a sequence database online, to find out whether similar proteins have been described already.

- open your browser (Chrome/Firefox), and take it to the NCBI website: https://www.ncbi.nlm.nih.gov
- towards the right side of the page, under "Popular Resources", click "BLAST".
- Under the "Web BLAST" section, click on "Protein Blast".
- now, copy-and-paste our test-protein from above, and paste it into the big search box. Scroll down the page, and click the "BLAST" button – database and algorithm choices are already correct.
- wait until the results show up. Scroll down the page a bit, to get to the detailed overview of 'similar' proteins that are already known.
- let's see what we can find out about our query protein. Click on the second similarity hit that is reported ('MFS transporter […]'). This will bring up your first protein alignment … so far, it is not a multiple alignment but just an alignment between two proteins (our "query", and a similar protein found in the database of known proteins ["Subject"]).
- From which organism is the Subject protein? Is that an organism that we would expect to see on the surface of a tooth? Can you translate its Latin name, or at least parts of it ? … ☺
- how long is the subject protein? Longer or shorter than our test protein? Go back and look at a few other subject proteins. What seems to be a typical length for this protein family? Can you think of any reason why our query

protein might be shorter than the typical family member? Remember under what circumstances we found it …

- using Google, try to found out what “MFS transporter” means. Why would this perhaps be an interesting protein to study?
- now go back to the top of the page. There should be a link called “Multiple Alignment” … follow that one now (you may have to wait some time for the results to show up) and then scroll down a bit to the section “Alignment”. Voilà – this is your first multiple sequence alignment. Our test protein is shown at the very top … try to look around and understand what you see. What do the positions labeled ‘-‘ mean? Sometimes there are square brackets with numbers ‘[20]’, what might those mean?

  Hint: compare amino acid sequence length of different proteins in each block

- Is our protein the only protein that is shorter than the others? It looks like there are some other shorter proteins known as well. Is there a part that in the alignment that every protein indeed covers?

  *[indeed there is, towards the end. This is the most ‘conserved’ part, and hence likely the most important and characteristic part of the family. It forms part of a protein ‘domain’, as we will see in the next exercise.]*

  You can find more of multiple sequence alignment result explanations in at https://www.ncbi.nlm.nih.gov/tools/msaviewer/

**2) download the aligned sequences.**

- BLAST has done an alignment for us, but what if we want to use a different algorithm and/or different parameter settings? For that, we need to download the protein sequences and work with them locally on our computer.
- in the BLAST multiple alignment page, towards the top, there is a link labeled ‘Download’. Follow that link, and then select “Fasta plus gaps”. You will be offered to save the alignment into a file and rename them as “`input_proteins_1.fa`” (the file-ending ‘fa’ stands for ‘fasta’, a frequently used file format for storing biological sequences).
- store the file into a suitable location on your laptop.
- now take a look at the file you just saved: open a command prompt (either gitbash or Terminal). There, go to your folder and look at the first few lines of the file:

```
cd [your directory name]            [brings you to your folder, if not already the case]
head -n 30 input_proteins_1.fa      [prints the first 30 lines of the file]
```

# Exercise 2 – Domain Analysis using 'Interpro' online

**Objectives:**

- to roughly understand what a 'protein domain' is
- to learn to discover and browse domain information using InterPro.
- to learn how to download the 'seed' alignment sequences for a given domain.

**Our test protein:**

same protein as in exercise 1, from the 1000-year old skeleton:

```
GFFGDRVGRKFIIWFSILGTAPFALWLPYADADTTAILVILIGFIISSAFASILVYSQELLPKKIGMISGV
FYGFAFGMGGLASALLGKLIDLTDITFVYKVCSFLPLMGLIAYFLPNLRKVKMKE
```

**1) What is a 'protein domain':**

in short, a *protein domain* is a short stretch of multiple sequence alignment that somehow seems 'important' enough to be given its own name and annotation. Most proteins typically consist of one or more protein domains, but also of non-domain sections that align less well and may perhaps be less important or at least less diagnostic of function.
In practice, protein domains are discovered by experts staring at alignments all day. As part of describing and naming a new domain, these experts will usually provide a summary on where the domain can be found, and what function it might have. Within the actual proteins, domains often represent autonomously folding structural subunits. Many proteins are consisting of multiple different domains, which can be rearranged and exchanged over evolutionary timescales ('domain shuffling'). As of today, most domains in existence have probably been discovered and described already.

**2) submit the test protein for a domain search**

after exploring an unknown protein against sequence databases (exercise 1), the next typical strategy is to search it for previously described domains. This may provide a broader idea of its function, and often also some three-dimensional structure information.

- open your browser (Chrome/Firefox), and take it to the EBI website: https://www.ebi.ac.uk/
- Click on "Services" (on the very top of the page), then type "InterPro" into the search box on the newly opened page and navigate to the 'InterPro' page.
- Use copy-and-paste to enter our test protein into the box '`Search by sequence`', and click 'Search'. You may need to open the 'Advanced options' section once

to make the 'Search' button active and available. No settings need to be changed.

- after a while, your results should become available, and you can first click on the jobId and then on "Sequence1" to get a graphical summary, indicating domains and features that have been found for our test protein. First choose "Options" and then "Color by" (drop-down box just above graph) "Domain Relationship". Then, under "Feature Display Mode", choose 'Full'.

  At the bottom of the page, you can see the predicted Features section. Hover your mouse over grey lines in this section, you will find at least four of them are likely transmembrane regions, so the protein may be sitting in a membrane. The colored lines above are the actual domains. They refer to the very same domain several times, but some of the classifications are more generic ('family'), whereas other ones are more specific ('domain').

- now, click on domain accession 'IPR020846' on the right of first line under "Domains". This brings up the so-called 'Interpro-abstract', which provides an excellent summary of the general function of this protein family. As you will see, it is a transporter, which the cell uses to transport a variety of 'small molecules' across the membrane. Now we know a lot more already, but of course the annotation is too generic to let us know which is the preferred 'small molecule' that our protein wants to transport.
- now, notice that in the annotation, it said that these proteins typically have 12 trans-membrane regions, but we only found four … this is another indication that our test protein from the skeleton is indeed incomplete. Hence, let's repeat the analysis with the closest relative it has in the sequence databases (this is the 'best hit' we found in exercise 1)
- Copy-and-paste the first best hit found in the file "`input_proteins_1.fa`" (the file generated from exercise 1). It should be the second entry in that file. Submit this protein to InterPro, as before. How many transmembrane sections do you find now? (take care to remove the gap characters ["-"] before submitting)
- Next, let's find out a bit more about this domain. Proceed to the Interpro abstract again, like before ('IPR020846').
- On the left of the abstract, click on 'Taxonomy' … this will tell us in which organisms the domain is found. You will find that it occurs virtually everywhere (Bacteria, Eukaryotes, Archaea).
- finally, click on the 'Structures' section. This will provide examples of known three-dimensional structures. Select one entry from the 'PDBe' section, and look at the protein structure. Now we know how our protein generally looks like … but again no useful information about the substrate (in this case, the proteins with the solved structure transport mainly sugars).

**3) download a 'seed' alignment from Pfam**

The Pfam database (= "Protein families") is similar to InterPro, but is actually one of the original databases for protein domains (in contrast, InterPro is a 'meta-resource' bundling several original databases such as Pfam).

For many of its domains, Pfam maintains a 'seed' alignment, which is the original alignment that was made by the discoverer of the domain (or an updated version thereof). It is made up of non-redundant, representative sequences, and often has been quality-checked manually. For further work on alignments and trees, we'll get one such seed alignment from Pfam now.

- Copy-and-paste the first, best hit found in the file "`input_proteins_1.fa`" (the file generated from exercise 1). It should be the second entry in that file. Submit this protein to InterPro, as before.
- now, click on Pfam accession 'MFS_1 - PF07690' on the right of second line under "Family".
- Next, on the left, find "Alignment", then from the "Available alignments" box choose "seed (192)". Click the Download button to download the seed alignment file.
- Uncompress the downloaded file on the command-line, with command "`gunzip PF07690.alignment.seed.gz`" and move it to a folder on your laptop.
- one thing to notice here: the downloaded file is a "STOCKHOLM"-format file instead of the FASTA format file that we will need. We thus use python to transform the downloaded file to FASTA format. Download the python script "`stoTransform.py`" from OLAT or from github to your laptops folder.
- the script requires the "biopython" module, so let's install that. Make sure your python environment is active, then use this command:

  ```
  pip install biopython
  ```

- after that, we can run the stoTransform.py script:

  ```
  python stoTransform.py -i PF07690.alignment.seed -o input_proteins_2.fa
  ```

- In case the InterPro website happens to be down or too slow, please feel free to proceed directly to Exercise #3 and use the file "`input_proteins_2.fa`" that we have provided through OLAT.

# Exercise 3 – Multiple Alignment on your Computer

**Objectives:**

- to perform a multiple alignment of proteins using three different programs.
- to prepare the input files, bringing them into the right format.
- to check whether the results are identical.

## 1) Prepare the input files.

In the last two exercises, we've made two files with protein sequences already. One of them came from BLAST, it will form the basis for an 'easy' alignment because the proteins are very similar. The other is from a domain database, this will be the 'hard' alignment because the proteins span a very large area in sequence space. We'll use those input files to create six different multiple alignments; but first we'll have to properly prepare the input. Both input files should be un-aligned, and for one of them we also still need to add our test protein.

Prior to the course, the files "`input_proteins_1.fa`" and "`input_proteins_2.fa`" were unaligned using regular expressions. The character "-" was replaced with empty character "" and empty lines were removed. This removed all the gaps, and hence destroyed the alignment. This was saved into "`unaligned1.fa`" and "`unaligned2.fa`" respectively; the files are available on OLAT and via github.

- download the files "`unaligned1.fa`" and "`unaligned2.fa`" from OLAT and store them to your local disk (alternatively, you may also retrieve them from the github repository).
- now, to deal with our test protein: in one of the files, it simply needs to be renamed … and in the other file it is missing, so we need to add it.
- open the File '`unaligned1.fa`' in an editor of your choice, and change the name of the very first protein, to name 'query_protein'. Then save the file. The first lines of the file should now look something like this:

```
>query_protein
GFFGDRVGRKFIIWFSILGTAPFALWLPYADADTTAILVILIGFIIS
SAFASILVYSQELLPKKIGMISGVFYGFAFGMGGLASALLGKLIDLTDITFVYKVCSFLP
LMGLIAYFLPNLRKVKMKE
>ref|WP_002666381.1| MFS transporter [Capnocytophaga gingivalis]
METKQRTQYLIIILISL
SHCLNDLLQGVLPSIYPALQSKFALSMAQIGLITFCYQIAASILQPIVGAYTDKHPKPYA
QVVGMAFSALGIGLLSWVDSYTLVLCSVVFVGIGSSIFHPEASRISFLASGGKRSFAQAV
[…]
```

- open the second File '`unaligned2.fa`' in an editor of your choice, and add the query protein sequence to the beginning of the file (simply copy-and-paste from the example output above).
- *optional extra task for the geeks among you: can you do the above file manipulations using the unix-editor "**`vi`**" as your editor? If you can't, you're not a geek … ☺*

**2) Multiple alignment using Clustal Omega.**

- Open your browser (Chrome/Firefox), and take it to the EBI Clustal Omega website: https://www.ebi.ac.uk/jdispatcher/msa/clustalo
- then on the Clustal Omega web page, click 'choose file' and choose the file '`unaligned1.fa`' from your computer.
- then click "submit" button at the end of the page
- after switching to the result page, click the "Download" button to download the aligned file and rename it as '`aligned.clustalo.easy.aln`'
- now let's make a nice graphical overview (for example to print it out later). Go back to result page. Choose "Results Viewers" at the top, then click "Sent to MView" at the bottom. On the MView input form, under "Parameters", choose an "Alignment Width" of 120, then click "Submit" at the bottom. Finally click "Download" and rename the downloaded pure html file as '`aligned.clustalo.easy.html`'
- repeat the exact same procedure for the second input file. This time, name the output file as '`aligned.clustalo.hard.aln`'. The alignment is more difficult; it should take around five to ten minutes. Again, create a nice graphical overview ('`aligned.clustalo.hard.html`').

**3) Multiple Alignment with Muscle on the Command Line**

the program 'muscle' is somewhat faster and often produces better alignments than ClustalX in benchmarks, so it is often preferred for larger and/or more difficult alignments.

- Let's install 'muscle' on your local computer:

  on Windows:

  ```
  curl -L -o muscle https://github.com/rcedgar/muscle/releases/download/v5.3/muscle-win64.v5.3.exe
  ```

  On Mac:

  download one of the two binaries below, depending on your CPU:

  ```
  curl -L -o muscle https://github.com/rcedgar/muscle/releases/download/v5.3/ muscle-osx-arm64.v5.3
  curl -L -o muscle https://github.com/rcedgar/muscle/releases/download/v5.3/ muscle-osx-x86.v5.3
  ```

  then, run this command to make the file executable:

  ```
  chmod +x muscle
  ```

- now, we can run muscle on both of our input files (make sure you are in directory directory "`~/Documents/Bio334_Data/`" before running following command):

  ```
  ./muscle -align unaligned1.fa -output aligned.muscle.easy.fa
  ./muscle -align unaligned2.fa -output aligned.muscle.hard.fa
  ```

- ok, let's use Mview to format this new alignment in a pretty way again: first go to the URL: https://www.ebi.ac.uk/jdispatcher/msa/mview; then click "choose file" button to select the "`aligned.muscle.easy.fa`" file we just made; set the alignment width to 120, then click "Submit" at the bottom. Finally, "Download" the generated html file and

rename it as '`aligned.muscle.easy.html`'. Do the same for the other file ('`aligned.muscle.hard.fa`') and save file as '`aligned.muscle.hard.html`'

### 4) Multiple Alignments using HMMAlign

the program HMM-align produces very good alignments, but it can only be used if the proteins to be aligned have a previously known domain.

- Let's install HMM-align on your local computer:

  on Windows:

  download the two files precompiled executable to a directory of your choice:

  ```
  curl -L -o hmmalign https://string-db.org/bio334/hmmalign_windows
  curl -L -o cygwin1.dll https://string-db.org/bio334/cygwin1.dll
  ```

  On Mac:

  choose one of the two binaries below, depending on your CPU:

  ```
  curl -L -o hmmalign https://string-db.org/bio334/hmmalign_x86_mac
  curl -L -o hmmalign https://string-db.org/bio334/hmmalign_arm64_mac
  ```

  then, run this command to make the file executable:

  ```
  chmod +x hmmalign
  ```

- **Optional** on Mac/Linux (but not on Windows) you can also choose to compile the source code by following commands in exact order (this will take some time):

  ```
  curl -OL  http://eddylab.org/software/hmmer/hmmer.tar.gz
  tar zxf hmmer.tar.gz
  cd hmmer-3.4
  ./configure -prefix=[your_directory_here]
  make
  make check
  make install
  ```

- next, we need a so-called HMM-file, which describes all the knowledge that has been assembled for a given domain. In our case, follow the steps below:
- Go to this link: https://www.ebi.ac.uk/interpro/entry/pfam/PF07690/
- Next, on the left, find "Profile HMM", then towards the top of that section find the link to download the raw HMM file, then story it onto your laptop. Uncompress it with command "`gunzip PF07690.hmm.gz`" and give it the filename "`MFS_1.hmm`"
- now, we can use the hmmalign commend to produce the alignments (make sure you are in a directory where both the executable and the input files are stored:

  ```
  ./hmmalign --outformat Stockholm MFS_1.hmm unaligned1.fa > aligned.hmmalign.easy.sto
  ./hmmalign --outformat Stockholm MFS_1.hmm unaligned2.fa > aligned.hmmalign.hard.sto
  ```

- One thing to notice here is that that hmmalign uses both lower-case and upper-case residues, and it uses two different characters for gaps.  In a match column, residues are upper case, and a '-' character means a deletion relative to the consensus. In an insert column, residues are lower case, and a '.' is padding.  However, most software tools will

only accept input with upper'case letter and "-" character as gaps. We thus use python to further modify the output to transform all '.' to '-' and all amino acid letters to upper'case.

- Download the python script "`stoTransform.py`" either from OLAT or from github, and store it locally.
- the script requires the "biopython" module, so let's install that. Make sure your python environment is active, then use this command:

```
pip install biopython
```

- after that, we can run the stoTransform.py script:

```
python stoTransform.py -i aligned.hmmalign.easy.sto -o aligned.hmmalign.easy.fa
python stoTransform.py -i aligned.hmmalign.hard.sto -o aligned.hmmalign.hard.fa
```

- as before, upload the alignments ("`aligned.hmmalign.easy.fa`" and "`aligned.hmmalign.hard.fa`") that you just created to the Mview website, and create visual representations for them (save them accordingly).

**5) compare the alignments**

You should now have six different alignments: two each from the three different algorithms (one easy, one hard). Compare them side-by-side … are there differences? If so, which of these alignments looks 'better'? What criteria might be useful for deciding that?
As expected, the 'easy' alignments overall appear to be more similar to each other. With one exception: the hmmalign may put our query protein in the first half of the alignment, whereas the others usually put it in the second half (!). Any idea what this might mean?

# Exercise 4 – cut out a domain of interest before aligning

**Objectives:**

- to learn how to identify domain positions in a protein with 'hmmsearch'.
- to cut domains from the input proteins before aligning, using Python.

**Introduction:**

This exercise is somewhat more difficult than the first three, and it involves some Python coding. It should be stressed that there are much simpler ways to cut out domains from proteins, but we'll use Python here while the memory is still fresh from the Python introduction. Also, the generic task of parsing some large data files and manipulating them is frequently encountered in Bioinformatics, so this is a good 'real-world' training case for Python.

- on the OLAT course-website and on github, there should be the two files below. Please download them and upload them to your working direcotry

| | |
|---|---|
| `mfs_domain_proteins.fa` | [this is a collection of 15 bacterial proteins, which have at least three domains each: one of our MFS-1 domains, and two enzymatic domains. From each, we want to cut out and align the MFS domain only, and discard the rest] |
| `MFS_1.hmm` | [this is a 'hidden markov model' describing how to identify "mfs" domains] |

**1) identify the MFS domains with 'hmmsearch':**

- next, will use the 'hmm' file to find the positions of each MFS domain in the set of proteins. We'll run the '`hmmsearch`' utility to scan the hidden markov model along the sequences and to report any hit it finds.

- Let's install HMM-search on your local computer:

  on Windows:

  download the two files precompiled executable to a directory of your choice:

  ```
  curl -L -o hmmsearch https://string-db.org/bio334/hmmsearch_windows
  curl -L -o cygwin1.dll https://string-db.org/bio334/cygwin1.dll
  ```

  On Mac:

  choose one of the two binaries below, depending on your CPU:

  ```
  curl -L -o hmmsearch https://string-db.org/bio334/hmmsearch_x86_mac
  curl -L -o hmmsearch https://string-db.org/bio334/hmmsearch_arm64_mac
  ```

  then, run this command to make the file executable:

  ```
  chmod +x hmmsearch
  ```

- then, run HMM-search as follows:

```
hmmsearch --domtblout domains_found.tsv MFS_1.hmm mfs_domain_proteins.fa
```

```
ls -lart                                [list the contents of the directory. there should be a new file]
head domains_found.tsv                  [check the first few lines of that new file; the domain positions are in 'env coord']
```

**2) use Python to cut out the domains:**

- now, we have all the information we need: one file with the protein sequences, and a second file containing the domain coordinates on these sequences.
- here is the challenge: can you write a Python script that uses this information to cut out the domain sequences and to print them into a third file?

*hints: first, read the protein sequences, and store these into a dictionary (one 'string' of amino-acids per protein name). Then, read the second file and parse the coordinates. Whenever you've parsed one valid line from the second file, retrieve the corresponding protein sequence from the hash you've made earlier, and cut out the part that contains the domain. Use the 'substr' function in Python to cut out the relevant part of the string. Then, print the substring and proceed to the next line.*

- the above task is about as difficult as it ever gets in terms of file-processing for Bioinformatics. So, do play around a bit with the task. Try to write a Python script that does at least part of it … and don't give up too easily … ;)
- when you do get stuck, on the OLAT course website or the github you will find the solution (which you can download and run at any time):

  first, download the python script "cut_domains_from_proteins.py" to your local directory. Then:

  ```
  python cut_domains_from_proteins.py > input_proteins_mfs_domain_only.fa
  ```

- finally, using Muscle and Clustal Omega as in the exercises before, align and visualize the proteins. You should now get a very nice and very compact alignment.

# Exercise 5 – Analysing and aligning newly discovered proteins

**Objectives:**

- to apply what has been learned today

**Anonymous Test Proteins:**

below, we provide 20 randomly chosen proteins. All have been derived from DNA on the teeth of ancient skeletons found in a german monastery (same as for the previous exercises). None of the proteins have been analyzed in detail before … Please select arbitrarily one of the proteins below, and analyze it like we did in exercises #1 through #3.

Optionally you can also study some protein sequence related to SARS-CoV-2, which are listed at the bottom below.
Questions:

- what protein family does your protein belong to?
- which domain(s), if any, does the protein contain?
- from which organism is it, likely?
- what function might it have?
- is it complete?
- how can it be best aligned to other members of its family?

```
>NODE_4178_length_1047_cov_6.240688_S6
SATSAAAMKLIAPSWPSRYVSPPRRRQGAAMAEWWSAQLVDRDGRIRGELPDIRGGSLEW
NISSAVRTGGSVEFAEPPSAGIDWVTTRIRILHHDGAEVRPMGVYRASWPNRKLRDGHTS
STLKLEAPTSRLRSQLGYWTQYEAGIVVTDRVAMTLRQLGESQLALTPSPQTLRTPLTWD
PDKTWGTLYSELLDAIGYGGIWCDANGWWRAAPYVAPMERPLAATYGGDPADYRCRTTYG
DEADWTDVPNRVLLYTRATSEAPALTSEVWITDPANPWHPDRVGPHTRCEAVEATSQEVL
DAKAKRLLAEGQERSRYITWTHPVDDTTLGDRVRIRRLGLDVAIEARK

>NODE_25515_length_1898_cov_8.371970_S6
RGGKMKIIIAGIGNIGAGLAGRLLNEGHDIVLVDRDIDRLEYNEETQDVMTVKGTCAAME
TLRKAGVEDADLLITATSSDEKNLLSCMTAHGMNPNIKTVARVRMQEYLETTSVFGEKFG
LSMIIDPGMYAAKDIEAILTYPGFLHRERFTKGMTDVVEAELHPESELCGKPVSAIQEIT
GSGALVCVVKRGDTAITPGRDFILREKDRIYVTAEEEDLSTLLRLFGKKKETVEKVMIVG
GGRIAKGLIPRLQKEGMEIIVIDTDKEICEELAMEFPKVNIVHGDGRKFALLERKNVREQ
DALICLTNDDESN

>NODE_60099_length_1027_cov_6.267770_S6
RVASVCLSTIVAVWMTSLIVPWASYSVKEGSRWAIESMYESRMVTKDDRDAFNWLAKQPH
AYDGIIFGNSADGYGWMYAYNKLPSLARHYDGVSAKPGAPSHVLRDSAYLIGAGNHGDPD
QRNRADLAAENLGVNFIMLSPPNFWWFQQSNLEMSAKLDKAPGLTLVYQKNSIRIYAVNA
KFKDAELTRMRASGPASNQLPVPQCPKDSADGKAAATAGETTQVEYDPDTGEQTTVTKPK
PCYHRPSKPDIPPRANDAKGKTPATPKSGGGDDKSNKYSEKTGLDLTEKEARRRRLDNGY
VHNEKATLRF
```

>NODE_77700_length_886_cov_21.930023_S6
WANMCPRKCKVMSMKRNQSAVHQLITYGMVIAAYILCQILVENGSMTRSLKGQLIPIAVY
IVMAVSLNLTVGISGELSLGHAGFMSVGAFSGIVVSQWMGTVYPNVHVYVRLVFAIVTGG
IAAGIAGVLIGIPVLRLRGDYLAIVTLAFGEIIRNIMNLLYVSVDQGRLRMAFNDGALPG
EQVIAGPKGAVGIEKIATFTMGFILVMITLFVVLNLINSRSGRAIMAIRDSRIAAESVGI
NVTKYKMMAFVISSVLAGMAGALFGLNYSTVSAGKFKFDMSILVLVFVVLGGIGNIRGSV

>NODE_87482_length_1095_cov_5.276712_S6
NPLIARTRGQQRDAHSVHARYGDKYLPFSDLENSMRDMEGLLNKVADLAVKAGSIMLSDS
DVEVGNKGTKENYVTSTDLKVQRFLREGLATLLPGAVFRGEEDDLPREDEGTRGEYVWIV
DPIDGTANYARGFGESAVSIALAKDDEPVLGVVRNPYARETYCAIKGRGAFLNGTPIHVS
GRSKENAMICLSWSAYDKSRSADCFRISQDLYAVCEDIRRTGSAAYELCLLARGSVDMHF
EIRLAPWDYAAGGLIIEEAGGRTGSLEGRLDMRRQCLVMAANSEKNFAFLKGVVSENLSL
RRRLAPVHV

>NODE_107984_length_1345_cov_80.271378_S6
ERKAYSMGKRTIIPFGPQHPVLPEPVHLDLVIEDETVVEAIPSIGFIHRGLEKLVEKKEY
PEMVYVIERICGICSFGHGWGYCAAVEGAMNVEIPERAMYLRTILHELGRMHSHLLWLGL
LADGFGFESLFQHCWRIRETVLDLFEQTTGGRVIFSICKVGGLNKDIDNETLNKIVKTLR
GIEKEIREYTSVFINDTSVKNRLTGVGVLSREDAEALCTVGPMARASGLRQDMRLAGEGK
YLELGFEPVLEEAGDCMARCKVRIGELLQAIDIIEKAVAQIPDGDIAVAVKGNVDGEFIN
RLEQPRGEAFYYCKGQGTKFLERIRVRTPTNMNIPAMVKILQGCDLADVPMIVLTIDPCI
SCTER

>NODE_123020_length_4291_cov_7.623631_S6
AVFEERWGDRPFMRSYRIPSIPVRPIWICVSRQNRAVLCLKTYIQMEQAILGAKREPVCQ
AASHALGPSAEDSCLTARPDPMRVDYDTDVRAFAQRLLGGNVFEPVTFAGITLPLISFIL
FGAALAFLLIVQVARTMISNKLQNLFASKLYDEFLDTVDEPLTRFFIPAYNRTYLRLNAF
MAKGSVEKAMEAFDQLLAMRSTRAQRDDLLFKAFQFYMQQEDFKGAKAVLDEMQSYGRHE
KRVEECVQAYEIFGNNSYAYIDEMEAAFDEAPYALKVSYALMLAAQYTSKKDGEAAEKWQ
DTARELLENPPKKGPAETR

>NODE_182329_length_1939_cov_4.566271_S6
APDDPPRHRRREQREKLFRRTTCLHPWGRVLLRGDHGAAQRRSTRAYRTASQTARREKVR
DHRAAARSGRARPAARSGARRGATGGYRELGGKRAVRCRAVRRPRIRVLGRPRGRLRAHH
GRNRPSERALLPSRSRHLRSRVGRTPHRTRNALLYRAYRTGELHDCRPGSNGARVRGKHR
ATAQRGICRMTALRSIALAFTLFSRVPMPHVEWNPENMRYTMLAFPLVGCVIGTAVATWC
ALCATLGLNGAAFGAGTVLVPLFVTGGIHMDGFADVVDAQSSHAAPERKREILADPHIGA
FAAIGIGGYLLAWAALAS

>NODE_212586_length_1033_cov_30.919651_S6
ISKTDESYPDFLRPSDGALHPAVNEYRSLWISLSLKGALPGLYPIHIVVEQDGEECYRAT
LCVRVCTAPLEKQKLIHTEWLHADCLCSYYNVEAFSERHFALLENFIRAAVQDYGINMIL
TPVFTPPLDTQVGGERRTVQLVDIACDSRGYHFDFSKLARWADICKRCGVEYLEIAHLFT
QWGAQHCPKIIVTEKGRERKKFGWQSDAAGTEYRKFLEQFLPALRSALQGMGYPDEKVYY
HISDEPSEDNLEHYRRAKAQVADLLEGANVVDALSSYRFYQEGLVTEPIVSSDHIQAFLD
AGVPNLWVYYCCGQDKLVPNRFFAMPSPRNRVFGVLLYLSGVKGFLHWGYNFY

>NODE_238737_length_1166_cov_7.374785_S6
NRHQTMFKGEIVMNSLIIVSAALGLCALLFALVLAARVKSQDSGTERMTEIAAYIHQGAK
AFLMAEYRILVIFVAILFVLIGLGISWITAVCFLVGAAFSTVAGYIGMNVATAANVRTAA
AAKDKGMNAALSVAFSGGAVMGMCVVGFGLLGASLIYFVTGNSEILSGFSLGASTIALFA
RVGGGIYTKAADVGADLVGKVEAGIPEDDPRNPAVIADNVGDNVGDVAGMGADLFESYVG
SVVSAVTLGLVAYNQEGAVFPLLIAALGIGASIIGSFFVKGDEKSSPHKALKFGSYASSV
LVAVGSLALSYKFFGNLNAGMAIVFGLVVGLLIGLVTEIYTSSDYKFVKKIADQSETGAA
TTVISGIAVGMQ

>NODE_264747_length_1361_cov_29.963263_S6
GICQGGHSSRQPYHRLLWHRTGGYMIRLLLKRRELSALFFLILLFLIAGIVNPAFLTLNN
VFLSINSSVVYAVVAMGIAFVIITGEIDVSVGAIVGISATVVGSMIRDGQPWLLALLAGI
GIGMLIGLINGFGVVTLRIPSIIMTLGTSSIIRGLMYVYTDGKWVENVPFEFKQLSQQKF
LDSFTYFYLAILLFMLLVHLIMMRSKRGKYYAAVGDNAAGANLLGIPVARTKLTAFVICG
VLSALGGVIFVSRVGFVTPIAGVGYEMKVIAACVIGGISLSGGVGNILGACIGAAFMASI
SRVLVFIGLSSDLDDTITGVLLIIIVVVDALLRKRSIEHARRERLSAKTLDLGGINNEAK
TV

>NODE_301074_length_916_cov_4.279476_S6
VVVGTMARSAELPLIIQIGATFNSIFGNFLGFCIPLIIIGFVVSGIAELGDGAGKTLGLT
VLIAYASTLFAGLLAYFVDVSVFPSFLKVGSIVLEDAQNAEETMLKGLFSIDMPPLMGVM
TALLLSFIFGIGIAVTHSTSLKNGFSEVQHIIEKLVAGVLIPLLPLHVYGIFANMTYAGT
VMDIMSVFIRVFAIIILLHVAVILIQYTIAGTVVGRNPIKLIRRMLPAYFTAIGTQSSAA
TIPVTVACTKSNDVSDRIAEFVCPLCATIHLSGSTITLTSCSIALMMLNGMDVTLGGLFP
FILMLGITMVAAPG

>NODE_313178_length_2508_cov_7.222488_S6
MLNKYGADATRWYLLHVSPAWSPTKFDEGGLQELASKFFGTLRNVYNFFVLYGNLDKIDV
KKLSVPYEKRSELDRWILSKYNKLIAEVTEHMDRYDHMKTVRAITDFVNEDLSNWYIRRA
RRRFYTPGMSADKESVFATTFEVLEGVARLIAPIAPFISDEMYSKLTGEETVHIAYYPKT
NAALIDEKVEKRMDIVRSVCNLGRGIREKKGLKVRQPLSEILVDGKYKDLISDMIPLIMD
ELNVKQVVFADELGEYMNFELKPNFKVAGPALGKKINTFAGVLAKEDAEKFTEKLEKDGF
VTCKMDGEDFKIEKEFVDIGINAKQGFAVAMENNVFVIIDTNLSQELIDEGIAREVISKI
QQMRKQNDYDMMDNINVYISADAEVLGAVSKHEAYIKSETLAKTLEEAANLPEVDINGHK
TGLQVERVQN

>NODE_338494_length_1128_cov_14.833333_S6
HGRLRDEHLQRGPRLQDDPGRQPAHQRPAAPGADQPLPGPGVLRGHRRADDPARPGRVLR
GRLRLRGLPLARQGHRHEHPPARDDDPLRRHDDPAVPALREGRARQLPVGRHPADDLHAL
PHPAVPAGLALLPARDHRGGPSRRSERDRHLRAYVRAYNEVDLRGGRRRHFHERVEQLHV
AQDHPRRRQVPDDADARVQPRGRVRHRLRRPHARRPHRVAARDGGLPRPAALLRQRNHGI
SQVNTELSHLTDPTCFADNRLPAHSDHLWYATEAEVASGRSSFQVCLDGVWKLHYATNPS
QAVEGFEVPSYDVSEWDDIAVPAHLQLHGYDKPQYANIQYPWDGHEQLEPGQVPSRYNPT
ASYVRAFTLPQVLPEGERLVLRLE

>NODE_377851_length_1918_cov_6.185089_S6
LRALARLDEAHRAARTHLHPLETGRKDRIMTMLSRRAFLSTCSGLGAAALAGCAPASGTD
DDATPDGGADGPSGLTKVSFVLDYSPNVNHTGIYVAIDQGFFAKEGIEVEIVPVPADGSD
ALIGAGGADMGLTYQDYIANSLSSANPLPYTAVAAVVQHNTSGIMSRAEDGIVRPKDMEG
HSYATWGLPIEQATVKQVVEEDGGDFSKVALVPYEVDDEVMGLQAGLFDTVWVYEWWAVQ
NAKLQEYPVNYFAFADISPQFDFYTPVIAANDAFAAADPELVRAFLRACEQGYELAATSP
ERAAEILCGAVPELDPALIAAAQASISPQYTADASRWGVIDRSRWTRFYEWLNDTGLVEN
GFDPALGFTNEYLEG

>NODE_414935_length_1586_cov_4.661412_S6
GKKNDMGMTMTQKILAAHAGLPQVKAGQLIEAKLDMVLANDITGPVSIGEFYRSGFENVF
DRKKIALVMDHFVPNKDIKSAEQCKKCRTFAKRLDIENYYDVGEMGIEHALLPEKGLVAS
GEAIIGADSHTCTYGALGAFSTGVGSTDVTAAIATGKTWFKVPQAVRFVLRGALKPYVCG
KDVILHIIGMIGVDGALYKSMEFTGDGVRSLTIDDRLTIANMAIEAGAKNGIFPVDSVTE
EYMAGRVTRPYKVCEADEDAEYEKTYNIDLSSIEPTVSFPHLPENTKAISECPDIEIDQV
IIGSCTNGRMQDMKQAADILRGKHMAKGVRGIVIPATMTVYKECIRLGYINDFIDAGCIV
STPTCGPCLGGYMGILADGERCVSTTNRNFVGRMGASGSEVYLAGPAVAAASGIAGKIAD
PRKTL

>NODE_458259_length_940_cov_5.839362_S6
RLYELTNKIAKPAVSFGGKYRIIDFPLSNCANSNINIVGVLTQYESVFLNSYVTADARWG
LDASDSGIFVLPPREKAGEDLNVYRGTADAISQNIDFVDQYEPDFVLILSGDHIYKMNYE
KMLEEHKASYADASIAVIEVPMKEASRFGIMNADATGRILEFEEKPEKPKSNLASMGIYI
FNWKVLRRMLVSDQKNDLSSHDFGKDIIPKMLDENKILHAYKFSGYWKDVGTVDSFWEAN
MDLLDPHNELSMFDPTWKIYTEDSYTLPQYIGKEAKISSAFITQGCVVEGRIERSVLFTG
VRVAKGAKIVDSVLMPGVEIGE

>NODE_515146_length_1002_cov_3.901198_S6
IFMKKHLVIVESPSKSKTIEKYLGNEYRVVSSKGHICDLATRGKERLGIDVDNNFEATYS
ISKEKKEVVKELQAFVKKSKDVYLASDPDREGEAIAWHLARVLDLDIENTNRIVFHEITK
PAVLEALKHPTHIDMDLVRSQETRRFLDRIIGFKLSRLLQNKIHSKSAGRVQSVALRLIV
ERENEIKAFQPQEYWTIHADVTKGKKKFEAVLSKVDGKKPKLNNEEDSHVILERCKEGDF
IVGKRTKRAKKKQARIPFTTSTLQQEASTKLNFGARRTMSIAQKLYEGIDLGGQQEGLIS
YMRTDSTRLSPMFVDDTLKYIEQTYGKEYKGTIRQKNSANAQD

>NODE_1060560_length_4372_cov_6.979186_S6
PVMERIIQDIVSAVRSAHRPPDEAWLAKLIRRYNKDVRDVARHTKKQQILAFYRKAREER
GQLWESWGIGAEEDRQILRLLKVKPRRTASGVATITVLTMPHPCSSACLYCPNDIRMPKS
YLANEPACQRAERNFFDPYLQVRARLALLESNGHITDKIELIVLGGTWSDYDPSYQIWFI
SELFRALNDGDGEAERICAERAAFYRSCGLIAEADTLAEQTRDLQRCVTAGALSYNQAIA
RLYASEAWVRARARQTATFGELEEQQRINESAHHRTVGLCVETRPDLVDDASAQLMRHLG
CTKVQMGIQSLDQDILDACGRHIRVEQIARAFSVLRLHGFKILAHMMVNLVGSTPEHDRL
DYGRLVGDPRFLPDEIKLYPCVLVESAALARLYDQGIWRPYTEDELLDVLAADVAATPAY
VRISRMIRDISSGDIVAGNKKTNLRQMVDARTEAAESAIAEIRSREIATGDVSACDVRLD
CISYTTAVSEERFLQWITDAGSIAGFLRLSLPHGRSTAMIREVHIYGRVAELGSIEAGGA
QHLGLGSALVETACKQASAAGCSAINVISSVGTRAYYRKLGFIDDGLYQRRVLGT

>NODE_1102966_length_2142_cov_5.032213_S6
WLRAVPAVSRCEYLTPLLRAVCVRCQFVTLPPLASKADRKRDASRYSRERACELPACFLG
WNKQPQLLFIYSTRDCRSRARPYFLHAGECAGRPCGSMRNRGHMAISVGIVGAAGFAGIE
LVRLVLRHSPFDLMAVTSTELSGRRLDEAYPAFAGQCDLAFSPHDADDLQSCDVVFLAVP
HTAALTFAPALIARGATVIDLSADFRLKDPAIYEEWYRVPHTEPELLARAAFGLPELFGE
ELAALAQRRSAGEGVLVACAGCYPTATSLAAAPVLRAGLSPAGLVVVDAVSGVTGAGRKA
TERTHFCFANEGVEAYGVGAHRHTPEIEQILGLEGRLIFTPHLAPYNRGLLSTVTMPVTR
GAFDQAELAEMYRSFFKDAPFVTVLPEGRQPRTVSVAGTNYAHVSACYNERAGAVVATCA
IDNIGKGAAGQAVQCANIVCGLPETCGLDAVALPI

SARS-CoV-2 related proteins:

>pdb|6YLA|A Chain A, SARS-CoV-2 RBD
ETGPNITNLCPFGEVFNATRFASVYAWNRKRISNCVADYSVLYNSASFSTFKCYGVSPTKLNDLCFTNVY
ADSFVIRGDEVRQIAPGQTGKIADYNYKLPDDFTGCVIAWNSNNLDSKVGGNYNYLYRLFRKSNLKPFER
DISTEIYQAGSTPCNGVEGFNCYFPLQSYGFQPTNGVGYQPYRVVVLSFELLHAPATVCGPKKSTNKHHH
HHH

Or go to this URL:

https://tinyurl.com/2f8h5vwf

(https://www.ncbi.nlm.nih.gov/protein/?term=Severe+acute+respiratory+syndrome+coronavirus+2%5Borganism%5D+AND+protein_structure_direct%5BFilt%5D)

click any SARS-CoV-2 related protein name and then click “FASTA” button at the top of the new page and use them to repeat exercise 1-3