

Protein Bioinformatics

Milot Mirdita Eli Levy Karin Wanwan Ge Franco Simonetti
Ruoshi Zhang Johannes Söding

November 5 – 6, 2019

Contents

1	Introduction to Linux and Bash	2
2	Metagenomic pathogen detection	5
3	Discovering candidate Cas14 orthologs	10
4	Protein structure prediction	14
5	Appendix	22

Introduction to Linux and Bash

1.1 Linux

Throughout this tutorial you will work in a **Linux** environment. Briefly, Linux is a descendant of the UNIX operating systems family. It is popular because it is open-source, free and runs on everything from tiny micro controllers, to phones, computer clusters and even super computers. It has found wide adoption in the bioinformatics community. An operating system has many important roles, which include:

- managing a file system: information (generally: “files”) is stored on the computer hard disk. The operating system manages the access to files. To do so, it represents their location as a tree hierarchy. Each file has a **path**, starting from the root and going through **directories**. For example:

```
/home/coder/seriously_important.txt
```

- managing resources: all software running on the computer cannot access its resources directly but rather, they get services from the operating system, which makes sure the resources are allocated fairly and safely. The same is true for us, **users** of the computer.

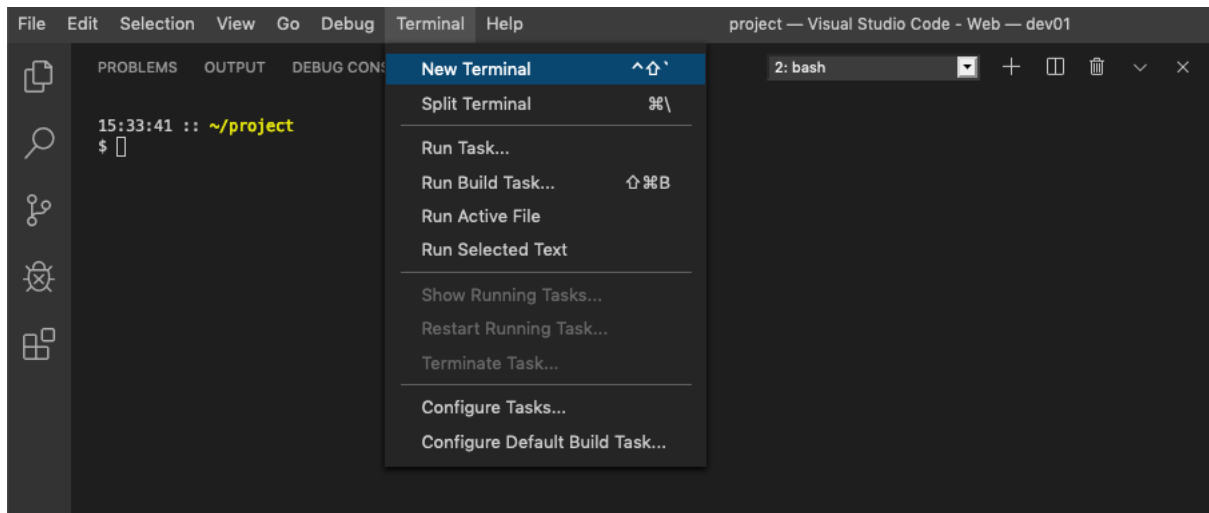
If we want to save a new file to the disk, we do it through the operating system. We usually do it using a graphical interface (press some button and save). Today we will communicate with the Linux operating system using a **textual interface**.

1.2 Bash

A “**Shell**” is a basic textual interface to communicate with the operating system. We do so by typing commands in a designated command window. These commands allow us for example, to create a new file or to navigate to some directory. Below you will get familiar with a few basic textual commands in a specific type of Linux Shell, called “**Bash**”.

You will work remotely on one of our servers, where we have prepared an integrated development environment¹ for you that contains a text editor and a shell. We will assign a number NN to each of you. Replace NN with your number in this URL <https://devNN.mmseqs.com> and open it in Firefox. You will see something like the following image. You can open new terminal at “Terminal -> New Terminal”

¹<https://github.com/cdr/code-server>



Now, in the Bash window, let's type the following commands:

```
# print working directory: the full path from the root of the current directory  
pwd
```

This should result in a sub-folder of your **home directory**:

```
/home/coder/project
```

```
# change directory: navigate to the data directory under your home directory  
cd data
```

Validate that your location (directory) has indeed changed.

```
# list files and sub-directories in the directory:  
ls
```

You should see:

- useful_links.txt

Bash Tip 1 To avoid typos and save time, if you partially type a command or a file name, you can press the `[TAB]` key to get the automatic completion of your command or file. If what you are typing cannot be uniquely completed, you can press the `[TAB]` key twice to see a list of suggestions.

In this tutorial, whenever you see YourSomething it means you need to replace it with a value you choose.

```
# create a copy of a file:  
cp useful_links.txt YourFileNameCopy
```

```
# print the first 5 lines of a file:  
head -n 5 useful_links.txt
```

```
# print the entire content of a file to the screen:  
cat useful_links.txt
```

Validate that `useful_links.txt` and `YourFileNameCopy` have the same content

```
# print the number of lines in a file:
wc -l useful_links.txt

# remove a file (permanently deletes it! Achtung!!!):
rm YourFileNameCopy
```

Now, let's play with directories.

In the commands below, instead of `YourDirName`, you can type any name you choose.

```
# make directory: create a directory in the current location.
mkdir YourDirName
```

Change directory to `YourDirName` and validate that you are indeed in the right location

```
# go back to the parent directory:
cd ..

# remove a directory (-r for "recursive"; permanently deletes it! Achtung!!!):
rm -r YourDirName
```

Some more commands are described in the appendix 5.1.

There are many more features to Bash. Check out this resource to learn more: ryanstutorials.net/linuxtutorial

Later today, we will use Bash to run metagenomics software.

Bash Tip 2: To cancel a running program you can press `Ctrl` + `C`.

1.3 File formats

Biological information is conventionally stored in specific textual formats. These formats indicate where, for example the name of the gene/protein is stored and where the sequence itself is stored. This way bioinformatic tools can extract the needed information from the files. One of the most common formats is called **FASTA**. It is used, for example, to store metagenomics sequence reads. In FASTA format, an identifier (a protein ID, for example) is written after the “>” symbol, and its corresponding sequence is written in the following lines.

The **tsv** (tab separated values) formatted files, with which you are going to work later, contain one record per line, with attributes about this record separated by “TAB” characters. This is a common representation of data in bioinformatics and easy to explore with standard Linux tools.

Metagenomic pathogen detection

2.1 The Patient

A 61-year-old man was admitted in December 2016 with bilateral headache, gait instability, lethargy, and confusion. Because of multiple tick bites in the preceding 2 weeks, he was prescribed the antibiotic doxycycline for presumed Lyme disease. Over the next 48 hours, he developed worsening confusion, weakness, and ataxia. He returned to the referring hospital and was admitted. He lived in a heavily wooded area in New Hampshire, had frequent tick exposures, and worked as a construction contractor in basements with uncertain rodent and bat exposures. His symptoms were diagnosed as Encephalitis and the causative agent — not known.

? Your task will be to identify the pathogenic root cause of the disease.

This pathogen is usually confirmed by a screening antibody test, followed by a plaque reduction neutralization test. However, this takes 5 weeks, which was too slow to affect the patient's care. As traditional tests done in the first week of the patient's hospital stay did not reveal any conclusive disease cause, the doctors were running out of options. Therefore a novel metagenomic analysis was performed.

2.1.1 The Dataset

Metagenomic sequencing from cerebrospinal fluid was performed on hospital day 8. It returned 14 million short nucleotide sequences (reads).

The authors of the study removed all human reads using Kraken [1] and released a much smaller set of 226,908 reads on the SRA (<https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi>). Kraken extracts short nucleotide subsequences of length k , also called k -mers, and compares them to a reference database where k -mers point to taxonomic labels. In case of exact matching it is able to assign taxonomy.

? Why didn't the authors release the complete dataset?

? Demanding exact k -mer matching in Kraken has benefits for removing human reads. Why?

? What is the SRA? How many samples are there in the SRA currently? How many bases are publicly available on the SRA in total?

2.2 Metagenomic pathogen detection using MMseqs2

We will use the sequence search tool MMseqs2 [2] to find the cause of this patient's disease. MMseqs2 translates the nucleotide reads to putative protein fragments, searches against a protein reference database and assigns taxonomic labels based on the found reference database hits.

? Why might a protein-protein search be useful for finding bacterial or viral pathogens? How does this compare with Kraken's approach?

2.2.1 Assigning taxonomic labels

We already placed a FASTA file at `pathogens/reads.fasta` containing the reads.¹ First, change to the exercise directory: `cd pathogens`. Here you will see the previously mentioned `reads.fasta` file and a couple files starting with `uniprot_sprot`. This contains all the reference proteins from Swiss-Prot which is the manually curated, high-quality part of the Uniprot[4] protein reference database. We are using this smaller subset of about 500,000 proteins, since the full Uniprot with over 175,000,000 sequences requires too many computational resources. Each protein in Uniprot has a taxonomic label. Through a similarity search we will transfer the annotation of the reference protein to our unknown reads.

```
mmseqs createdb reads.fasta reads
mmseqs taxonomy reads uniprot_sprot lca_result tmp -s 2
```

MMseqs2 will create a result database in your current working directory. This database consists of files, whose names start with `lca_result`. We can convert this database into a human readable tab separated values file (TSV), a common format in bioinformatics:

```
mmseqs createtsv reads lca_result lca.tsv
```

In this file you see for every read a numeric taxonomic identifier, a taxonomic rank and a taxonomic label. However, due to the large number of reads, it is hard to gain insight by skimming the file. MMseqs2 offers a module to summarize the data into a single file `report.txt`:

```
mmseqs taxonomyreport uniprot_sprot lca_result report.txt
```

? What is the most common species in this dataset?

? Why are there so many different eukaryotic sequences? Were they really in the spinal fluid sample?

¹The sequencing machine returns paired-end reads where sequencing starts in opposite directions from two close-by points to cover the same genomic region. Some of these paired reads overlap enough to be merged into a single read with FLASH [3].

2.2.2 Visualizing taxonomic results

MMseqs2 can also generate an interactive visualization of the data using Krona [5]. Adapt the previous call to generate a Krona report:

```
mmseqs taxonomyreport uniprot_sprot lca_result report.html --report-mode 1
```

This generates a HTML file that can be opened in a browser. This offers an interactive circular visualization where you can click on each label to zoom into different parts of the hierarchy. Since your editor only display the content of the HTML file and not render it. You have to first navigate to it. Open the URL <https://devNN.mmseqs.com/web> in a new tab. There you will see your `report.html` file.

2.2.3 What is the pathogen?

Look up the following encephalitis causing agents in Wikipedia.

1. Borrelia bacterium
2. Herpes simplex virus
3. Powassan virus
4. West Nile virus
5. Mycoplasma
6. Angiostrongylus cantonensis

- ? Based on the literature, which one is the most likely pathogen?
- ? For which species do you find evidence in the metagenomic reads?
- ? Approximately how many reads belong to the pathogen? Based on this number, how would you determine if it is significant evidence for an actual presence of this agent?

2.3 Investigating the pathogen

We now want to take a closer look only at the reads of the pathogen. To filter the result database, we will need the pathogen's numeric taxonomic identifier. Use the NCBI Taxonomy Browser to find it, by searching for its name:

<https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi>.

- ? What is the taxon identifier of the pathogen? Did you find one or more?

Now we can call a different MMseqs2 module to retrieve only the reads that belong to this pathogen. Replace **XXX** with the number(s) you just found. If you found multiple, concatenate them with a comma `,` character.


```
mmseqs filtertaxdb uniprot_sprot lca_result lca_only_pathogen --taxon-list XXX
```

We now get a list of all queries that were **filtered out**, meaning they were annotated as pathogenic.

```
grep -Pv '\t1$' lca_only_pathogen.index > pathogenic_read_ids
```

With a few more commands we can convert our taxonomic labels back into a FASTA file:

```
mmseqs createsubdb pathogenic_read_ids reads reads_pathogen
```

```
mmseqs convert2fasta reads_pathogen reads_pathogen.fasta
```

? **How many reads of the pathogen are in this resulting FASTA file?**

2.4 Assembling reads to proteins

We want to try to recover the protein sequences of the pathogen.

? **Which proteins do you expect to find in the pathogen you discovered?**
Search the internet.

We will use the protein assembly method Plasm [6] to find overlapping reads and generate whole proteins out of the best matching ones.

```
plasm assemble reads_pathogen.fasta pathogen_assembly.fasta tmp
```

Take a look at the generated FASTA file `pathogen_assembly.fasta`.

? **How many sequences were assembled?**

? **Do some of the sequences look similar to each other?**

2.5 Clustering to find representative proteins

Plasm will uncover a lot of variation in the reads and output many similar proteins. We can use the sequence clustering module in MMseqs2 to get only representative sequences.

```
mmseqs easy-cluster pathogen_assembly.fasta assembly_clustered tmp
```

You will see three files starting with `assembly_clustered`:

1. `assembly_clustered_all_seqs.fasta`
2. `assembly_clustered_cluster.tsv`
3. `assembly_clustered_rep_seq.fasta`

Take a look at the last one `assembly_clustered_rep_seq.fasta`. This file contains all representative sequences, meaning the sequence that the algorithm chose as the most representative within this cluster.

? **How many sequences remain now? How well does this agree with what you expected according to your internet search?**

2.6 Annotating the proteins

We will look for known protein domains to identify the proteins we found. Instead of the MMseqs2 command line, we use the MMseqs2 webserver, which will visualize the results. Paste the content of the FASTA file containing the representative sequences into the webserver and make sure to select all three target databases (PFAM, PDB, Uniclust): <https://search.mmseqs.com>

? Which of the expected proteins do you find?

2.7 Aftermath

Despite being able to identify the causative agent. The pathogen is very hard to treat. The patient had minimal neurological recovery and was discharged to an acute care facility on hospital day 30. Seven months after discharge, he was reportedly able to nod his head to questions and slightly move his upper extremities and toes.

You can find the publication about this patient and dataset here [7]. Please look at it only after trying to answer the questions yourself.

Discovering candidate Cas14 orthologs

3.1 Introduction

CRISPR-Cas9 systems provide bacteria and archaea with adaptive immunity to infectious nucleic acids (e.g., viruses). Recently, Harrington and Burstein et al. [8] discovered CRISPR-Cas systems in archaea that consist of a previously unreported Cas14 proteins. These proteins are compact RNA-guided nucleases (400 to 700 amino acids in length). In their work, the authors identified a set of 45 Cas14 proteins by constructing and iteratively refining hidden Markov models (HMMs) and using them to query public metagenomes from IMG/M.

3.2 Goal and motivation

We will examine candidate orthologs of Cas14 in order to enrich the authors' original set. This is very useful for improving HMMs, identifying taxa that have this system as well as to better understand the functionality of the protein. This in turn could improve any biotechnological use of CRISPR-Cas engineering.

In the interest of time, we carried some of the computational steps for you. Your tasks are in **red**.

3.3 Where to look?

Our chances of finding highly diverse orthologs increase as we explore more comprehensive protein databases. We thus chose to use “**BFD**” (**B**ig well... let's just say... **F**antastic Database; <https://bfd.mmseqs.com/>), which was constructed from 2,500,000,000 protein sequences of various sources, including environmental samples of soil and ocean. For similar tasks in the future, keep an eye open for comprehensive databases. The BFD has been clustered using **Linclust** [9] to 65,983,866 clusters of 30% sequence identity. A multiple sequence alignment (MSA) was computed from each of the BFD clusters.

3.4 Input

Our starting point will be the previously reported 45 sequences.¹

Change to the exercise directory: Cas14

Download the sequences by using the command:

```
wget <url_to_sequences_see_above>
```

Using Bash commands

? What is the average Cas14 length (in amino acids)?

A) 45 B) 563.2 C) 553.5 D) 626.4

Solution: (25344 - 438) / 45

```
grep -v ">" aav4294_Data_S2.fasta | wc -c
# the number of characters (including \n) in sequence lines is: 25344

grep -cv ">" aav4294_Data_S2.fasta
# the number of sequence lines is: 438

grep -c ">" aav4294_Data_S2.fasta
# the number of sequences is: 45
```

commands.

There are several possible solutions. Here is one that doesn't require more than basic

3.5 How to search?

Like Harrington and Burstein et al., it is useful to use HMMs. One approach is to align the previously reported 45 sequences using **MAFFT** [10]. Then, run **HHblits** [11] with this alignment as input and the BFD_MSAs as database. This conducts an iterative HMM to HMM search and results in a set of alignments computed from the constructed profiles. We performed this step for you because the BFD database is too large for you to have in your exercise environment. Following such a search, we performed several steps and obtained the candidate sequences from the output and saved them in **cas14_bfd_candidates.fasta**.

3.6 Aligning known Cas14 and bfd candidates

The **cas14_bfd_candidates.fasta** file contains three types of sequences: previously reported Cas14 ("CAS" headers), sequences that were found in standard reference databases, such as UniProt ("REF" headers) and sequences that were found strictly in environmental metagenomic samples ("ENV" headers).

¹ https://science.sciencemag.org/highwire/filestream/716984/field_highwire_adjunct_files/1/aav4294_Data_S2.fasta

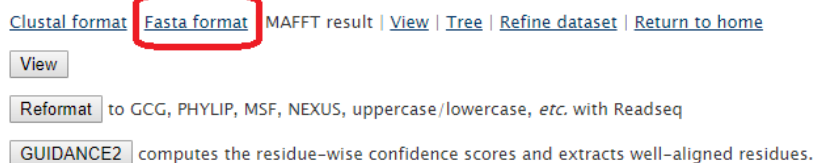
Using Bash commands, inspect `cas14_bfd_candidates.fasta` file.

? How many sequences are there of each type?

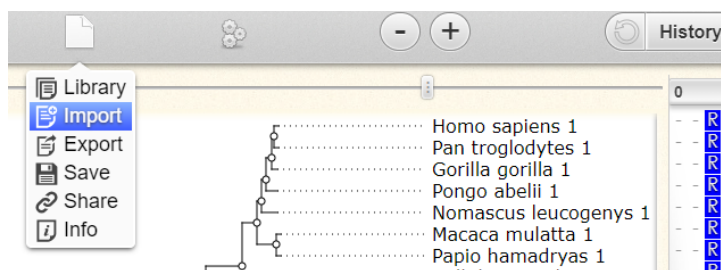
```
grep ">ENV" cas14_bfd_candidates.fasta | wc -l 791#
grep ">REF" cas14_bfd_candidates.fasta | wc -l 25#
grep ">CAS" cas14_bfd_candidates.fasta | wc -l 45#
```

Solution:

Align all these sequences using [the MAFFT online server](#) and save the result as `cas14_bfd_candidates_MSA.fasta`.²



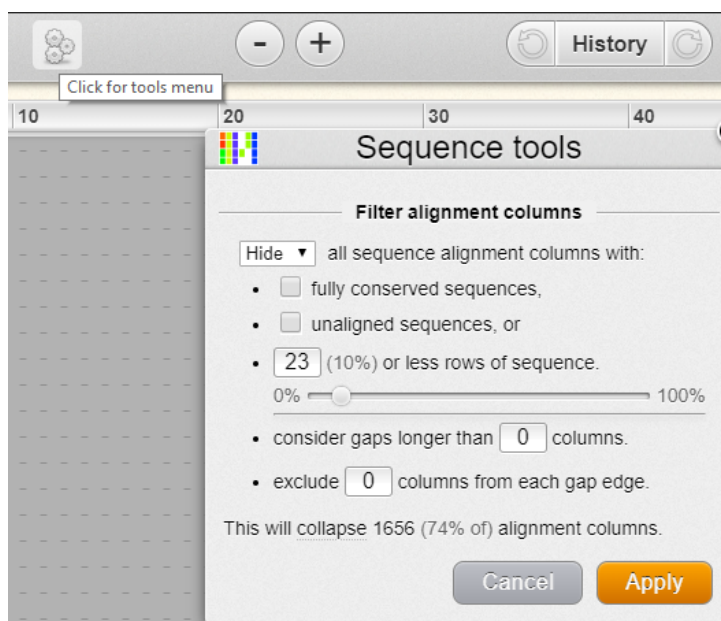
Upload the MSA file to [the Wasabi MSA viewer](#).³



Scroll and zoom in and out to get an overall impression.

? What can you say about the MSA? For example, what is its length?

Use the “collapse gaps” option:



² <https://mafft.cbrc.jp/alignment/server/>

³ <http://wasabiapp.org/>

? What would be interesting things to consider?

Some of the ENV sequences resemble the CAS proteins, suggestive of true homologs. which reassures us that it is not a single domain that matches. Also, we can see at least one of the ENV sequences resemble the CAS proteins, suggestive of true homologs. We can see that these columns correspond to the full length of the original MSA, which reassures us that it is not a single domain that matches. Also, we can see at least one of the ENV sequences resemble the CAS proteins, suggestive of true homologs. There are 2,217 columns in the MSA. The MSA has quite a lot of gap characters so it is hard to examine. After collapsing the gaps with default settings 26% of the columns remain. We can see that these columns correspond to the full length of the original MSA, which reassures us that it is not a single domain that matches. Also, we can see at least one of the ENV sequences resemble the CAS proteins, suggestive of true homologs. **Solution:**

3.7 Computing a phylogenetic tree

A phylogenetic tree represents the reconstructed evolution leading to the sequences in a multiple sequence alignment (MSA). There are several ways⁴ to infer phylogenetic trees based on MSAs. The likelihood criterion allows scoring each possible tree based on its probability to give rise to the sequences by using a statistical model of sequence evolution. This criterion is often used together with a search procedure to scan and score possible trees until the highest score is reached. Various software tools⁵ implement this tree reconstruction strategy. Today, we will use FastTree⁶[12], which approximates the maximum likelihood computation to achieve short running times.

Reconstruct a phylogenetic tree using FastTree:

```
FastTree cas14_bfd_candidates_MSA.fasta > cas14_bfd_candidates_MSA.nwc
```

3.8 Viewing the tree

By examining the tree we can learn of the divergence of the bfd candidates. We will use the interactive Tree Of Life (iTOL, Letunic and Bork 2019 Nucleic Acids Res) server to examine the tree. The server allows various tree displays, coloring branches and leaves, adding labels and exporting the tree to common formats, such as PDF.

Upload `cas14_bfd_candidates_MSA.nwc` to the [iTOL server](#)⁷.

We have prepared an annotation file (`cas14_bfd_candidates iTOL_leaves.txt`) based on the iTOL format of coloring leaf labels. **Drag and drop this file on your tree.**

Questions:

- ? What can you say about the diversity of the sequences with respect to their source?
- ? Do you think all candidate sequences are Cas14 orthologs? Please explain.

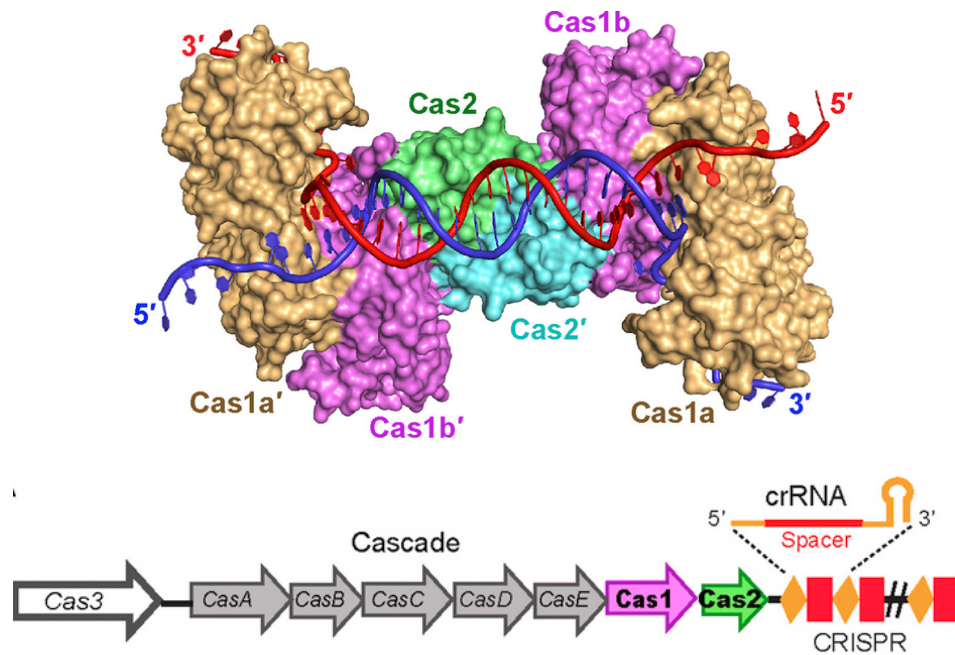
⁴ https://en.wikipedia.org/wiki/Phylogenetic_tree#Construction

⁵ <https://molbiol-tools.ca/Phylogeny.html>

⁶ <http://www.microbesonline.org/fasttree/>

⁷ <https://itol.embl.de/>

Protein structure prediction



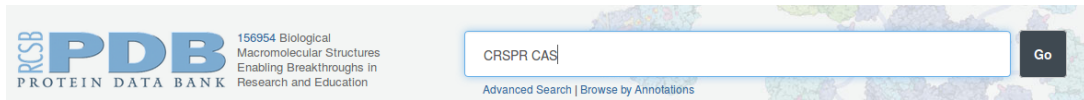
In this section you will learn how to:

1. Search for protein structures on the RCSB PDB[13] and UniProt websites[4];
2. Use visualization tools to explore protein structures and the interface of proteins and DNA;
3. Use homology modeling to predict protein structures and evaluate the predicted models [14].

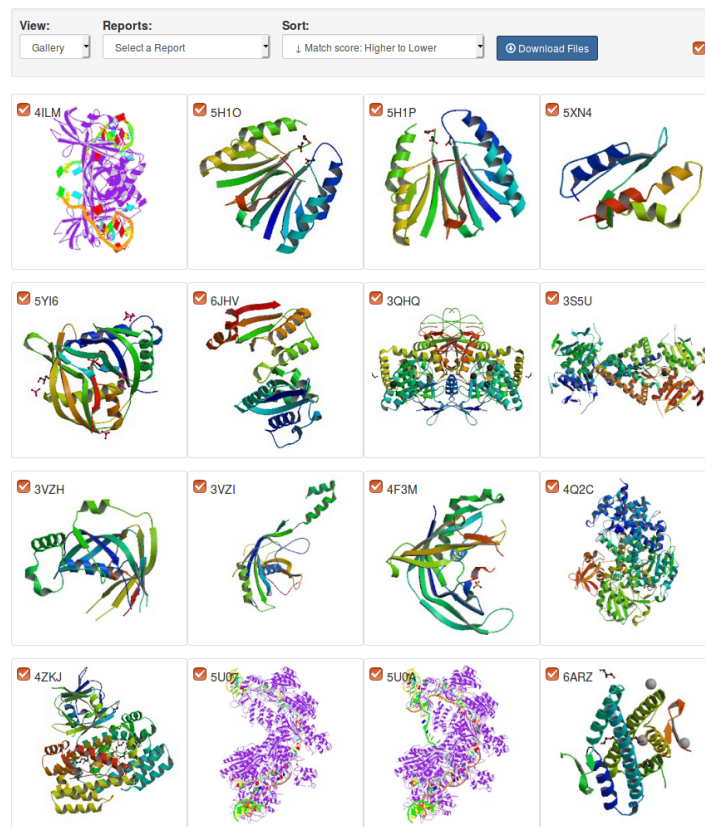
Have fun!

4.1 Find the solved CAS protein structures in the RCSB PDB (Protein Data Bank)

1. Go to the RCSB PDB website: <http://www.rcsb.org>
2. Search with a keyword such as “CRISPR”, click the “Go” button:



3. Explore the result page in the “Gallery” view, and you will see a collection of the CRISPR-related protein structures:



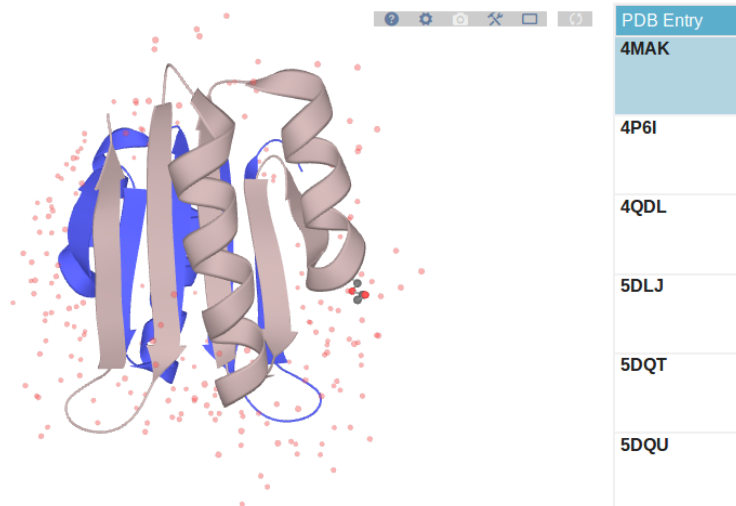
4.2 Basic protein visualization in the UniProt database

1. Go to the UniProt website: <https://www.uniprot.org/>.
2. Search “crispr cas2”:



3. Click on the Entry “**P45956**” for the Cas2 protein from *Escherichia coli* and go to the summary page of this protein.
4. Scroll down the page and find the structure of the Cas2 protein, which shows a homo-dimer from PDB ID **4MAK**. Chain A is shown in pink and chain B is shown in blue

Structureⁱ



5. Click and drag the mouse on the structure and you will see the structure rotates.
6. Point the mouse on the structure, you will see an amino acid will be highlighted in yellow color and the position and the name of the amino acid will be shown on the top left corner.

? What is the amino acid of the first alpha-helix on chain A

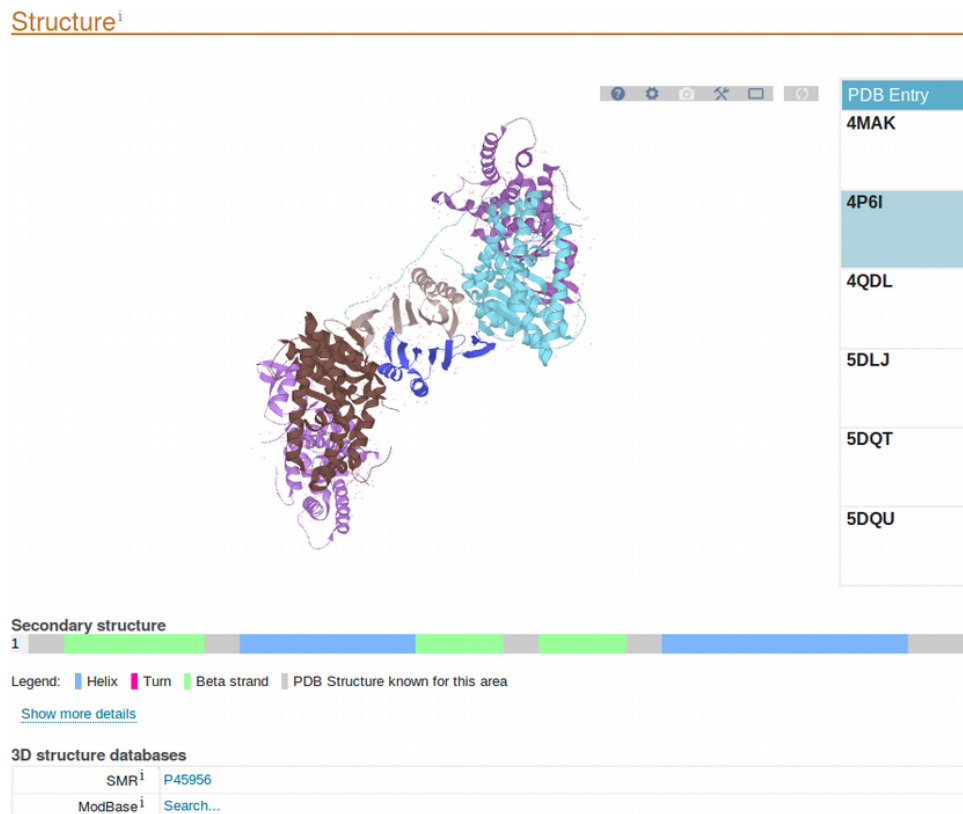
(Answer: Pro)

If you click on it, you can see the structure of that single amino acid.

4.3 Look at Cas1-Cas2 complex from different angles in SWISS-MODEL [15]

1. In the Structure section on the UniProt page <https://www.uniprot.org/uniprot/P45956>, choose the PDB entry **4P6I**, you will see a heteromer structure of Cas1-Cas2 complex:

Structureⁱ



Secondary structure

1

Legend: Helix Turn Beta strand PDB Structure known for this area

[Show more details](#)

3D structure databases

SMR ⁱ	P45956
ModBase ⁱ	Search...

2. Click on the SMR entry P45956 under “**3D structure databases**” and this leads you to the Swiss Model Repository.
3. On the left side of the page, it shows the protein sequence while on the right side, it shows the structure of the Cas1-Cas2 complex.

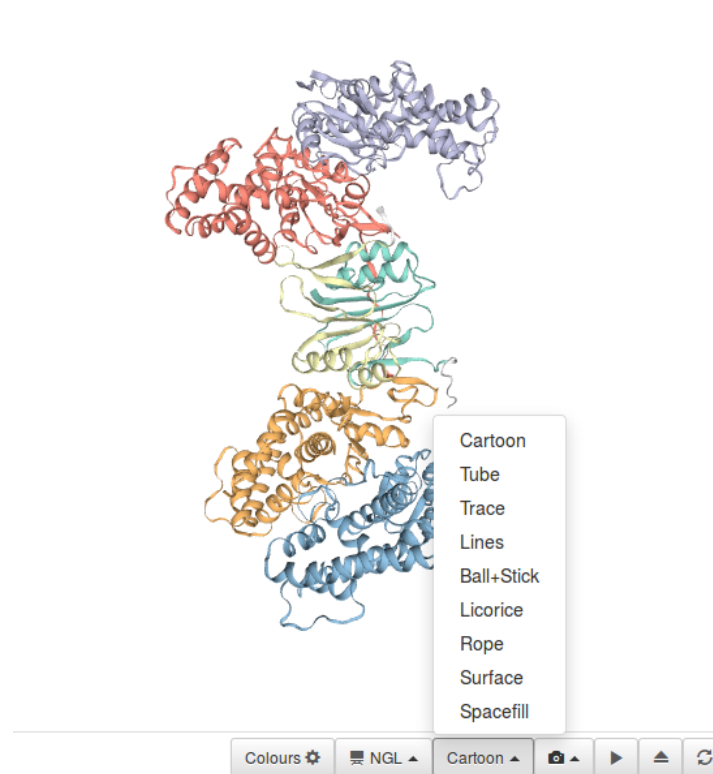
Tips for playing with the model:

- (a) You can rotate the structure by clicking and dragging the left button of the mouse over the structure;
- (b) You can zoom in and out of the structure by scrolling the middle button of the mouse;
- (c) You can move the structure by clicking and dragging the right button of the mouse over the structure.

4. To find out where Cas2 is located in the complex, click on the “**InterPro**” button in the middle of the page.

5. Change colors: (Before doing this, don't forget to unclick the "InterPro" button)

- (a) By default, the structure is shown in rainbow color scheme. Change it now to "**Chain**". Now you see one of the two Cas2 chains is highlighted in green color, which is corresponding to the sequence on the left, and the other chain is shown in yellow. The other 4 chains are from Cas1 protein.



- (b) Change the display mode from Cartoon to Surface, and change the Colours to Charged. Now you will see the positive charged amino acids are shown in red while the negative charged ones are in blue.

6. Find out the interface of Cas1-Cas2 to DNA strands:

- (a) On the bottom of the page, choose a different structure: the Cas-DNA-PAM complex with a PDB ID **5dqz** [16]. This structure illustrates the interface between the Cas1-Cas2 and the DNA strands.
- (b) Set the Colours to **Chains** and change the display mode to **Spacefill**.
- (c) Check the interactions between amino acids and ions. Notice there are 2 Magnesium ions in the structure. Change the display mode to **Licorice**. Click on:

2 x MAGNESIUM ION ☒

It will highlight the regions where the Mg^{2+} ions bind to. Click on the little box and you will see the details of interactions.

? Which amino acids do the Mg^{2+} ions bind to?

Click on **MG.1**, and you will see which amino acids it binds to. (Answer: E, 2xH, 2xD)

4.4 Homology modeling using SWISS-MODEL

Homology modeling is one of the most useful computational methods for protein structure prediction. Given the evolutionary conservation of protein structures, we can use the known protein structure to predict the structures of proteins which come from a common ancestor.

1. Go to modeling page: <https://swissmodel.expasy.org/interactive>

Target sequence:

```
>Cas1
MVQLYVSDSVSRISFADGRVIVWSEELGESQYPIETLDGITLFGRPMTTPFI
VEMLKRERDIQLFTTDGHYQGRISTPDVSYAPRLRQQVHRTDDPAFCLSLS
KRIVSRKILNQQALIRAHTSGQDVAESIRTMKHS LAWVDRSGSLAELNGFEG
NAAKAYFTALGHLVPQEFQGRSTRPPLDAFNMSVSLGYSLLYKNIIGAIE
RHSLNAYIGFLHQDSRGHATLASDLMEVWRAPIIDDTVLRRLIADGVVDTRA
FSKNSDTGAVFATREATRSIARAFGNRIARTATYIKGDPHRYTFQYALDLQL
QSLVRVIEAGHPSRLVDIDITSEPSGA
```

2. Copy our target sequences above and paste it to the “Target Sequence(s)” window, give a name to this project, and click “Search For Templates” button:

Start a New Modelling Project

Target Sequence(s): (Format must be FASTA, Clustal, plain string, or a valid UniProtKB AC)

Target: MVQLYVSDSVSRISFADGRVIVWSEELGESQYPIETLDGITLFGRPMTTPFIVEMLKRERDIQLFTTDGHYQGRISTPDVSYAPRLRQQVHRTDDPAFCLSLSKRIVSRKILNQQALIRAHTSGQDVAESIRTMKHS LAWVDRS 145

Target: GSLAELNGFEGNAAKAYFTALGHLVPQEFQGRSTRPPLDAFNMSVSLGYSLLYKNIIGAIE RHSLNAYIGFLHQDSRGHATLASDLMEVWRAPIIDDTVLRRLIADGVVDTRAFSKNSDTGAVFATREATRSIARAFGNRIART 299

Target: ATYIKGDPHRYTFQYALDLQLQSLVRVIEAGHPSRLVDIDITSEPSGA 338

Add Hetero Target Reset

Project Title: Cas1

Email: Optional

Search For Templates Build Model

3. The process will run for a few minutes, and the intermediate page shows the real time progress:

Cas1 Created: today at 12:37

Summary Templates Models

Template Results

The search for templates matching your target sequence is currently running. Please wait.

Run:ing

If you want to come back later, bookmark this link:
<https://swissmodel.expasy.org/interactive/29360/>

```
MVQLYVSDSVSRISFADGRVIVWSEELGESQYPIETLDGITLFGRPMTTPFIVEMLKRERDIQLFTTDGHYQGRISTPDVSYAPRLRQQVHRTDDPAFCLSLSKRIVSRKILNQQALIRAHTSGQDVAESIRTMKHS LAWVDR
S
GSLAELNGFEGNAAKAYFTALGHLVPQEFQGRSTRPPLDAFNMSVSLGYSLLYKNIIGAIE RHSLNAYIGFLHQDSRGHATLASDLMEVWRAPIIDDTVLRRLIADGVVDTRAFSKNSDTGAVFATREATRSIARAFGNRIAR
T
ATYIKGDPHRYTFQYALDLQLQSLVRVIEAGHPSRLVDIDITSEPSGA
```

4. The results page shows different protein structure templates that can be used to predict the 3D structure for our target protein. These templates are ranked due to how well their sequences align with the target sequence.

? To which structure does it have the most sequence identity?

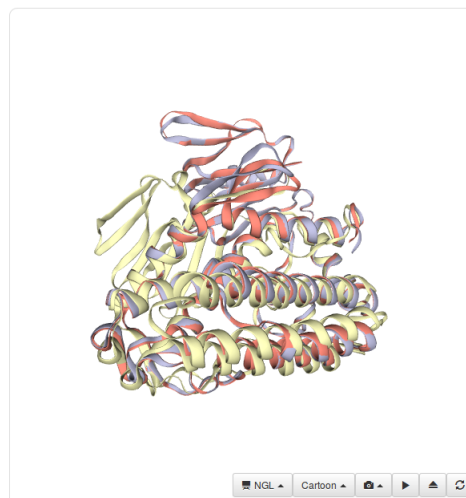
(Answer: 6qxf.1.1 and 6qxf.1.1 (34.43%))

Note that sequences falling below a 20% sequence identity can have very different structure.

Template Results

Template Results											
Templates Quaternary Structure Sequence Similarity Alignment of Selected Templates More											
Sort	Name	Title	Coverage	GMQE	OSQE	Identity	Method	Oligo State	Ligands		
<input checked="" type="checkbox"/>	4n06.1.B	CRISPR-associated endonuclease Cas1 1		0.65	0.43	24.53	X-ray, 2.4Å	homo-dimer ✓	None	▼	
<input type="checkbox"/>	4n06.1.A	CRISPR-associated endonuclease Cas1 1		0.65	0.43	24.53	X-ray, 2.4Å	homo-dimer ✓	None	▼	
<input type="checkbox"/>	4xtk.3.B	CRISPR-associated endonuclease Cas1		0.60	0.45	19.87	X-ray, 2.7Å	homo-dimer ✓	None	▼	
<input type="checkbox"/>	4xtk.1.B	CRISPR-associated endonuclease Cas1		0.61	0.44	19.87	X-ray, 2.7Å	homo-dimer ✓	None	▼	
<input type="checkbox"/>	4xtk.1.A	CRISPR-associated endonuclease Cas1		0.61	0.44	19.87	X-ray, 2.7Å	homo-dimer ✓	None	▼	
<input type="checkbox"/>	4xtk.2.A	CRISPR-associated endonuclease Cas1		0.60	0.43	19.87	X-ray, 2.7Å	homo-dimer ✓	None	▼	

Build Models 3
Clear Selection



- We select 3 templates with high identity scores and build models upon them. Click the “Build Models” button on the top right. Model results page is shown below. The predicted structures are ranked according to the quality of their models.

Summary **Templates** 60 **Models** 3

Model Results

Order by: GMQE

Model	Oligo-State	Ligands	GMQE	OMEAN
Model 01	Homo-dimer (matching prediction)	None	0.67	-1.36 kD
Model 02	Homo-tetramer (matching prediction)	None	0.63	-2.83 kD
Model 03	Homo-dimer (matching prediction)	None	0.48	-5.00 kD

Global Quality Estimate: OMEAN, Cp, All Atom, solvation, torsion

Local Quality Estimate: Residue Number

Comparison: Protein Size (Residues)

Template: 4n06.1.A, Seq Identity: 24.53%, Coverage:

Description: CRISPR-associated endonuclease Cas1 1

Model-Template Alignment

Template: 4xtk.2.A, Seq Identity: 28.57%, Coverage:

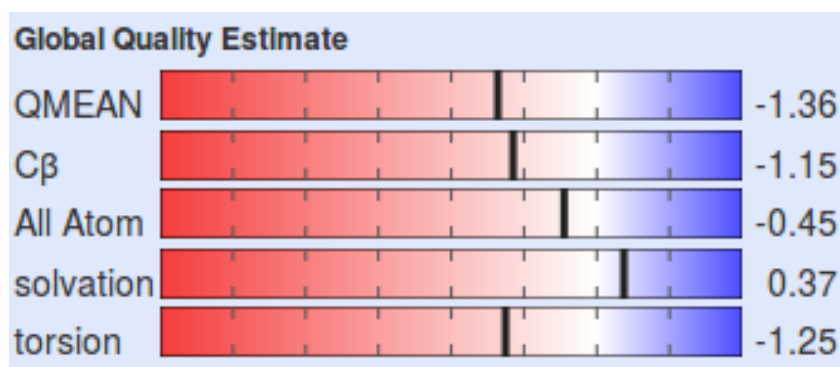
Description: CRISPR-associated endonuclease Cas1

Model-Template Alignment

1 338

6. Model evaluation

The first model (built using [4n06.1.A](#) as a template) has the most best GMQE score and all five QMEAN terms mostly fall near the white region. Therefore, it is the optimal model that we can get for the target sequence, given the known homologous protein structures.



GMQE(Global Model Quality Estimation) is a quality estimation which combines properties from the target–template alignment and the template search method. The resulting GMQE score is expressed as a number between 0 and 1, reflecting the expected accuracy of a model built with that alignment and template and the coverage of the target. Higher numbers indicate higher reliability.

QMEAN is a composite estimator based on different geometrical properties and provides both global (i.e. for the entire structure) and local (i.e. per residue) absolute quality estimates on the basis of one single model. Scores of -4.0 or below are an indication of models with low quality.

Appendix

5.1 Some useful Bash commands

```
# show a file inside the terminal  
less myFile
```

```
# show only the second column from a TSV file  
cut -f2 YourFile
```

```
# show the lexicographically sorted lines of a file  
sort YourFile
```

```
# show the numerically sorted lines of a file  
sort -n YourFile
```

```
# store in YourFileSorted, a sorted version of your file  
sort YourFile > YourFileSorted
```

```
# show only unique elements in a file (the file needs to be sorted first)  
uniq YourFileSorted
```

```
# show how often every unique element occurred in a file (file needs to be sorted)  
uniq -c YourFileSorted
```

```
# pipe example to count the number of files in the current directory:  
pwd | ls | wc -l
```

```
# another pipe example: sort lines lexicographically, count appearances of each line  
↪ and sort by the counts in reverse order  
sort YourFile | uniq -c | sort -n -r
```

5.2 Letter codes for amino acids in a protein chain

A	Alanine	Ala
C	Cysteine	Cys
D	Aspartic Acid	Asp
E	Glutamic Acid	Glu
F	Phenylalanine	Phe
G	Glycine	Gly
H	Histidine	His
I	Isoleucine	Ile
K	Lysine	Lys
L	Leucine	Leu
M	Methionine	Met
N	Asparagine	Asn
P	Proline	Pro
Q	Glutamine	Gln
R	Arginine	Arg
S	Serine	Ser
T	Threonine	Thr
V	Valine	Val
W	Tryptophan	Trp
Y	Tyrosine	Tyr

Bibliography

- [1] Derrick E Wood and Steven L Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.*, 15(3):R46, 2014.
- [2] Martin Steinegger and Johannes Soeding. Mmseqs2: sensitive protein sequence searching for the analysis of massive data sets. *bioRxiv*, page 079681, 2017.
- [3] Tanja Magoc and Steven L. Salzberg. Flash: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, 27(21):2957–2963, 2011.
- [4] UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2014.
- [5] Brian D Ondov, Nicholas H Bergman, and Adam M Phillippy. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics*, 12(1):385, 2011.
- [6] Martin Steinegger, Milot Mirdita, and Johannes Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manifold. *Nat. Methods*, 16(7):603–606, 2019.
- [7] Anne Piantadosi, Sanjat Kanjilal, Vijay Ganesh, Arjun Khanna, Emily P Hyle, Jonathan Rosand, Tyler Bold, Hayden C Metsky, Jacob Lemieux, Michael J Leone, Lisa Freimark, Christian B Matranga, Gordon Adams, Graham McGrath, Siavash Zamirpour, III Telford, Sam, Eric Rosenberg, Tracey Cho, Matthew P Frosch, Marcia B Goldberg, Shibani S Mukerji, and Pardis C Sabeti. Rapid Detection of Powassan Virus in a Patient With Encephalitis by Metagenomic Sequencing. *Clinical Infectious Diseases*, 66(5):789–792, 09 2017.
- [8] Lucas B Harrington, David Burstein, Janice S Chen, David Paez-Espino, Enbo Ma, Isaac P Witte, Joshua C Cofsky, Nikos C Kyrpides, Jillian F Banfield, and Jennifer Doudna. Programmed dna destruction by miniature crispr-cas14 enzymes. *Science.*, 2018.
- [9] Martin Steinegger and Johannes Soeding. Clustering huge protein sequence sets in linear time. *Nature Communications*, 2018.
- [10] Kazutaka Katoh and Daron M Standley. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Mol Biol Evol.*, 2013.
- [11] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Soeding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature Methods*, 2012.

- [12] Morgan N Price, Paramvir S Dehal, and Adam P Arkin. Fasttree 2 – approximately maximum-likelihood trees for large alignments. *PLoS ONE.*, 2010.
- [13] Andrei Kouranov, Lei Xie, Joanna de la Cruz, Li Chen, John Westbrook, Philip E Bourne, and Helen M Berman. The rcsb pdb information portal for structural genomics. *Nucleic acids research*, 34(suppl_1):D302–D305, 2006.
- [14] Elmar Krieger, Sander B Nabuurs, and Gert Vriend. Homology modeling. *Methods of biochemical analysis*, 44:509–524, 2003.
- [15] Marco Biasini, Stefan Bienert, Andrew Waterhouse, Konstantin Arnold, Gabriel Studer, Tobias Schmidt, Florian Kiefer, Tiziano Gallo Cassarino, Martino Bertoni, Lorenza Bordoli, et al. Swiss-model: modelling protein tertiary and quaternary structure using evolutionary information. *Nucleic acids research*, 42(W1):W252–W258, 2014.
- [16] Jiuyu Wang, Jiazhi Li, Hongtu Zhao, Gang Sheng, Min Wang, Maolu Yin, and Yanli Wang. Structural and mechanistic basis of pam-dependent spacer acquisition in crispr-cas systems. *Cell*, 163(4):840–853, 2015.