Protein Bioinformatics

Ruoshi Zhang Venket Raghavan Michel van Kempen Yazhini
Alexandra Kolodyazhnaya Johannes Söding

November 09 – 10, 2021

Contents

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/project/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 environment[1] 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://tutorialNN.mmseqs.com` and open it in your browser.

[1] `https://github.com/cdr/code-server`

We recommend Firefox, but any browser should work. If you want to download any of the files you produce to your own computer (e.g. for uploading it to a webserver) you can open `https://tutorialNN.mmseqs.com/web` and download the files from there.

You should see something like the following image:

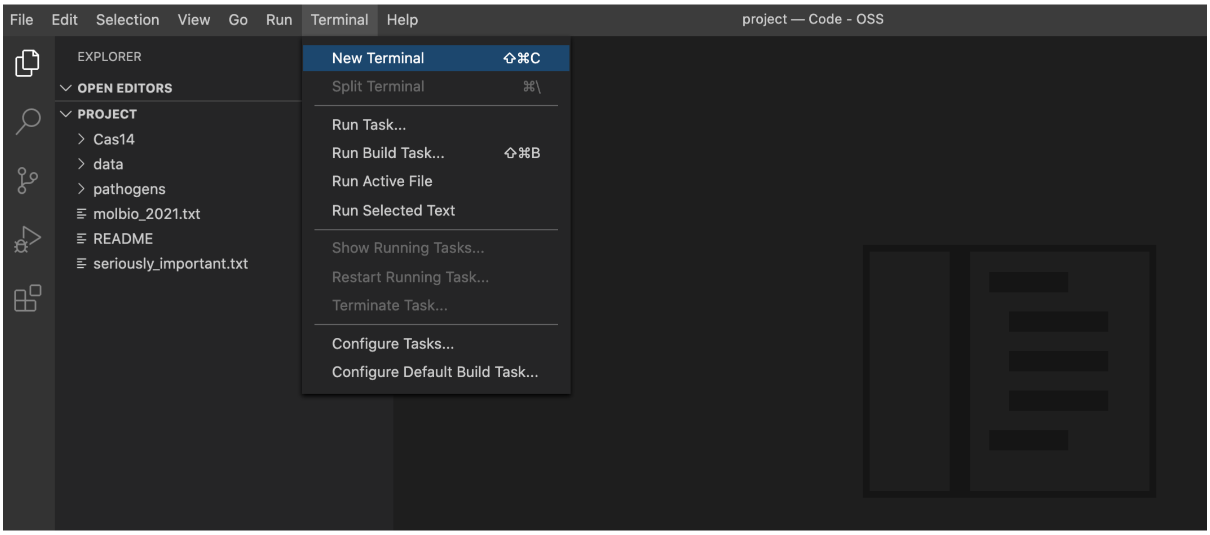


Figure 1.1: You can open a new terminal by clicking “Terminal -> New Terminal”.

Now, in the Bash window, let’s type the following commands (Lines that start with # are comments and will not be executed if entered):

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

This should result in navigating to 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

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

Try the following keystrokes:
`cat`SPACE `u`TAB
It should get expanded to the same command as above (as long as you are in the correct directory). You should liberally use TAB-expansion as it will reduce the number of typos you will make.

Bash Tip 2: Use the ↑ ↓ arrow keys to navigate to the previous commands you executed.

Today we will use the integrated text editor to make changes to files instead of also using a shell based text editor. When you have some time you should try to familiarize yourself with one of the popular shell based editors such as `nano`, `vim` or `emacs`.
In this tutorial, whenever you see **YourSomething** it means you need to replace it with a sensible 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 last 5 lines of a file:
tail -n 5 useful_links.txt
```

Visually confirm that useful_links.txt and YourFileNameCopy have the same contents.

```
# lists the files in more detail
ls -lah

# 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
```

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

Bash Tip 3: To cancel a running program you can press CTRL + C.

Bash Tip 4: Whenever you are not sure about what a command does or how to run it, you can always look up its manual page with the following command:

```
# show the manual page of a command (quit by pressing 'q')
man <commandtolookup>
# E.g., man mkdir
```

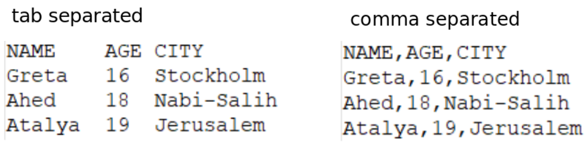
1.3 Text processing in Bash

In Bash, we can take textual data and transform it in a particular way that is more useful for us. We will introduce a few text processing commands in this section.

Note these commands usually have various command line options that will modify their behavior. Some more commands used in this section are described in the appendix 5.1.

The **cut** command lets you select certain columns from a text file if your content is separated into columns.
Options (flags [2]):

- `-f`: indicates columns to print (e.g.: 1,4-9,12-)
- `-d`: specifies column separator character (e.g.: ,), the default separator is the tab character



tab separated

```
NAME    AGE  CITY
Greta   16   Stockholm
Ahed    18   Nabi-Salih
Atalya  19   Jerusalem
```

comma separated

```
NAME,AGE,CITY
Greta,16,Stockholm
Ahed,18,Nabi-Salih
Atalya,19,Jerusalem
```

? Print the first column of `molbio_2021.txt` to the terminal with `cut`

Thus far, commands were entered into the terminal, and the output presented directly (also on the terminal). What if we want to store the output (of a command) in a file?

The **redirection operators** (> and >>), as the name suggests, route the Standard Output (stdout) [3] of a command to a location of the user's choosing.

[2] A flag is an (optional) input or parameter that is passed to a command to extend or modify its functionality. For example, we pass the `-l` flag to `wc` in order to show only the count of lines in a file like so:
`wc -l yourfile`.

[3] The standard output is default place where the Bash command presents its output.

There are two types of redirections at your disposal:

- > creates and/or overwrites(!) the file
- >> appends to the end of the file

? **From the file `molbio_2021.txt` print the country of origin to a file called `nationalities.txt`**

We also only entered a single command at a time. But what if we need to perform some other actions on this output using other Bash commands?

The **pipe operator** (|) passes the output of a command as input to another command.

```
command1 | command2 | command3 ...
```

? **What do these commands do? Guess the function of `uniq` and `sort`.**

```
uniq nationalities.txt
sort nationalities.txt | uniq
```

? **What do these commands do? Can you find out from the `man`-page what these flags mean: `-l`, `-c`, `-nrk1`?**

```
sort nationalities.txt | uniq | wc -l
sort nationalities.txt | uniq -c
sort nationalities.txt | uniq -c | sort -nrk1
```

What if we want to extract certain information from the text file?

grep finds and prints all the lines that match a specific pattern or string in the file(s):

- `-c`: counts occurrences of the pattern
- `-v`: print only the lines that DO NOT contain the pattern
- `-i`: case insensitive flag

? **Try the following command. What does it do?**

```
grep "China" molbio_2021.txt
```

? **Count number of students from *India*.**

? **Count number of students that are not from *Germany*.**

? **How many people contain the word fragment `an` in their names?**

- -E: let's you use *regular expressions* [4]

❢ What does this command do?

```
grep -E "^\w{5}\s" molbio_2020.txt
```

```
'^'  : the beginning of a line
'\w' : any word character(alphanumeric & underscore)
'{5}': exact nº of occurrences of last element
'\s' : any white space character
```

1.4 Programming in Bash

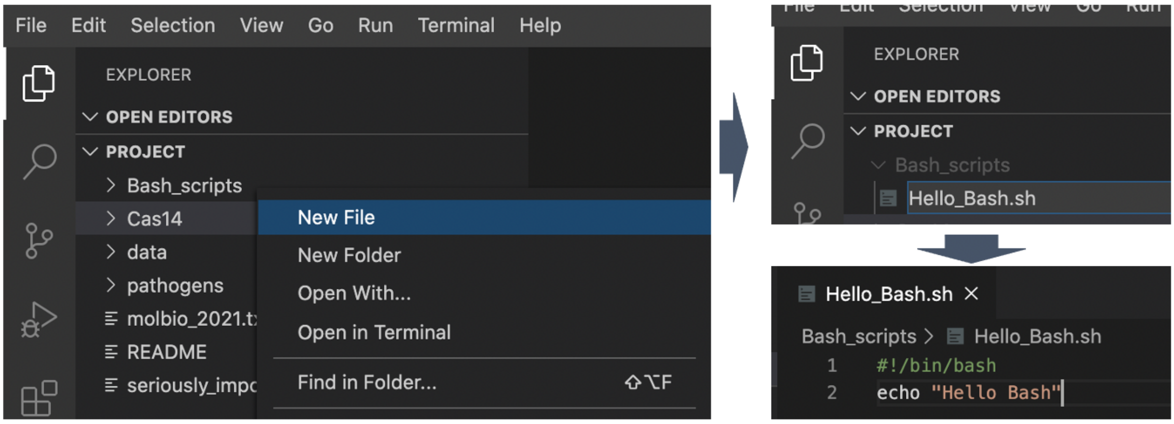
A Bash script is a plain text file which contains a series of commands. Bash programming is useful as it allows you to automate tasks (e.g., manipulating files and executing processes). In the MMseqs2 software suite, we also use Bash scripts to combine its modules and workflows, to create tailored computational tools.

1.4.1 The script file

Now, let's try and print something to the terminal using a self-written Bash script.

Under your home directory, create a new directory called `Bash_scripts`. We will create our Bash scripts here.

Create a new file and rename your file as `Hello_Bash.sh`, similar to the following image. This will be the file where we will enter our Bash commands.



```
#!/bin/bash
echo "Hello Bash"
```

[4] A regular expression is a pattern of meta-characters that is used to describe one or more strings of interest. For instance, think about how you would generically describe to someone–verbally–the way the date is written here: 20-04-2020. It would probably be something along the lines of "day hyphen month hyphen year", or to be more precise "zero-leading-day hyphen zero-leading-month hyphen four-digit-year". The programmatic equivalent `[0-9]{2}-[0-9]{2}-[0-9]{4}` would be one possible regular expression.

The first line of a Bash script is usually:

```
#!/bin/bash
```

This indicates this file is a Bash script [5]. Add this as the first line in the script.

Our Bash script here will contain a single command that will print "Hello Bash" to the terminal. The command for that is illustrated below. Go ahead and add this command to your script, and then save it.

```
# to print into the terminal
echo "Hello Bash"
```

Now the script can be executed. Almost.

To run your Bash script, you first need to give your script permission to execute:

```
chmod +x ~/project/Bash_scripts/Hello_Bash.sh
```

Now you can run it from the terminal.

Bash Tip 5: `~` means your **home directory**. Try the following:

```
echo $HOME
echo ~
cd ~
```

? Create a Hello_Bash.sh script and run it.

Hint: to run your Bash script, you can run either using the path based on your home directory:

```
~/project/Bash_scripts/Hello_Bash.sh
```

or first `cd` to the directory where the script is, and run it:

```
cd ~/project/Bash_scripts
./Hello_Bash.sh
```

1.4.2 Bash variables

Like any other programming language, Bash also provides variables to store values. There are no variable types in Bash. A variable in Bash can contain a number, a character, or a string of characters.

The assignment of a value to a variable is done by `=` ; note there should be no space around the `=` sign in variable assignment.

Then the value of this variable can be retrieved by putting a `$` before the variable name.

[5] Note: the `#!/bin/bash` sequence is called a **shebang** and is not an ordinary comment. By convention, every script that gets executed, first gets checked for a shebang. If one exists, the script is executed through the program mentioned in it (here: `/bin/bash`). Refer to this Stack Overflow discussion (`https://stackoverflow.com/q/3009192` and links therein) for more details regarding shebangs.

```
#!/bin/bash
NAME="Eli"
NUMBER_OF_EYES=3
echo "Hello $NAME, you have $NUMBER_OF_EYES eyes"
```

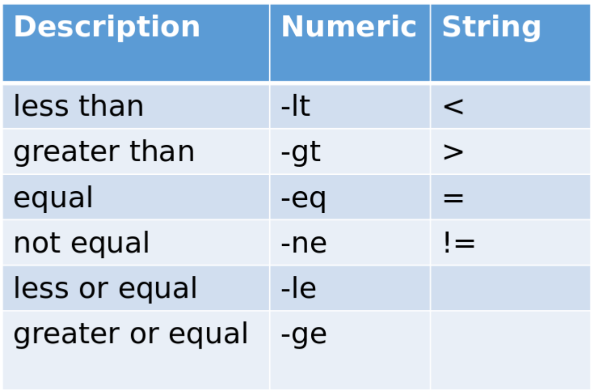
? Modify the Hello_Bash.sh script you created earlier to include a variable, and re-run it.

1.4.3 Conditional execution

`If` statements allow us to make decisions in our Bash scripts, and to execute commands only in certain cases.

```
AGE=20
if [ "$AGE" -eq 20 ]; then
    echo "Wow, you are exactly 20!"
fi
```

Anything between **then** and **fi** (**if** spelled backwards) will be executed only if the test condition (between the square brackets) is true. Some commonly used conditional operators are listed here:



Description	Numeric	String
less than	-lt	<
greater than	-gt	>
equal	-eq	=
not equal	-ne	!=
less or equal	-le	
greater or equal	-ge	

? Create a script with variable named `AGE` and serves beer only if the `AGE` is at least 18.

Bash Tip 6: There are many, many more features to Bash! Check out this resource to learn more: `https://ryanstutorials.net/linuxtutorial`

1.5 File formats

Biological information is conventionally stored in specific textual formats. The contents of such files are arranged in such a way that each unique kind of data within the file(s) is indicated clearly and unambiguously[6]. For example, there are file formats that store the name and polypeptide sequence of proteins. The data is demarcated in such a way

[6] uhm, yeah right

that the name string can be disambiguated from the sequence string. This way bioinformatic tools can extract the needed information from the files efficiently, without confusion and/or mistakes.

One of the most common bioinformatics file formats is called **FASTA**. FASTA-formatted files are typically identified by the filename extensions `.fa` or `.fasta` (e.g., myproteins.fasta). In the FASTA format, an identifier (a protein name, for example) is written after the ">" symbol, and its corresponding sequence is written in the lines following it. This format is used, for example, to store metagenomics sequence reads.

Another popular bioinformatics file format is the **TSV** (tab separated values) format. TSV-formatted usually files have the extension `.tsv` after the filename (e.g., mysamples.tsv). TSV files contain one record per line, with the contents of each line itself being separated by TAB characters. This file format is commonly used to represent tabular data in bioinformatics (e.g., a set of samples, species identities for each sample, and the rRNA sequence of each sample). TSV files are very popular as they are easy to explore with standard Linux tools (and most bioinformatics tools themselves are often Linux-based). This is a file format you will be working with later in the tutorial.

We will present examples of both FASTA files and TSV files later in the tutorial.

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 of the patient?**
- ? **What is the SRA? How many bases are currently 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 of 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 Swiss-Prot 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
```

In this file you see a summarized view of the data with the following columns: (1) the percent of reads covered by the clade rooted at this taxon, (2) number of reads covered by the clade rooted at this taxon, (3) number of reads assigned directly to this taxon, (4) rank, (5) taxonomy identifier, and (6) scientific name.

? **Based on `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?**

[1] The 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]. This offers an interactive circular visualization where you can click on each label to zoom into different parts of the hierarchy.

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. 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://tutorialNN.mmseqs.com/web` in a new tab. There you will see your `report.html` file. (Don't forget to replace the NN with the number assigned to you.)

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 taxonomic identifier(s) you just found. If you found multiple identifiers, 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 (i.e., reads) that were **filtered out**, meaning they were annotated as pathogenic.

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

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

```
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 into 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 tool Plass [6] to find overlapping reads and generate whole proteins out of the best matching ones.

```
plass 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

Plass 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

Proteins are generally comprised of one or more functional regions, called **domains**. Identifying the domains in a protein provides insights to the function of the protein. We will look for known protein domains to identify the proteins we found.

For this, we will use MMseqs2 search module to search all the representative sequences contained in `assembly_clustered_rep_seq.fasta` against the Pfam database. The Pfam database is a large collection of protein domain families. Each family is represented by multiple sequence alignments (MSAs). The Pfam MSA database was downloaded, and the MSAs have been converted to sequence profile database with MMseqs2. The Pfam profile database is stored as `pfamAfull` in the `pathogens` directory.

```
mmseqs easy-search assembly_clustered_rep_seq.fasta pfamAfull pfam_result.html tmp
↪  --format-mode 3
```

The search results are generated as a `HTML` file that can be opened in a browser. Download the `pfam_result.html` from the URL `https://tutorialNN.mmseqs.com/web` in a new tab. (Don't forget to replace the NN with the number assigned to you.) Open `pfam_result.html`. You can navigate through the representative protein sequences to find out about the matched PFAM domains and visualize how they are aligned with the query proteins.

Look up some of the PFAM domain entries that were matched. Which of the expected protein (domains) 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), and are widely used in genome editing tools. Recently, Harrington and Burstein et al. [8] discovered CRISPR-Cas systems in archaea that consist of previously unreported Cas14 proteins. These proteins are compact RNA-guided nucleases (400 to 700 amino acids in length). Due to its compact size and special enzymatic property, it has the potential to be exploited for gene editing tools, like Cas9. In their work, the authors identified a set of 45 Cas14 proteins by constructing and iteratively refining hidden Markov models (HMMs) of known Cas14 proteins, and using them to query public metagenomes from the IMG/M database.

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 diversity and functionality of the protein.

In this section you will learn how to:

- create and visualize multiple sequence alignments (MSAs) of protein sequences
- compute a phylogenetic tree
- visualize and interpret the phylogenetic tree

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

3.3 Input

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

[1] https://www.science.org/doi/suppl/10.1126/science.aav4294/suppl_file/aav4294_data_s2.fasta

1. Change to the exercise directory: Cas14

2. Download the sequences by using the command:
   `wget <url_to_sequences_see_footnote>`

❢ **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 -c "^>" aav4294_Data_S2.fasta
# the number of sequences is: 45

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

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

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

3.4 How we searched for candidate orthologs

Our chances of finding highly diverse orthologs increase as we explore more comprehensive protein databases. We thus chose to look for candidates in the "**BFD**" (**B**ig well... let's just say... **F**antastic **D**atabase; `https://bfd.mmseqs.com/`). The BFD is constructed from 2,500,000,000 protein sequences of various sources, including environmental samples from soils and water bodies. The BFD is clustered with Linclust[9] at 30% sequence identity to produce 65,983,866 clusters. Each cluster is represented by a multiple sequence alignment (MSA). The BFD was also used by AlphaFold2. (More details on this tomorrow)

Like Harrington and Burstein et al., we search for candidate orthologs of Cas14 among the BFD MSAs. We constructed a MSA from the aforementioned 45 Cas14 proteins. Subsequently, using HHblits[10], we searched the MSAs from the BFD using the MSA of the Cas14 proteins. This yielded a set of undiscovered Cas14 candidate proteins.

We have performed these steps for you already, as the BFD is too large to be manipulated in this tutorial environment. The candidate Cas14 proteins we found are available in the file `cas14_bfd_candidates.fasta`.

3.5 Aligning known Cas14 and BFD candidates

The **cas14_bfd_candidates.fasta** file contains three types of sequences: the previously reported 45 Cas14 proteins (**"CAS"** headers), sequences that were found in standard reference databases such as UniProt (**"REF"** headers), and sequences of environmental/metagenomic origin (**"ENV"** headers).

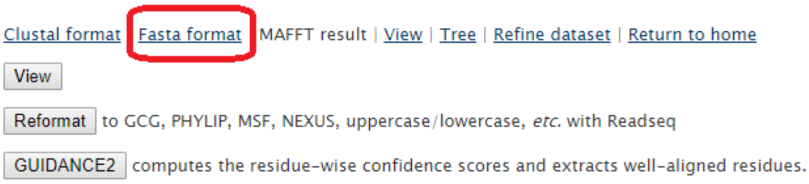
1. Using Bash commands, inspect `cas14_bfd_candidates.fasta` file.

   ? How many sequences are there of each type?

   Solution:

   ```
   grep ">CAS" cas14_bfd_candidates.fasta | wc -l        # 45
   grep ">REF" cas14_bfd_candidates.fasta | wc -l        # 25
   grep ">ENV" cas14_bfd_candidates.fasta | wc -l        #164
   ```

2. Download `cas14_bfd_candidates.fasta` from `https://tutorialNN.mmseqs.com/web` (replace NN with your number). Align all these sequences using the MAFFT online server and save the result as `cas14_bfd_candidates_MSA.fasta`.[2]

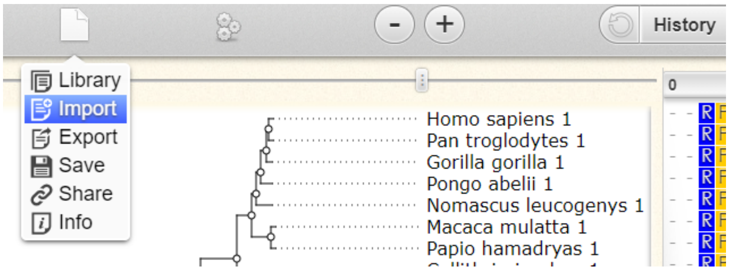


Clustal format | Fasta format | MAFFT result | View | Tree | Refine dataset | Return to home
View

Reformat to GCG, PHYLIP, MSF, NEXUS, uppercase/lowercase, *etc.* with Readseq

GUIDANCE2 computes the residue-wise confidence scores and extracts well-aligned residues.

3. Upload the MSA file to the Wasabi MSA viewer.[3]

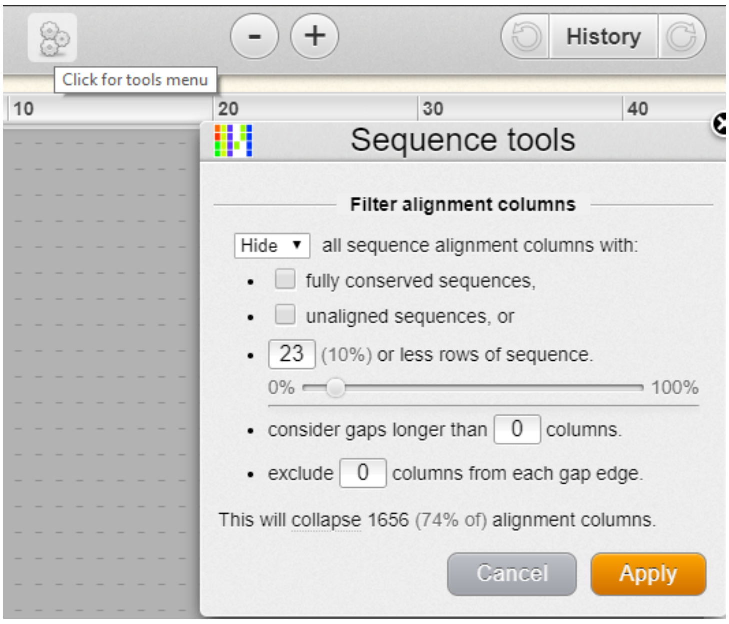


4. Scroll and zoom in and out to get an overall impression.

5. Use the "collapse gaps" option:

[2] `https://mafft.cbrc.jp/alignment/server/`

[3] `http://wasabiapp.org/`



Filter alignment columns

Hide ▾ all sequence alignment columns with:

- □ fully conserved sequences,
- □ unaligned sequences, or
- 23 (10%) or less rows of sequence.

  0% 100%
- consider gaps longer than 0 columns.
- exclude 0 columns from each gap edge.

This will collapse 1656 (74% of) alignment columns.

? **What can you say about the MSA? What is its length?**

? **How much did the length of the MSA change after collapsing the gaps?**

? **What is the utility of collapsing gaps in an MSA?**

3.6 Computing a phylogenetic tree

A phylogentic tree represents the reconstructed evolution leading to the sequences in a multiple sequence alignment (MSA). There are several ways[4] 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 tools[5] implement this tree reconstruction strategy. Today, we will use FastTree[6][11], which approximates the maximum likelihood computation to achieve short running times.

Reconstruct a phylogenetic tree using FastTree. To move the MSA results to your environment, create an empty file named `cas14_bfd_candidates_MSA.fasta` and paste the results there.

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

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

[5] https://molbiol-tools.ca/Phylogeny.html

[6] http://www.microbesonline.org/fasttree/

3.7 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) 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].

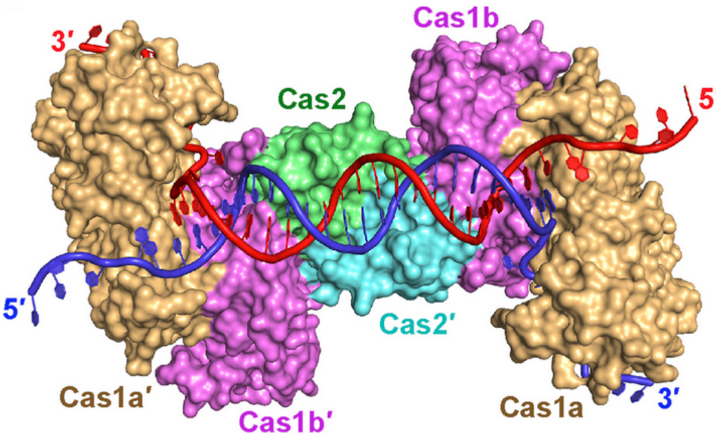
We have prepared an annotation file (`cas14_bfd_candidates_iTOL_leaves.txt`) based on the iTOL format of coloring leave labels. Drag and drop this file on your tree.

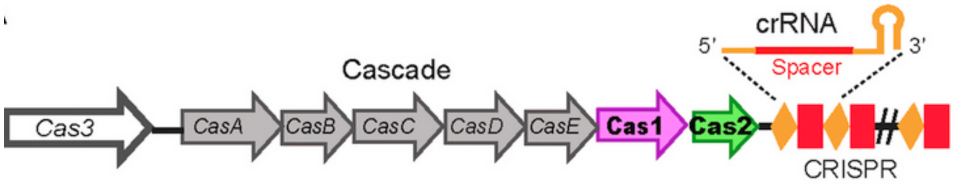
Questions:

- **What can you say about the diversity of sequences in the environmental metagenomic samples, compared to the standard reference database?**
- **Can you think of other bioinformatic approaches you can take to verify if they are likely true orthologs?**
- **How can you validate those predictions by experimental approaches?**

[7] `https://itol.embl.de/`

Protein structure prediction





In this section you will learn how to:

1. Predict the 3-D structure of a protein (Cas1) with AlphaFold
2. Search for protein structures on the websites UniProt[4] and RCSB PDB[12]
3. Use visualization tools to explore protein structures and the interface of proteins and DNA

Have fun!

4.1 Prediction of Cas1 protein structures using ColabFold

Cas1: CRISPR-associated protein 1 (Cas1) is a widely conserved component of the CRISPR adaptive immune system. It functions as a metal-dependent, DNA-specific endonuclease. It forms a complex with Cas2 to integrate phage DNA into the CRISPR array of the host (bacterial) genome. In this tutorial, we will work with Cas1 from *E. coli* strain K12.

ColabFold:

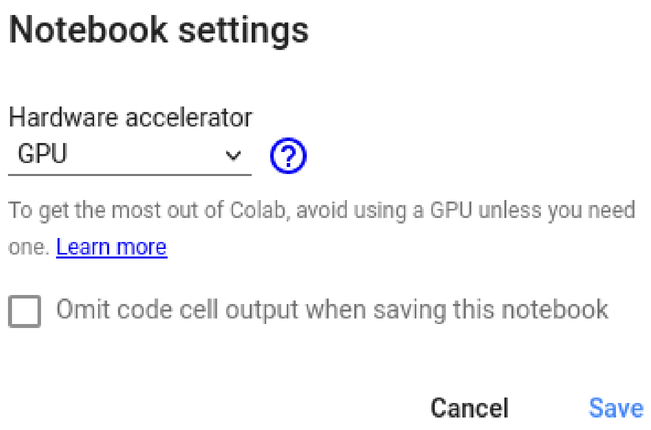
ColabFold is an easy-to-use, Google Colab-based implementation of the AlphaFold2 structure prediction suite. ColabFold [13] makes use of both to offer a simple, user friendly and fast tool to predict 3-D structures of proteins. AlphaFold2 predicts protein 3-D structures based on MSAs. Google Colab offers free CPU and, importantly, free GPU resources for running Jupyter Notebooks.

Tips for Colab:

1. You can show/hide the code with **View** → **Show/Hide code, or double click in the field**

1. Open the ColabFold Notebook[1] in Google Colab and sign in with your Google account. The usage of Google Colab is free, but requires a Google account.
2. A GPU is required for the structure prediction, therefore configure the notebook to use a GPU: **Runtime** → **Change Runtime type**

[1] `https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb`



Notebook settings

Hardware accelerator
GPU

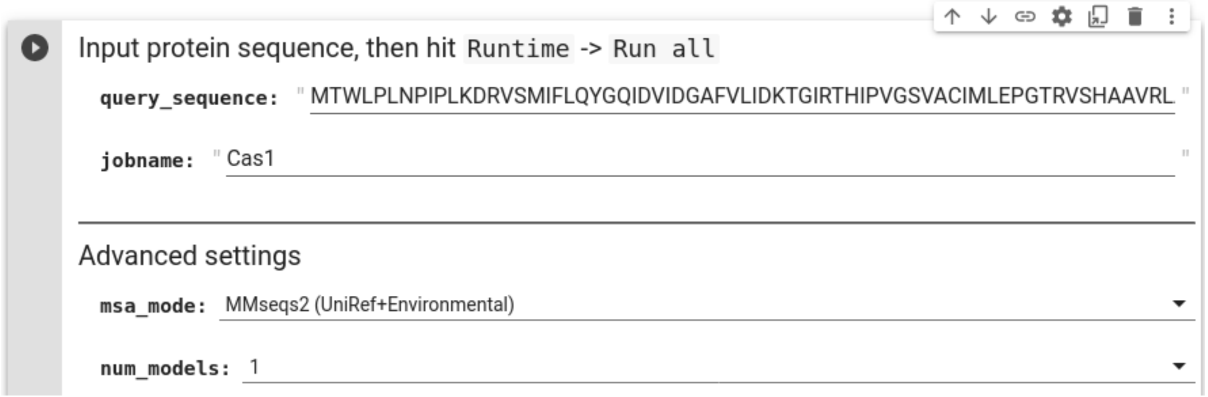
To get the most out of Colab, avoid using a GPU unless you need one. Learn more

□ Omit code cell output when saving this notebook

3. First enter the amino acid sequence of the protein into the field `query_sequence`. Then select `msa_mode` as MMseqs(UniRef+Environmental). By default, AlphaFold predicts five different structures and we can choose the best model afterwards. However, this would take half an hour for this protein. Therefore, set `num_models` to 1. You can give any jobname as you prefer. We used "Cas1" here.

   Sequence information can be shown in Cas1_Q46896.fasta:

```
>sp|Q46896|CAS1_ECOLI CRISPR−associated endonuclease Cas1
OS=Escherichia coli (strain K12) OX=83333 GN=ygbT PE=1 SV=1
MTWLPLNPIPLKDRVSMIFLQYGQIDVIDGAFVLIDKTGIRTHIPVGSVACIMLEPGT
RVSHAAVRLAAQVGTLLVWVGEAGVRVYASGQPGGARSDKLLYQAKLALDEDLRL
KVVRKMFELRFGEPAPARRSVEQLRGIEGSRVRATYALLAKQYGVTWNGRRYDPK
DWEKGDTINQCISAATSCLYGVTEAAILAAGYAPAIGFVHTGKPLSFVYDIADIIKFD
TVVPKAFEIARRNPGEPDREVRLACRDIFRSSKTLAKLIPLIEDVLAAGEIQPPAPPE
DAQPVAIPLPVSLGDAGHRSS
```



Input protein sequence, then hit `Runtime -> Run all`

`query_sequence:` " MTWLPLNPIPLKDRVSMIFLQYGQIDVIDGAFVLIDKTGIRTHIPVGSVACIMLEPGTRVSHAAVRL "

`jobname:` " Cas1 "

Advanced settings

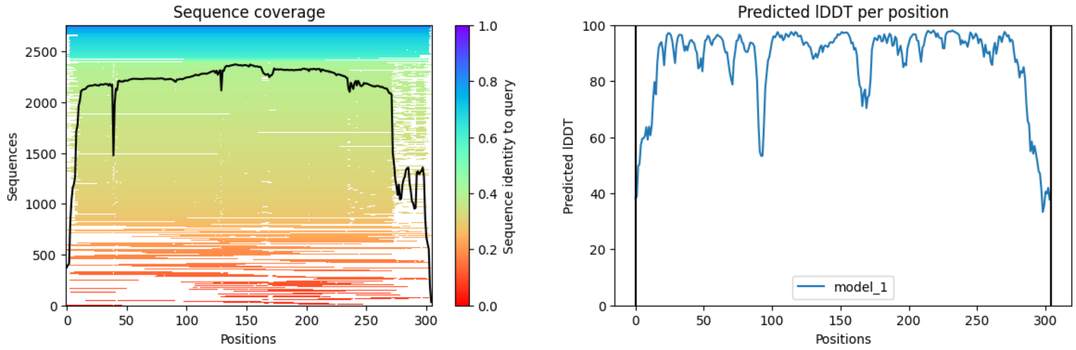
`msa_mode:` MMseqs2 (UniRef+Environmental)

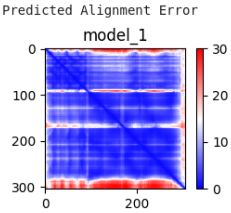
`num_models:` 1

4. To start the prediction hit **Runtime** → **Run all** (This will take some minutes...)

5. The prediction results can be visualized with the plots below.

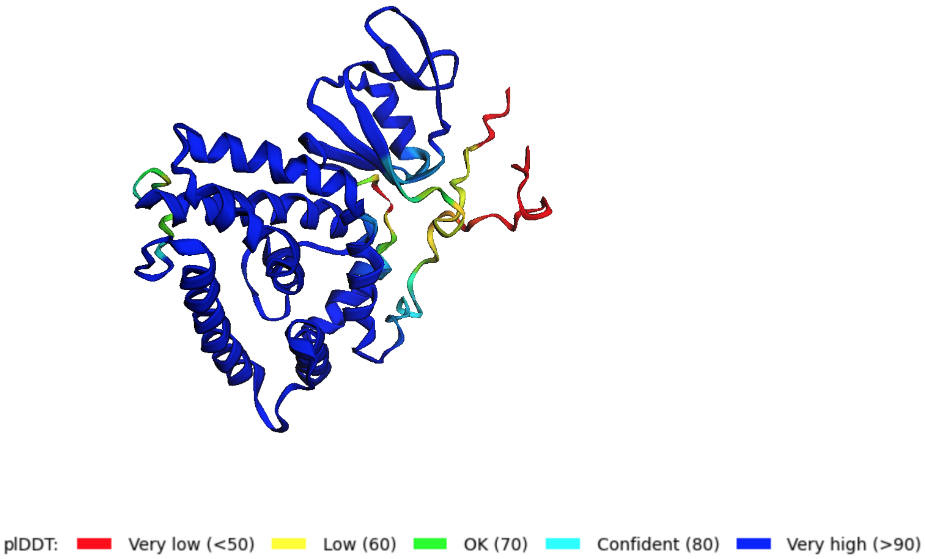
   ? **How confident is AlphaFold2 in its prediction and how good is the input MSA? Interpret the prediction quality by checking the plots (lDDT = local Distance Difference Test).**



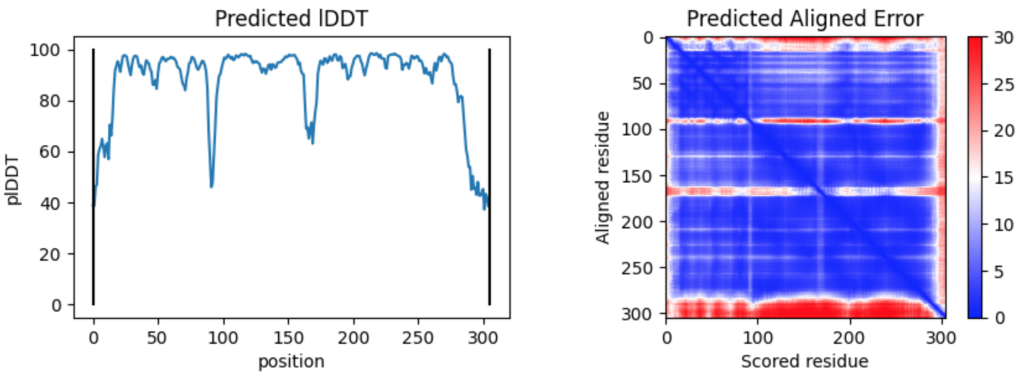


6. Check the predicted Cas1 3-D structure (model 1). Have fun playing with the cartoon view (Ribbon-diagram).

Show code



7. Take a closer look at the confidence and quality measures of model 1.

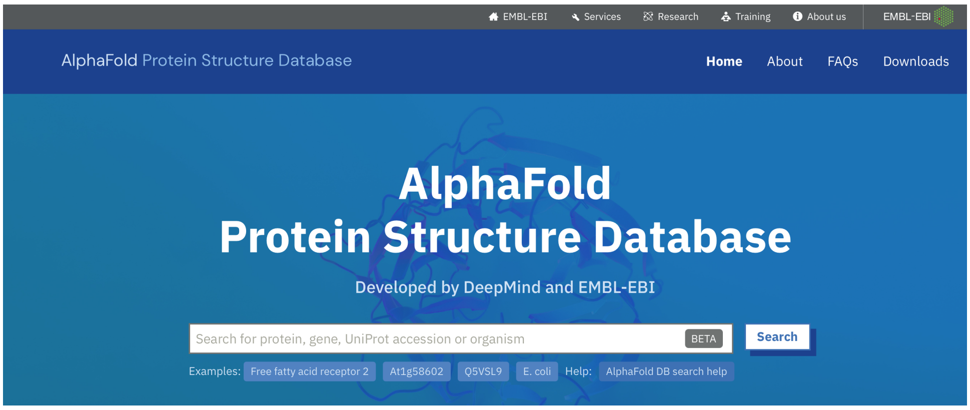


8. You can download the resulting structures as PDB files.

   Note: Instructions for how to use ColabFold, descriptions about the results and acknowledgements can be found at the bottom of the Colab page.

4.2 AlphaFold Protein Structure Database

EMBL-EBI and DeepMind have together developed a database for protein structure models predicted by AlphaFold (`https://alphafold.ebi.ac.uk`). Currently, it has the 3-D models for the complete human proteome and 20 other reference organisms such as *Arabidopsis thaliana*, *Caenorhabditis elegans*, *Danio rerio*, and *Rattus norvegicus*. You can retrieve predicted protein 3-D structures using keywords such as protein name, Gene ID, UniProt ID or species name.



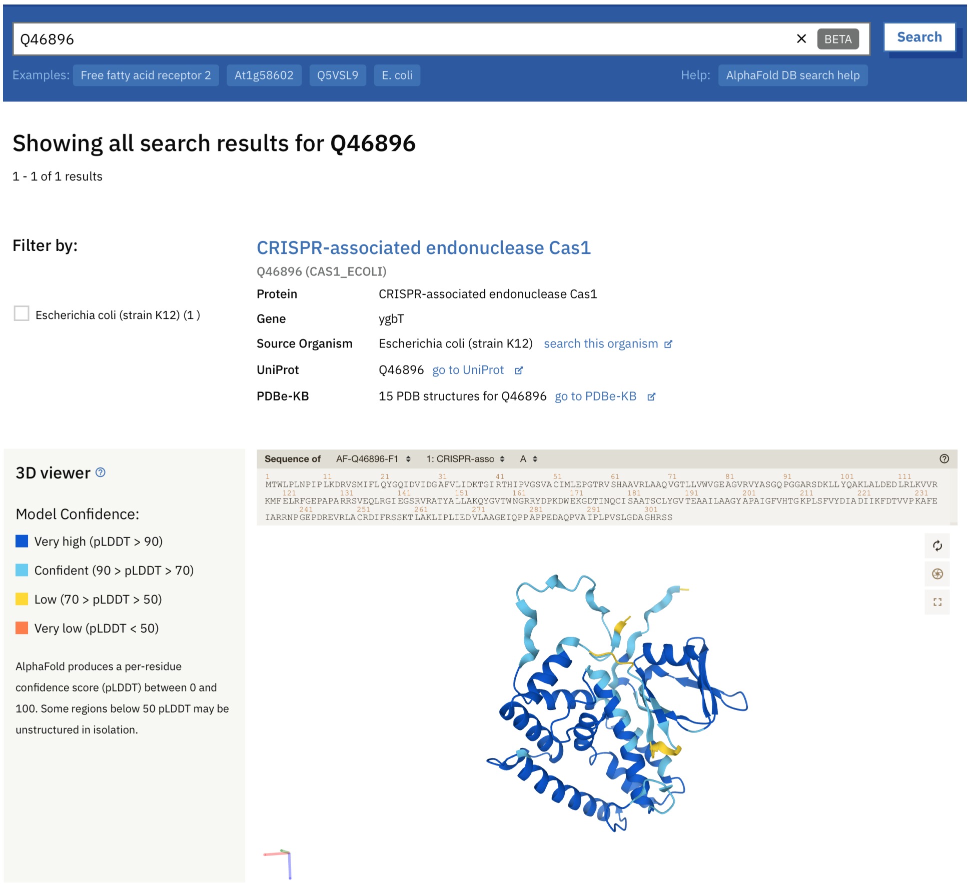
AlphaFold
Protein Structure Database

Developed by DeepMind and EMBL-EBI

Search for protein, gene, UniProt accession or organism BETA | Search

Examples: Free fatty acid receptor 2 | At1g58602 | Q5VSL9 | E. coli | Help: AlphaFold DB search help

Search for Cas1 protein using UniProt ID **Q46896** in the search box. You will find the details of Gene name, Source Organism, PDBe-KB link (if experimental structure is available) and predicted model.


Q46896
×
BETA
Search
Examples: Free fatty acid receptor 2 | At1g58602 | Q5VSL9 | E. coli
Help: AlphaFold DB search help

Showing all search results for Q46896

1 - 1 of 1 results

Filter by:

□ Escherichia coli (strain K12) (1)

CRISPR-associated endonuclease Cas1

Q46896 (CAS1_ECOLI)

Protein	CRISPR-associated endonuclease Cas1
Gene	ygbT
Source Organism	Escherichia coli (strain K12) search this organism
UniProt	Q46896 go to UniProt
PDBe-KB	15 PDB structures for Q46896 go to PDBe-KB

3D viewer

Model Confidence:

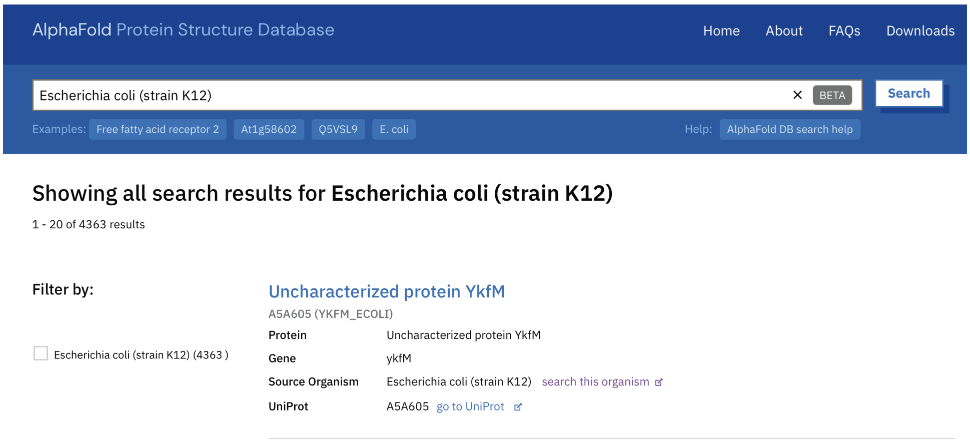
- 🟦 Very high (pLDDT > 90)
- 🟦 Confident (90 > pLDDT > 70)
- 🟨 Low (70 > pLDDT > 50)
- 🟧 Very low (pLDDT < 50)

AlphaFold produces a per-residue confidence score (pLDDT) between 0 and 100. Some regions below 50 pLDDT may be unstructured in isolation.

Sequence of AF-Q46896-F1 | 1: CRISPR-asso | A

```
1          11         21         31         41         51         61         71         81         91         101        111
MTWLPLNPIP LKDRVSMIFL QYGQIDVIDG AFVLIDKTGI RTHIPVGSVA CIMLEPGTRV SHAAVRLAAQ VGTLLVWVGE AGVRVYASGQ PGGARSDKLL YQAKLALDED LRLKVVR
    121        131        141        151        161        171        181        191        201        211        221        231
KMF ELRFGEPAPA RRSVEQLRGI EGSRVRATYA LLAKQYGVTW NGRRYDPKDW EKGDTINQCI SAATSCLYGV TEAAILAAGY APAIGFVHTG KPLSFVYDIA DIIKFDTVVP KAFE
      241        251        261        271        281        291        301
IARRNP GEPDREVRLA CRDIFRSSKT LAKLIPLIED VLAAGEIQPP APPEDAQPVA IPLPVSLGDA GHRSS
```

You can also find the models for all proteins in the proteome of the 20 species that they have covered so far.



Escherichia coli (strain K12)

Showing all search results for **Escherichia coli (strain K12)**

1 - 20 of 4363 results

Filter by:

□ Escherichia coli (strain K12) (4363)

Uncharacterized protein YkfM

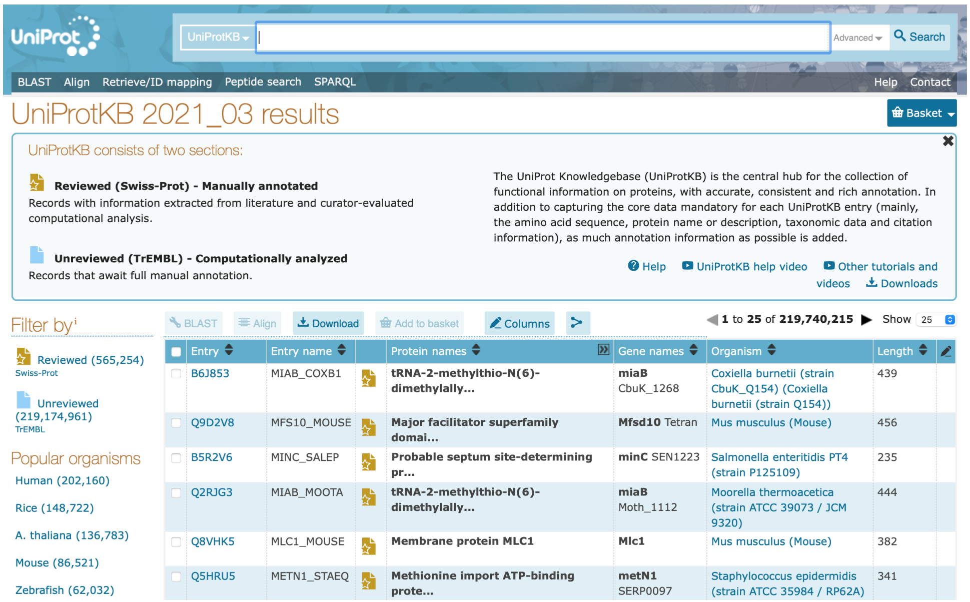
A5A605 (YKFM_ECOLI)

Protein	Uncharacterized protein YkfM
Gene	ykfM
Source Organism	Escherichia coli (strain K12)
UniProt	A5A605

In the coming months, the database will provide 3-D models for a large proportion of all catalogued proteins in the UniProt.

4.3 Understand more about the protein in UniProt Database

1. **UniProt** is a comprehensive, high-quality and freely accessible resource for protein sequence and functional information. Go to the UniProt website: `https://www.uniprot.org/`.



UniProtKB 2021_03 results

UniProtKB consists of two sections:

Reviewed (Swiss-Prot) - Manually annotated
Records with information extracted from literature and curator-evaluated computational analysis.

Unreviewed (TrEMBL) - Computationally analyzed
Records that await full manual annotation.

The UniProt Knowledgebase (UniProtKB) is the central hub for the collection of functional information on proteins, with accurate, consistent and rich annotation. In addition to capturing the core data mandatory for each UniProtKB entry (mainly, the amino acid sequence, protein name or description, taxonomic data and citation information), as much annotation information as possible is added.

Help · UniProtKB help video · Other tutorials and videos · Downloads

Filter by
Reviewed (565,254) Swiss-Prot
Unreviewed (219,174,961) TrEMBL

Popular organisms
Human (202,160)
Rice (148,722)
A. thaliana (136,783)
Mouse (86,521)
Zebrafish (62,032)

1 to 25 of 219,740,215

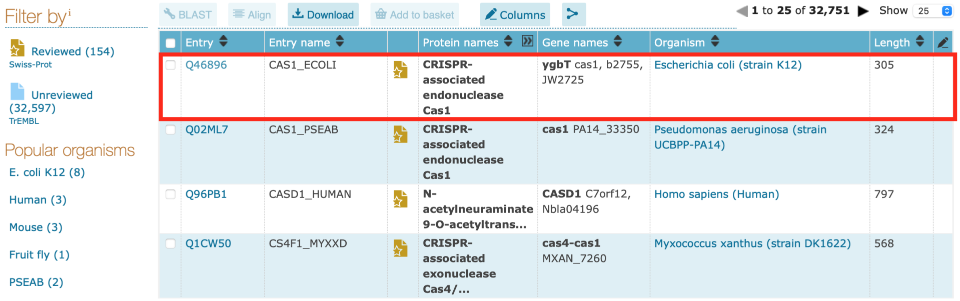
□	Entry	Entry name		Protein names	Gene names	Organism	Length
□	B6J853	MIAB_COXB1	☆	**tRNA-2-methylthio-N(6)-dimethylally...**	**miaB** CbuK_1268	Coxiella burnetii (strain CbuK_Q154) (Coxiella burnetii (strain Q154))	439
□	Q9D2V8	MFS10_MOUSE	☆	**Major facilitator superfamily domai...**	**Mfsd10** Tetran	Mus musculus (Mouse)	456
□	B5R2V6	MINC_SALEP	☆	**Probable septum site-determining pr...**	**minC** SEN1223	Salmonella enteritidis PT4 (strain P125109)	235
□	Q2RJG3	MIAB_MOOTA	☆	**tRNA-2-methylthio-N(6)-dimethylally...**	**miaB** Moth_1112	Moorella thermoacetica (strain ATCC 39073 / JCM 9320)	444
□	Q8VHK5	MLC1_MOUSE	☆	**Membrane protein MLC1**	**Mlc1**	Mus musculus (Mouse)	382
□	Q5HRU5	METN1_STAEQ	☆	**Methionine import ATP-binding prote...**	**metN1** SERP0097	Staphylococcus epidermidis (strain ATCC 35984 / RP62A)	341

2. Search for **CRISPR Cas1**.

? **How many entries do you get in the result table? How many of them are manually curated reviewed entries?**

(Answer: 32,751; 154)

3. Select the first entry (**Q46896**) corresponding to *E. coli* (strain K12).



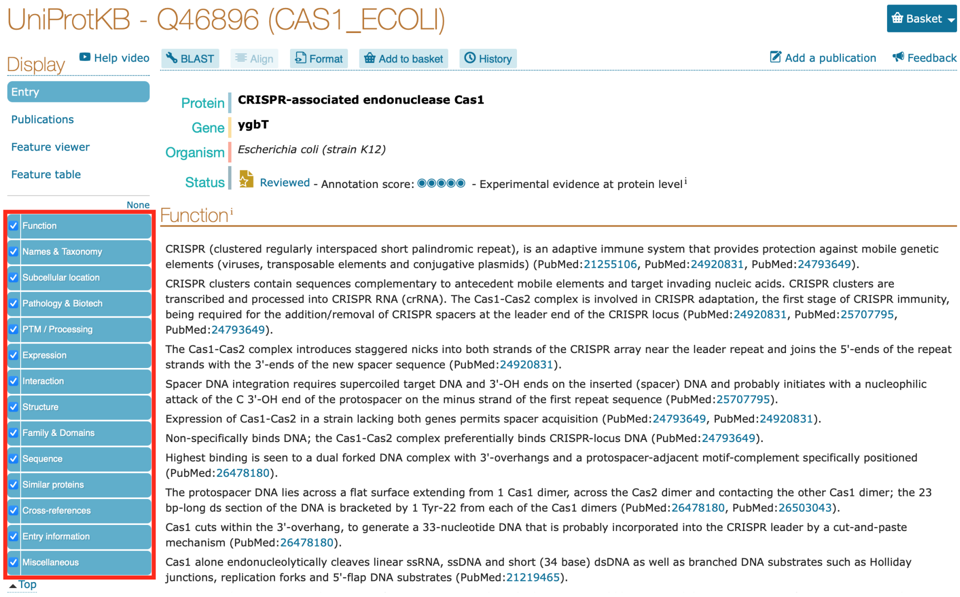
Filter by
Reviewed (154) Swiss-Prot
Unreviewed (32,597) TrEMBL

Popular organisms
E. coli K12 (8)
Human (3)
Mouse (3)
Fruit fly (1)
PSEAB (2)

1 to 25 of 32,751

□	Entry	Entry name		Protein names	Gene names	Organism	Length
□	Q46896	CAS1_ECOLI	☆	**CRISPR-associated endonuclease Cas1**	**ygbT** cas1, b2755, JW2725	Escherichia coli (strain K12)	305
□	Q02ML7	CAS1_PSEAB	☆	**CRISPR-associated endonuclease Cas1**	**cas1** PA14_33350	Pseudomonas aeruginosa (strain UCBPP-PA14)	324
□	Q96PB1	CASD1_HUMAN	☆	**N-acetylneuraminate 9-O-acetyltrans...**	**CASD1** C7orf12, Nbla04196	Homo sapiens (Human)	797
□	Q1CW50	CS4F1_MYXXD	☆	**CRISPR-associated exonuclease Cas4/...**	**cas4-cas1** MXAN_7260	Myxococcus xanthus (strain DK1622)	568

4. Find the functional description about the protein at the top. Other comprehensive details can be seen by navigating through various sections in the left panel.



UniProtKB - Q46896 (CAS1_ECOLI)

Protein: **CRISPR-associated endonuclease Cas1**

Gene: **ygbT**

Organism: *Escherichia coli (strain K12)*

Status: Reviewed - Annotation score: ●●●●● - Experimental evidence at protein level

Function

CRISPR (clustered regularly interspaced short palindromic repeat), is an adaptive immune system that provides protection against mobile genetic elements (viruses, transposable elements and conjugative plasmids) (PubMed:21255106, PubMed:24920831, PubMed:24793649).

CRISPR clusters contain sequences complementary to antecedent mobile elements and target invading nucleic acids. CRISPR clusters are transcribed and processed into CRISPR RNA (crRNA). The Cas1-Cas2 complex is involved in CRISPR adaptation, the first stage of CRISPR immunity, being required for the addition/removal of CRISPR spacers at the leader end of the CRISPR locus (PubMed:24920831, PubMed:25707795, PubMed:24793649).

The Cas1-Cas2 complex introduces staggered nicks into both strands of the CRISPR array near the leader repeat and joins the 5'-ends of the repeat strands with the 3'-ends of the new spacer sequence (PubMed:24920831).

Spacer DNA integration requires supercoiled target DNA and 3'-OH ends on the inserted (spacer) DNA and probably initiates with a nucleophilic attack of the C 3'-OH end of the protospacer on the minus strand of the first repeat sequence (PubMed:25707795).

Expression of Cas1-Cas2 in a strain lacking both genes permits spacer acquisition (PubMed:24793649, PubMed:24920831).

Non-specifically binds DNA; the Cas1-Cas2 complex preferentially binds CRISPR-locus DNA (PubMed:24793649).

Highest binding is seen to a dual forked DNA complex with 3'-overhangs and a protospacer-adjacent motif-complement specifically positioned (PubMed:26478180).

The protospacer DNA lies across a flat surface extending from 1 Cas1 dimer, across the Cas2 dimer and contacting the other Cas1 dimer; the 23 bp-long ds section of the DNA is bracketed by 1 Tyr-22 from each of the Cas1 dimers (PubMed:26478180, PubMed:26503043).

Cas1 cuts within the 3'-overhang, to generate a 33-nucleotide DNA that is probably incorporated into the CRISPR leader by a cut-and-paste mechanism (PubMed:26478180).

Cas1 alone endonucleolytically cleaves linear ssRNA, ssDNA and short (34 base) dsDNA as well as branched DNA substrates such as Holliday junctions, replication forks and 5'-flap DNA substrates (PubMed:21219465).

- **What is the sequence length of *E. coli* Cas1 protein? Click on the *Sequence* section in the left panel.**

(Answer: 305)

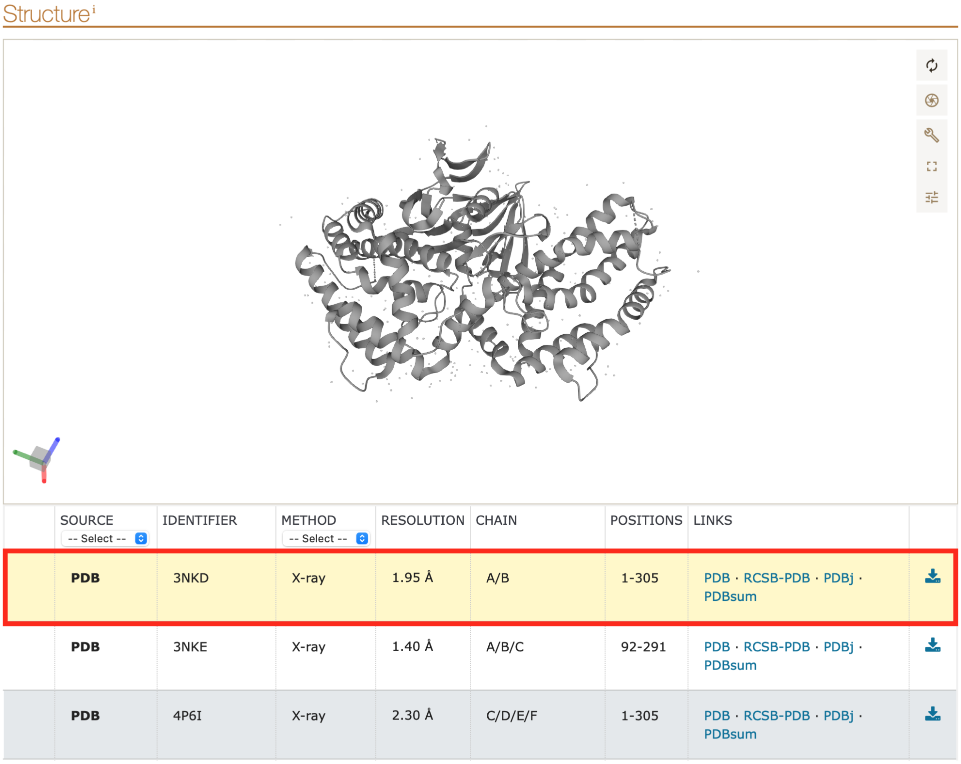
- **Where is this Cas1 protein expressed inside the *E. coli*? Click on the *Subcellular location* section.**

(Answer: Cytoplasm and Cytosol)

- **Does this protein has a experimentally solved structure? Click on the *Structure* section.**

(Answer: Yes)

5. As the table shows, the protein has 15 experimentally solved structures and one predicted model from AlphaFold. In this tutorial we will focus on the first PDB entry **3NKD**.



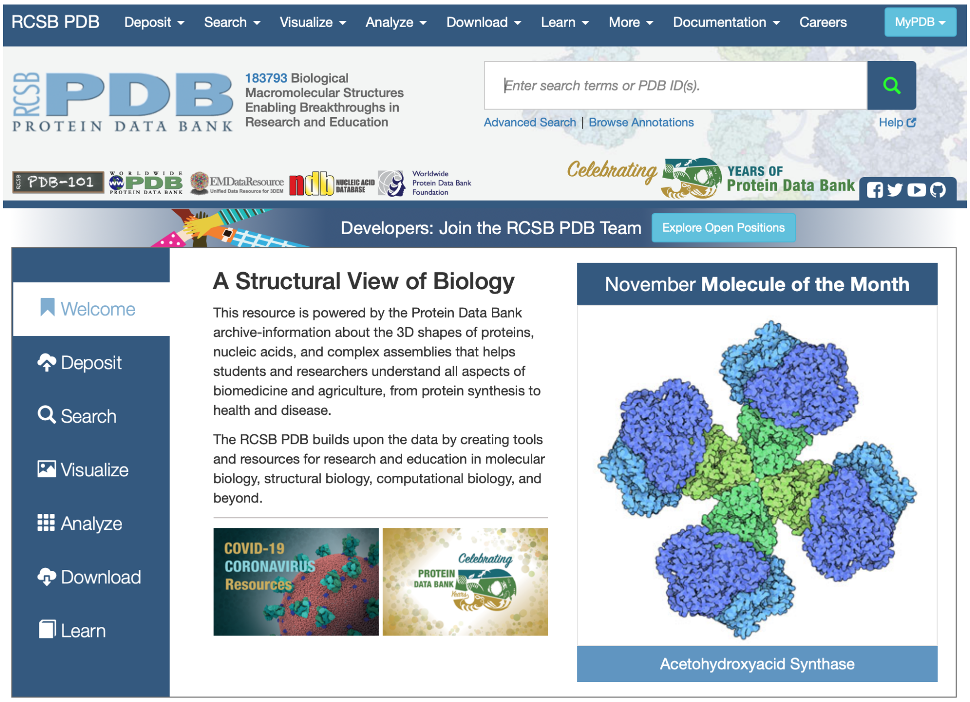
Structure

SOURCE -- Select --	IDENTIFIER	METHOD -- Select --	RESOLUTION	CHAIN	POSITIONS	LINKS
PDB	3NKD	X-ray	1.95 Å	A/B	1-305	PDB · RCSB-PDB · PDBj · PDBsum
PDB	3NKE	X-ray	1.40 Å	A/B/C	92-291	PDB · RCSB-PDB · PDBj · PDBsum
PDB	4P6I	X-ray	2.30 Å	C/D/E/F	1-305	PDB · RCSB-PDB · PDBj · PDBsum

For interested candidates, check out the recently constructed UniProt **beta** version `https://beta.uniprot.org`

4.4 Searching for experimentally solved Cas1 protein structures in the Protein Data Bank (PDB)

1. **RCSB PDB** is a repository for 3-D macromolecular structures (Proteins, nucleic acids and macromolecular complexes). Go to the RCSB PDB website: `http://www.rcsb.org`



183793 Biological Macromolecular Structures Enabling Breakthroughs in Research and Education

Developers: Join the RCSB PDB Team

A Structural View of Biology

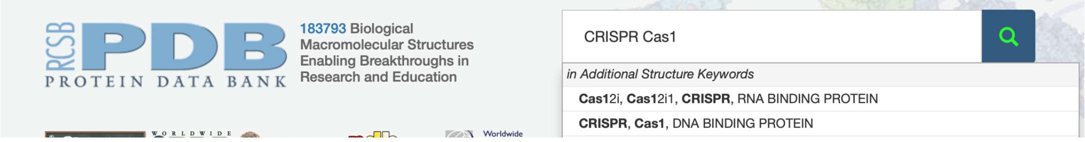
This resource is powered by the Protein Data Bank archive-information about the 3D shapes of proteins, nucleic acids, and complex assemblies that helps students and researchers understand all aspects of biomedicine and agriculture, from protein synthesis to health and disease.

The RCSB PDB builds upon the data by creating tools and resources for research and education in molecular biology, structural biology, computational biology, and beyond.

November Molecule of the Month

Acetohydroxyacid Synthase

2. Search with the keyword **CRISPR Cas1**.



183793 Biological Macromolecular Structures Enabling Breakthroughs in Research and Education

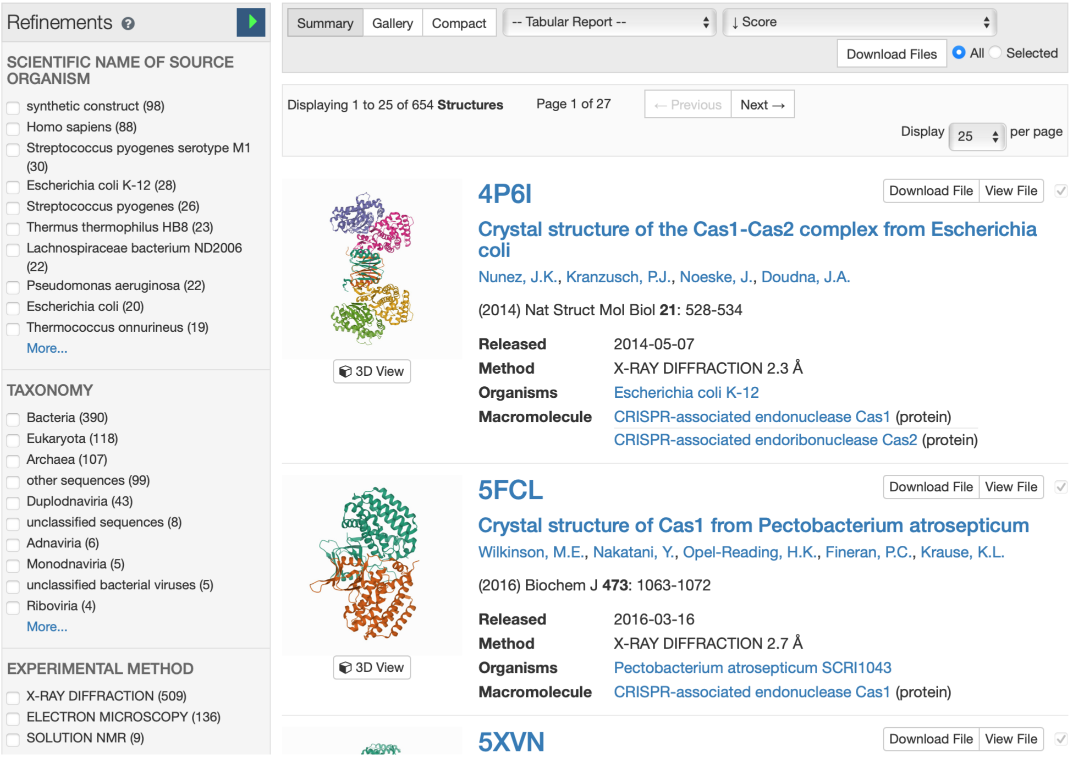
CRISPR Cas1

in Additional Structure Keywords

Cas12i, **Cas12**i1, **CRISPR**, RNA BINDING PROTEIN

CRISPR, **Cas1**, DNA BINDING PROTEIN

3. Explore the result page with different **Refinements** options and the summary of the results.



Refinements

SCIENTIFIC NAME OF SOURCE ORGANISM

□ synthetic construct (98)
□ Homo sapiens (88)
□ Streptococcus pyogenes serotype M1 (30)
□ Escherichia coli K-12 (28)
□ Streptococcus pyogenes (26)
□ Thermus thermophilus HB8 (23)
□ Lachnospiraceae bacterium ND2006 (22)
□ Pseudomonas aeruginosa (22)
□ Escherichia coli (20)
□ Thermococcus onnurineus (19)
More...

TAXONOMY

□ Bacteria (390)
□ Eukaryota (118)
□ Archaea (107)
□ other sequences (99)
□ Duplodnaviria (43)
□ unclassified sequences (8)
□ Adnaviria (6)
□ Monodnaviria (5)
□ unclassified bacterial viruses (5)
□ Riboviria (4)
More...

EXPERIMENTAL METHOD

□ X-RAY DIFFRACTION (509)
□ ELECTRON MICROSCOPY (136)
□ SOLUTION NMR (9)

Summary | Gallery | Compact | -- Tabular Report -- | ↓ Score

Download Files ● All ○ Selected

Displaying 1 to 25 of 654 **Structures** Page 1 of 27 ← Previous Next →

Display 25 per page

4P6I

Download File | View File

Crystal structure of the Cas1-Cas2 complex from Escherichia coli

Nunez, J.K., Kranzusch, P.J., Noeske, J., Doudna, J.A.

(2014) Nat Struct Mol Biol **21**: 528-534

Released	2014-05-07
Method	X-RAY DIFFRACTION 2.3 Å
Organisms	Escherichia coli K-12
Macromolecule	CRISPR-associated endonuclease Cas1 (protein) CRISPR-associated endoribonuclease Cas2 (protein)

3D View

5FCL

Download File | View File

Crystal structure of Cas1 from Pectobacterium atrosepticum

Wilkinson, M.E., Nakatani, Y., Opel-Reading, H.K., Fineran, P.C., Krause, K.L.

(2016) Biochem J **473**: 1063-1072

Released	2016-03-16
Method	X-RAY DIFFRACTION 2.7 Å
Organisms	Pectobacterium atrosepticum SCRI1043
Macromolecule	CRISPR-associated endonuclease Cas1 (protein)

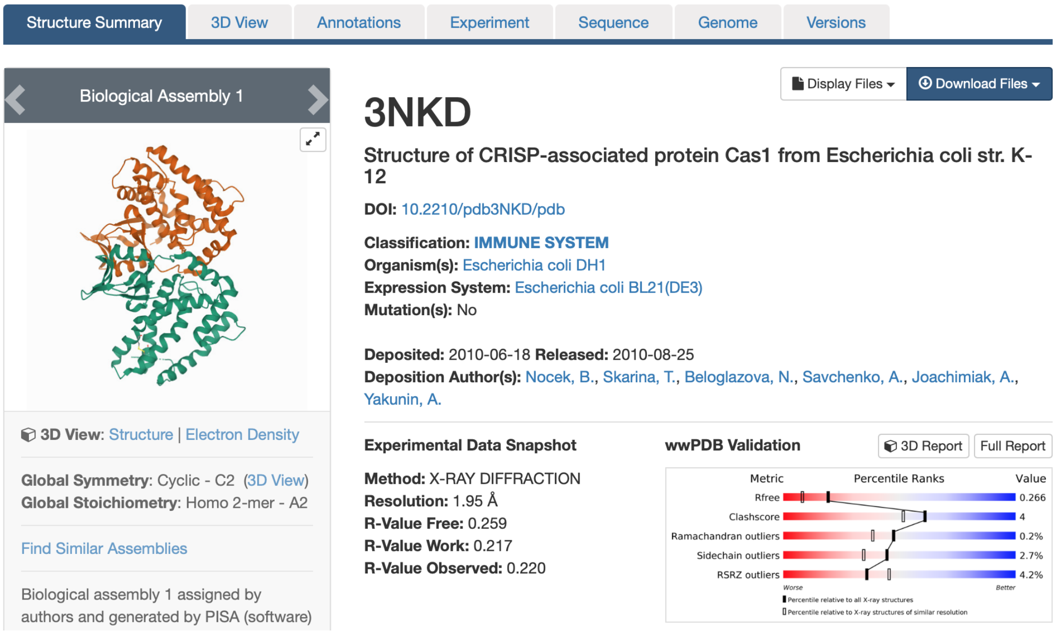
3D View

5XVN

Download File | View File

4. You can click on any of the structures and briefly explore its web page.

5. Let's analyze the PDB entry **3NKD** further here.



Structure Summary | 3D View | Annotations | Experiment | Sequence | Genome | Versions

Biological Assembly 1

Display Files ▾ | Download Files ▾

3NKD

Structure of CRISP-associated protein Cas1 from Escherichia coli str. K-12

DOI: 10.2210/pdb3NKD/pdb

Classification: **IMMUNE SYSTEM**
Organism(s): Escherichia coli DH1
Expression System: Escherichia coli BL21(DE3)
Mutation(s): No

Deposited: 2010-06-18 **Released:** 2010-08-25
Deposition Author(s): Nocek, B., Skarina, T., Beloglazova, N., Savchenko, A., Joachimiak, A., Yakunin, A.

3D View: Structure | Electron Density

Global Symmetry: Cyclic - C2 (3D View)
Global Stoichiometry: Homo 2-mer - A2

Find Similar Assemblies

Biological assembly 1 assigned by authors and generated by PISA (software)

Experimental Data Snapshot

Method: X-RAY DIFFRACTION
Resolution: 1.95 Å
R-Value Free: 0.259
R-Value Work: 0.217
R-Value Observed: 0.220

wwPDB Validation

3D Report | Full Report

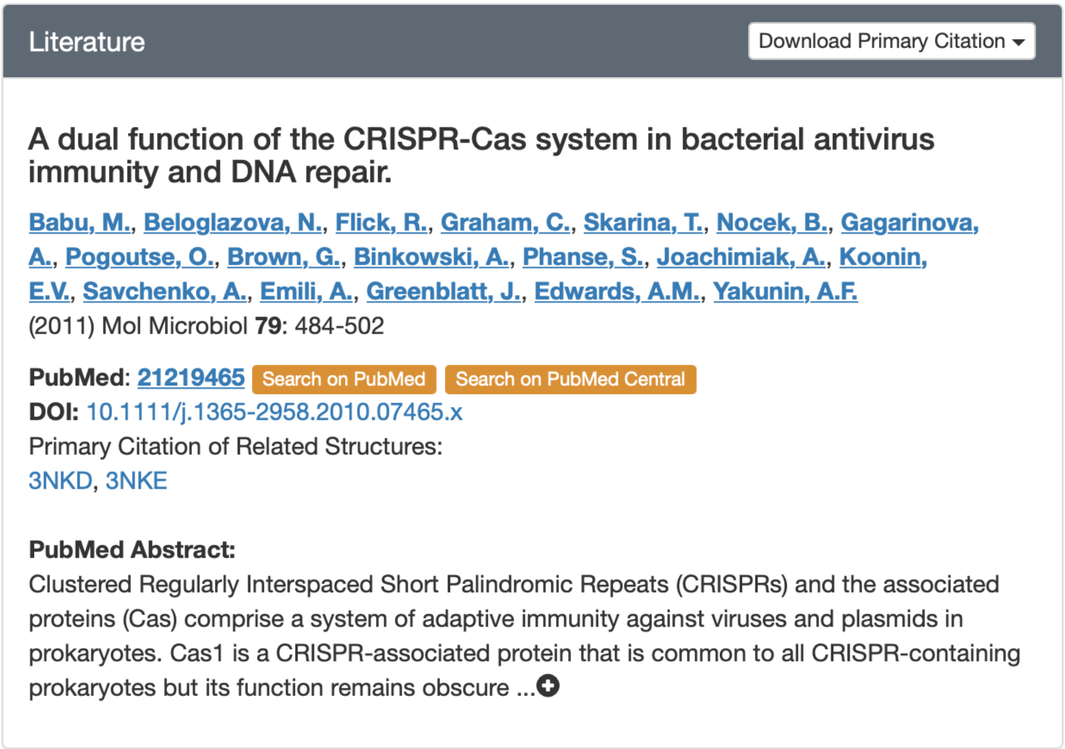
? What is the resolution of the structure?

(Answer: 1.95Å)

? **Does this structure belong to a wild-type protein or does it have mutated residues?**

(Answer: Wild-type, no mutations)

6. The details of the research article that has published this structure is given in the **Literature** section.



Literature

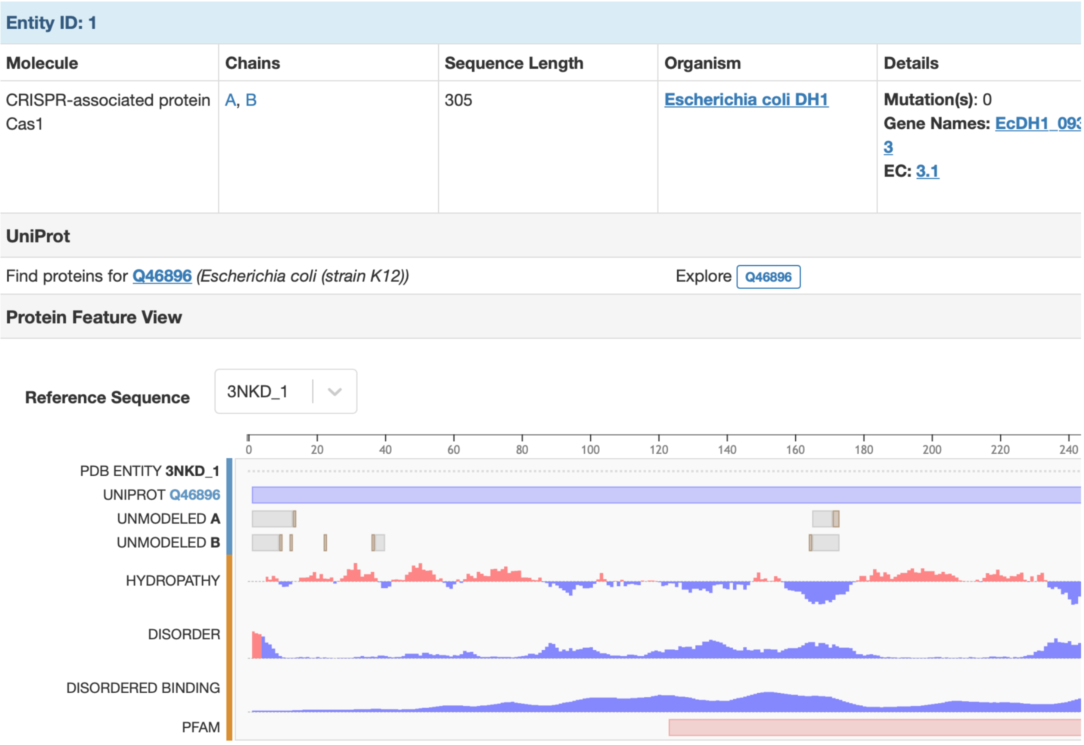
A dual function of the CRISPR-Cas system in bacterial antivirus immunity and DNA repair.

Babu, M., Beloglazova, N., Flick, R., Graham, C., Skarina, T., Nocek, B., Gagarinova, A., Pogoutse, O., Brown, G., Binkowski, A., Phanse, S., Joachimiak, A., Koonin, E.V., Savchenko, A., Emili, A., Greenblatt, J., Edwards, A.M., Yakunin, A.F.
(2011) Mol Microbiol **79**: 484-502

PubMed: 21219465
DOI: 10.1111/j.1365-2958.2010.07465.x
Primary Citation of Related Structures:
3NKD, 3NKE

PubMed Abstract:
Clustered Regularly Interspaced Short Palindromic Repeats (CRISPRs) and the associated proteins (Cas) comprise a system of adaptive immunity against viruses and plasmids in prokaryotes. Cas1 is a CRISPR-associated protein that is common to all CRISPR-containing prokaryotes but its function remains obscure ...

7. Residue-level secondary structural states and sequence annotations (mapped from UniProt) are provided in a graphical representation for an easy interpretation.



Entity ID: 1

Molecule	Chains	Sequence Length	Organism	Details
CRISPR-associated protein Cas1	A, B	305	Escherichia coli DH1	**Mutation(s):** 0 **Gene Names:** EcDH1_093 **EC:** 3.1

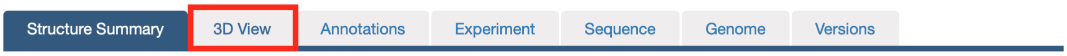
UniProt

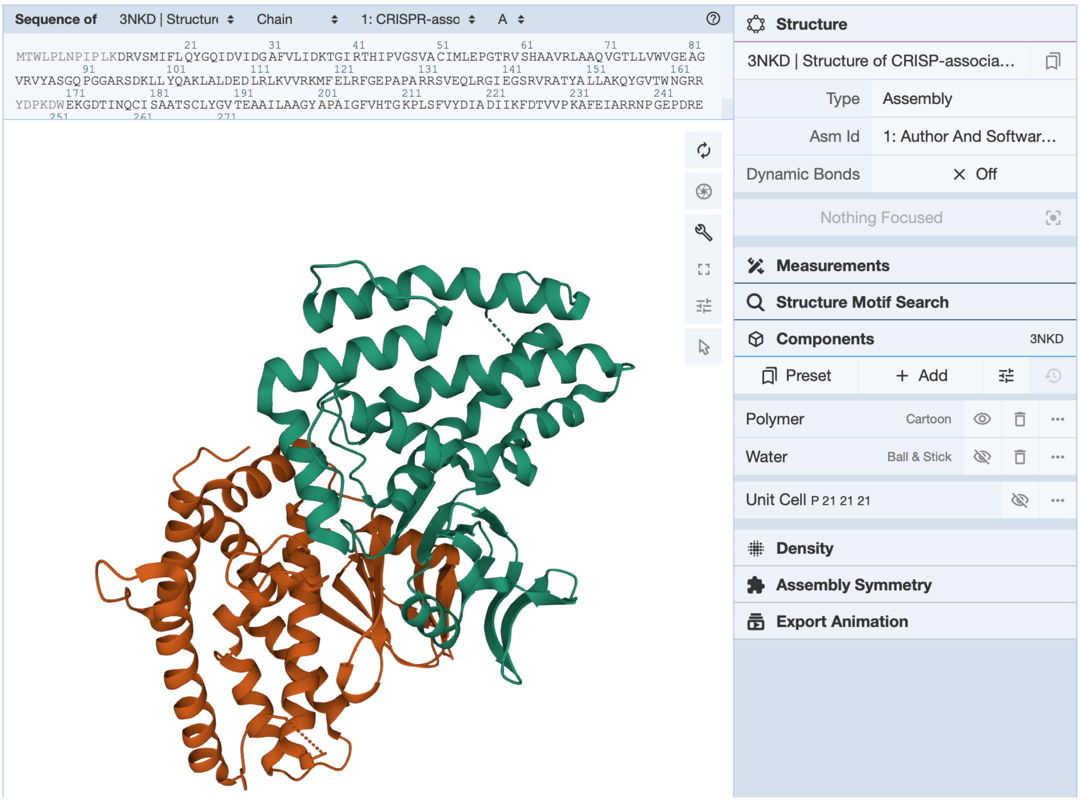
Find proteins for Q46896 *(Escherichia coli (strain K12))* Explore Q46896

Protein Feature View

Reference Sequence 3NKD_1

8. Go to **3D view**.





Sequence of 3NKD | Structure Chain 1: CRISPR-asso A

```
                     21        31        41        51        61        71        81
MTWLPLNPIPLKDRVSMIFLQYGQIDVIDGAFVLIDKTGIRTHIPVGSVACIMLEPGTRVSHAAVRLAAQVGTLLVWVGEAG
         91       101       111       121       131       141       151       161
VRVYASGQPGGARSDKLLYQAKLALDEDLRLKVVRKMFELRFGEPAPARRSVEQLRGIEGSRVRATYALLAKQYGVTWNGRR
      171       181       191       201       211       221       231       241
YDPKDWEKGDTINQCISAATSCLYGVTEAAILAAGYAPAIGFVHTGKPLSFVYDIADIIKFDTVVPKAFEIARRNPGEPDRE
    251       261       271
```

Structure

3NKD | Structure of CRISP-associa...

Type	Assembly
Asm Id	1: Author And Softwar...
Dynamic Bonds	× Off

Nothing Focused

Measurements

Structure Motif Search

Components 3NKD

Preset + Add

Polymer	Cartoon	
Water	Ball & Stick	
Unit Cell P 21 21 21		

Density

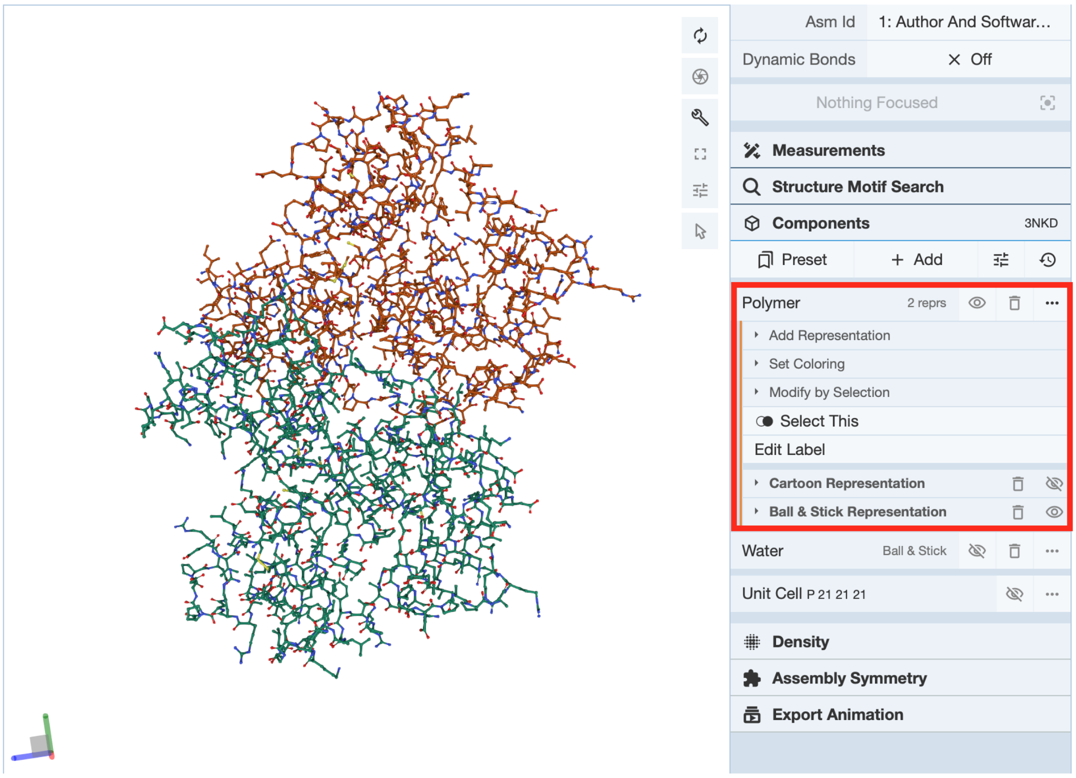
Assembly Symmetry

Export Animation

? Why do we see two colors in the cartoon view?

(Answer: It is a homodimer)

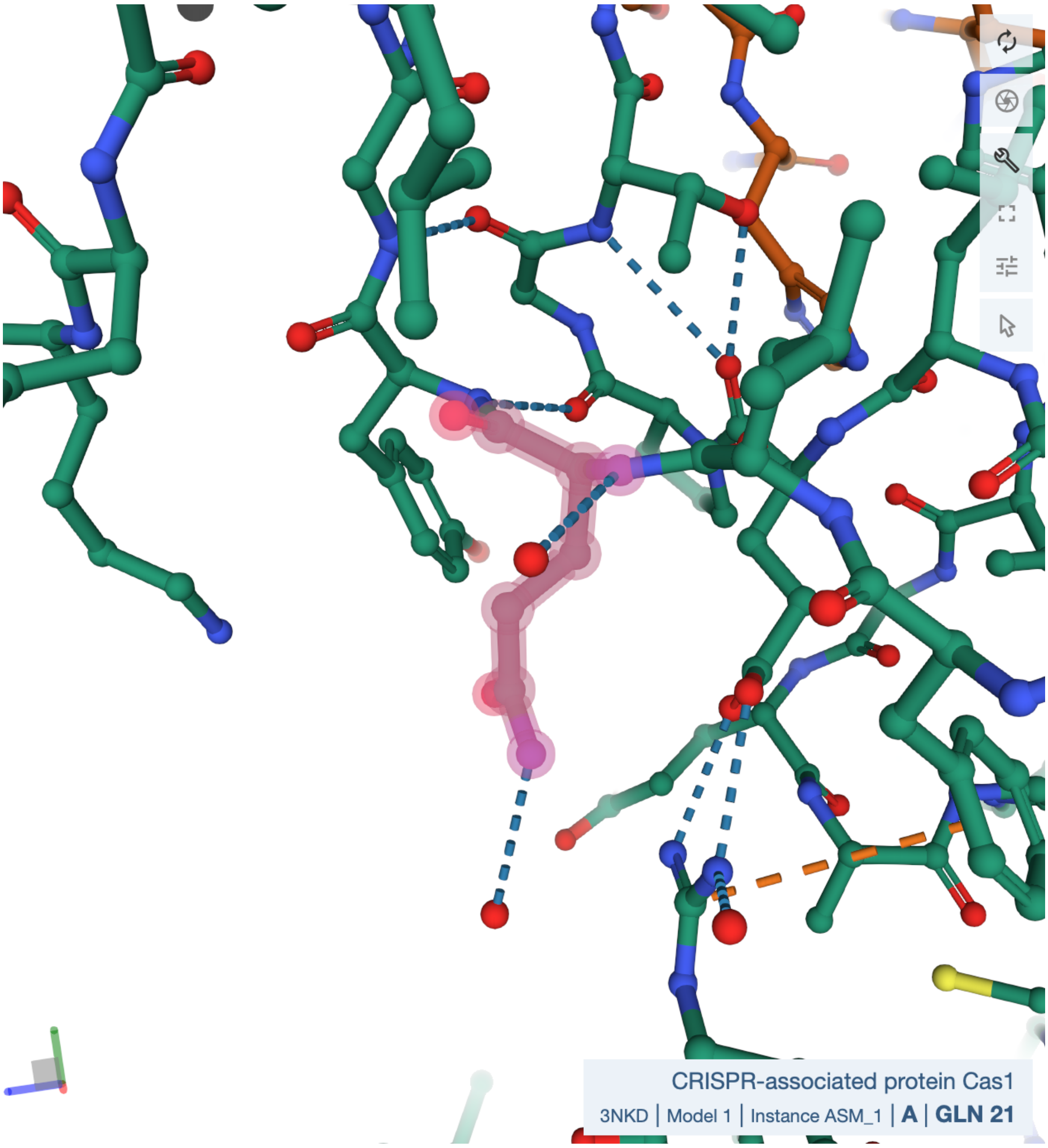
9. Have fun with adding different representation in the **Polymer** drop-down menu. Click on the **Add representation** to view multiple representation options. Shown below is the Ball & Stick representation.



Asm Id 1: Author And Softwar...

Dynamic Bonds Off

10. Select residue **Q21** in the sequence shown at the top panel. The cartoon automatically focuses on the local region around this residue. Interactions between Q21 and other residues are shown by dashed lines.



CRISPR-associated protein Cas1

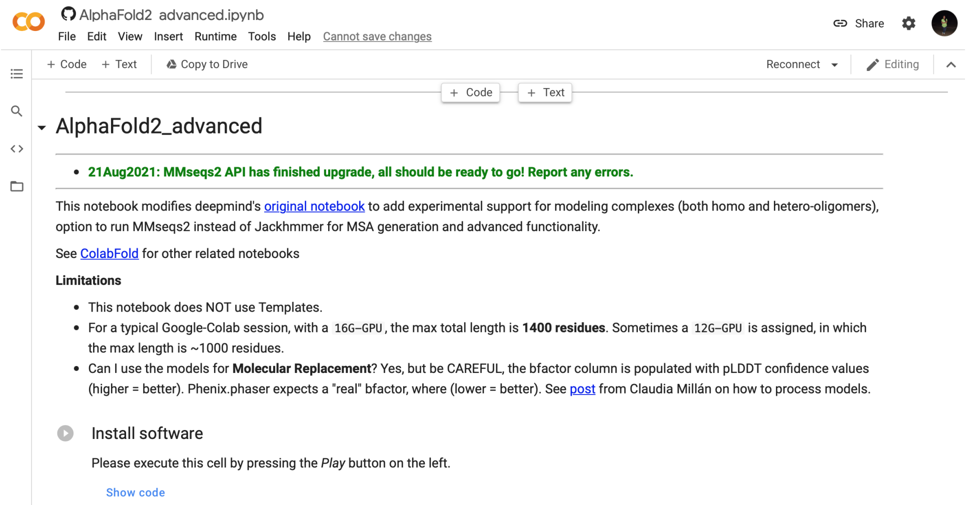
3NKD | Model 1 | Instance ASM_1 | A | GLN 21

11. If you want to explore more sophisticated tools for protein structure visualization and analysis, have a look at **Pymol, Chimera(X) or VMD**. They are GUI-based (graphical user interface) tools and offers several options to examine single as well as multiple protein structures.

4.5 Predict structure for Cas1-Cas2 protein complex using AlphaFold2_advanced (optional)

In general, proteins interact with other biomolecules and perform their functions. Likewise, Cas1 interacts with Cas2 to form a complex. The Cas1-Cas2 complex functions to integrate phage DNA into the CRISPR repeat of host bacterial viral genome.

1. Go to AlphaFold2_advanced. `https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/beta/AlphaFold2_advanced.ipynb`


AlphaFold2 advanced.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
Reconnect
Editing
+ Code + Text

AlphaFold2_advanced

- **21Aug2021: MMseqs2 API has finished upgrade, all should be ready to go! Report any errors.**

This notebook modifies deepmind's original notebook to add experimental support for modeling complexes (both homo and hetero-oligomers), option to run MMseqs2 instead of Jackhmmer for MSA generation and advanced functionality.

See ColabFold for other related notebooks

Limitations

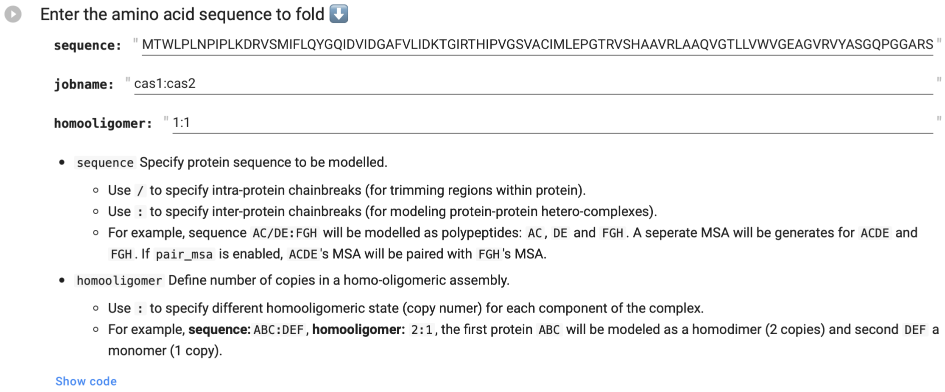
- This notebook does NOT use Templates.
- For a typical Google-Colab session, with a `16G-GPU`, the max total length is **1400 residues**. Sometimes a `12G-GPU` is assigned, in which the max length is ~1000 residues.
- Can I use the models for **Molecular Replacement**? Yes, but be CAREFUL, the bfactor column is populated with pLDDT confidence values (higher = better). Phenix.phaser expects a "real" bfactor, where (lower = better). See post from Claudia Millán on how to process models.

Install software

Please execute this cell by pressing the *Play* button on the left.

Show code

2. Paste amino acid sequences of Cas1 and Cas2 proteins separated by ';' in the input. Sequence information can be found in Cas1_Q46896.fasta and Cas2_P45956.fasta files.



Enter the amino acid sequence to fold ⬇️

sequence: " MTWLPLNPIPLKDRVSMIFLQYGQIDVIDGAFVLIDKTGIRTHIPVGSVACIMLEPGTRVSHAAVRLAAQVGTLLVWVGEAGVRVYASGQPGGARS "

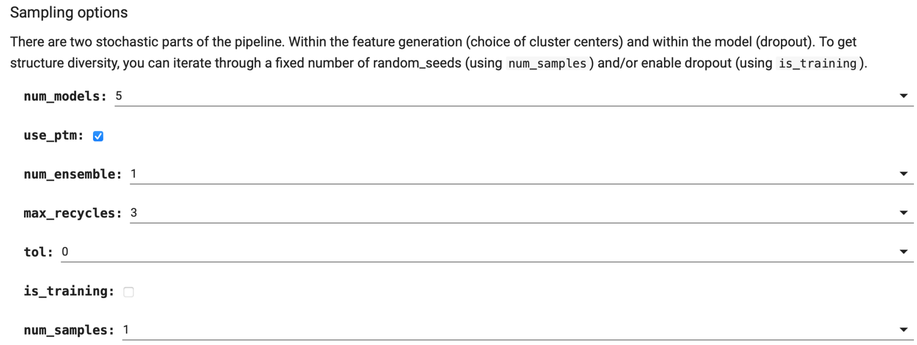
jobname: " cas1:cas2 "

homooligomer: " 1:1 "

- `sequence` Specify protein sequence to be modelled.
    - Use `/` to specify intra-protein chainbreaks (for trimming regions within protein).
    - Use `:` to specify inter-protein chainbreaks (for modeling protein-protein hetero-complexes).
    - For example, sequence `AC/DE:FGH` will be modelled as polypeptides: `AC`, `DE` and `FGH`. A seperate MSA will be generates for `ACDE` and `FGH`. If `pair_msa` is enabled, `ACDE`'s MSA will be paired with `FGH`'s MSA.
- `homooligomer` Define number of copies in a homo-oligomeric assembly.
    - Use `:` to specify different homooligomeric state (copy numer) for each component of the complex.
    - For example, **sequence:** `ABC:DEF`, **homooligomer:** `2:1`, the first protein `ABC` will be modeled as a homodimer (2 copies) and second `DEF` a monomer (1 copy).

Show code

3. Go with the default settings for `num_models`, `num_ensemble`, `max_recycles`, etc. Set `msa_method` to MMseq2 and `pair_mode` to unpaired+paired.



Sampling options

There are two stochastic parts of the pipeline. Within the feature generation (choice of cluster centers) and within the model (dropout). To get structure diversity, you can iterate through a fixed number of random_seeds (using `num_samples`) and/or enable dropout (using `is_training`).

num_models: 5

use_ptm: ☑

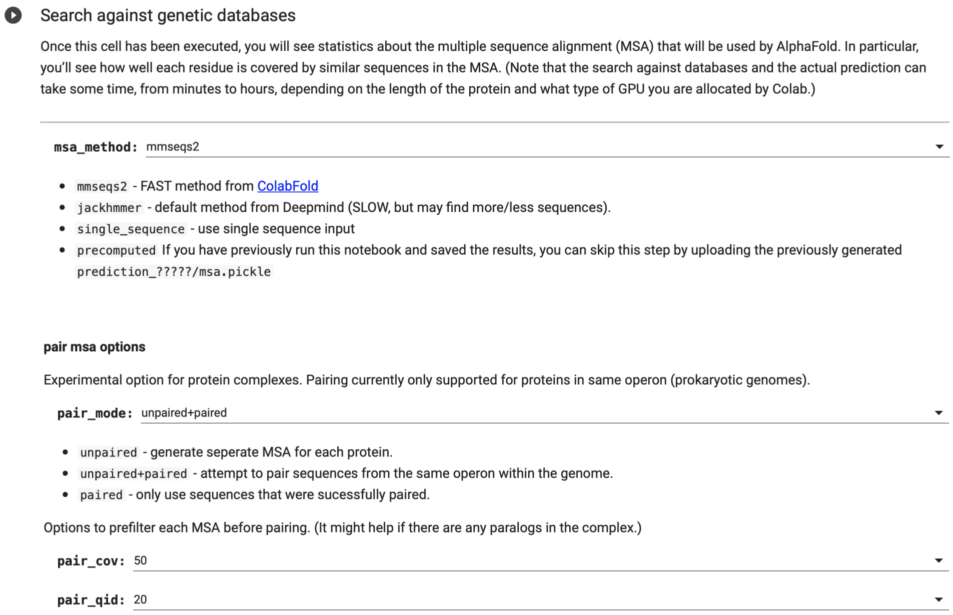
num_ensemble: 1

max_recycles: 3

tol: 0

is_training: □

num_samples: 1



Search against genetic databases

Once this cell has been executed, you will see statistics about the multiple sequence alignment (MSA) that will be used by AlphaFold. In particular, you'll see how well each residue is covered by similar sequences in the MSA. (Note that the search against databases and the actual prediction can take some time, from minutes to hours, depending on the length of the protein and what type of GPU you are allocated by Colab.)

msa_method: mmseqs2

- `mmseqs2` - FAST method from ColabFold
- `jackhmmer` - default method from Deepmind (SLOW, but may find more/less sequences).
- `single_sequence` - use single sequence input
- `precomputed` If you have previously run this notebook and saved the results, you can skip this step by uploading the previously generated `prediction_?????/msa.pickle`

pair msa options

Experimental option for protein complexes. Pairing currently only supported for proteins in same operon (prokaryotic genomes).

pair_mode: unpaired+paired

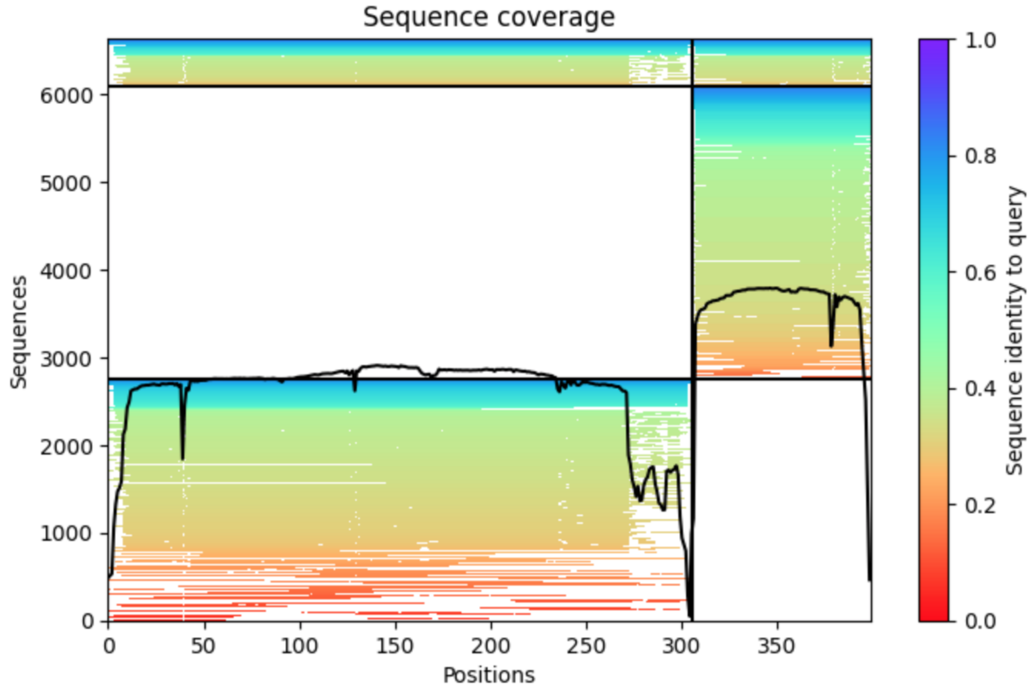
- `unpaired` - generate seperate MSA for each protein.
- `unpaired+paired` - attempt to pair sequences from the same operon within the genome.
- `paired` - only use sequences that were sucessfully paired.

Options to prefilter each MSA before pairing. (It might help if there are any paralogs in the complex.)

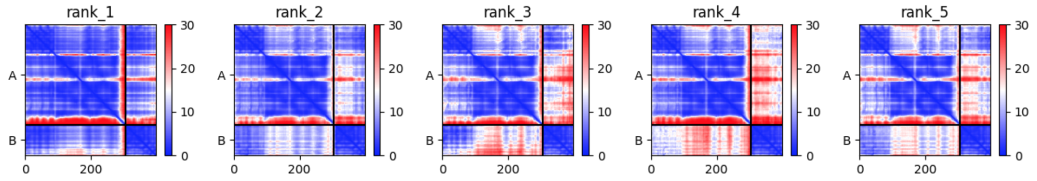
pair_cov: 50

pair_qid: 20

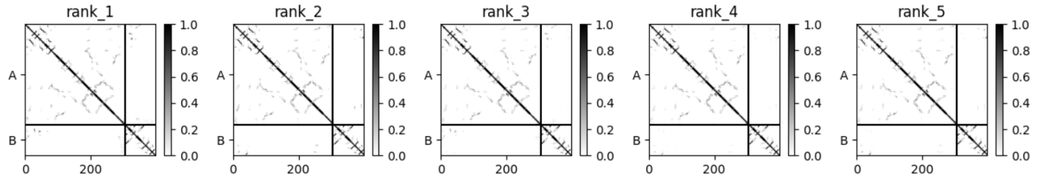
4. Similar to ColabFold, predicted results are given in the form of sequence coverage, confidence of the model and per-residue lDDT measures.



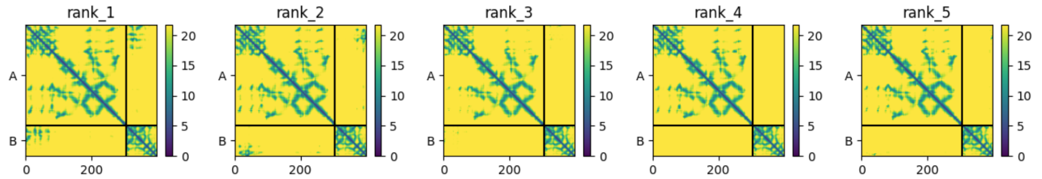
predicted alignment error



predicted contacts

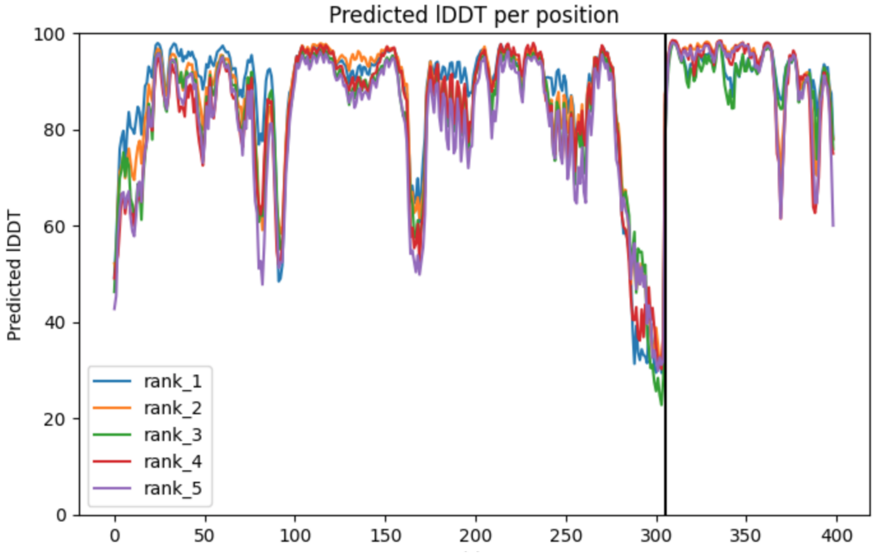


predicted distogram



5. Per-residue lDDT is provided in the line plot. As you see, 5 models have similar predicted lDDT profiles and in some regions they vary which are in general loops or unstructured.

```
predicted LDDT
```



6. We can choose the best model for the further analysis based on the model confidence and per-residue lDDT score.

Appendix

5.1 Some useful Bash commands

```
# show a file inside the terminal (hint: use q to exit again)
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

5.3 Exercise solutions for section 1.4

1.
   ```bash
   #!/bin/bash
   echo "Hello Bash"
   ```

2.
   ```bash
   #!/bin/bash
   AGE = 30
   if [ $AGE -ge 18 ]; then
           echo "Here is your beer"
   fi
   ```

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 Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, 35(11):1026–1028, 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] Alex Bateman. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, 47(D1):D506–D515, 2019.

[5] Brian D Ondov, Nicholas H Bergman, and Adam M Phillippy. Interactive metagenomic visualization in a Web browser. *BMC Bioinform.*, 12(1):385, 2011.

[6] Martin Steinegger, Milot Mirdita, and Johannes Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *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. *Clin. Infect. Dis.*, 66(5):789–792, 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 Söding. Clustering huge protein sequence sets in linear time. *Nat. Commun.*, 9(1):2542, 2018.

[10] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, 9(2):173–175, 2012.

[11] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2 - Approximately maximum-likelihood trees for large alignments. *PLoS One*, 5(3), 2010.

[12] Andrei Kouranov. The RCSB PDB information portal for structural genomics. *Nucleic Acids Res.*, 34(D1):D302–D305, 2006.

[13] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold - making protein folding accessible to all. *bioRxiv*, 2021.