UNIVERSITY OF TORONTO

An Introduction to Foundation Models for Science

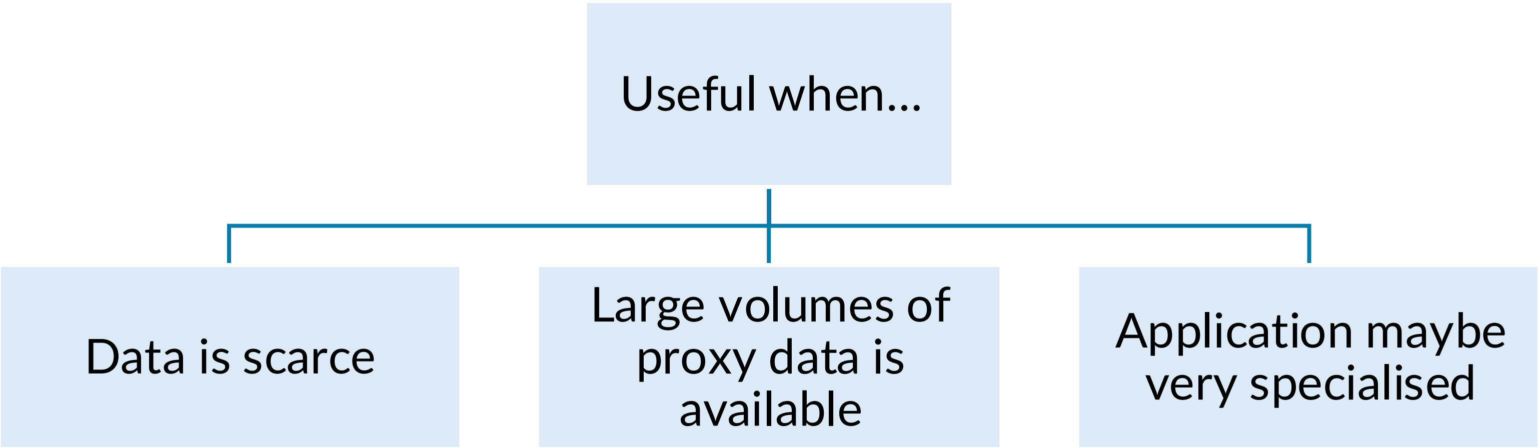
Tutorial 1: Background and introduction

Schmidt Sciences

FOUNDATION MODELS for SCIENCE

What is a Foundation Model?

"Foundation models are artificial intelligence (AI) models **trained on vast, immense datasets** and can fulfill a **broad range of general tasks**. They serve as the **base or building blocks** for crafting more specialized applications"

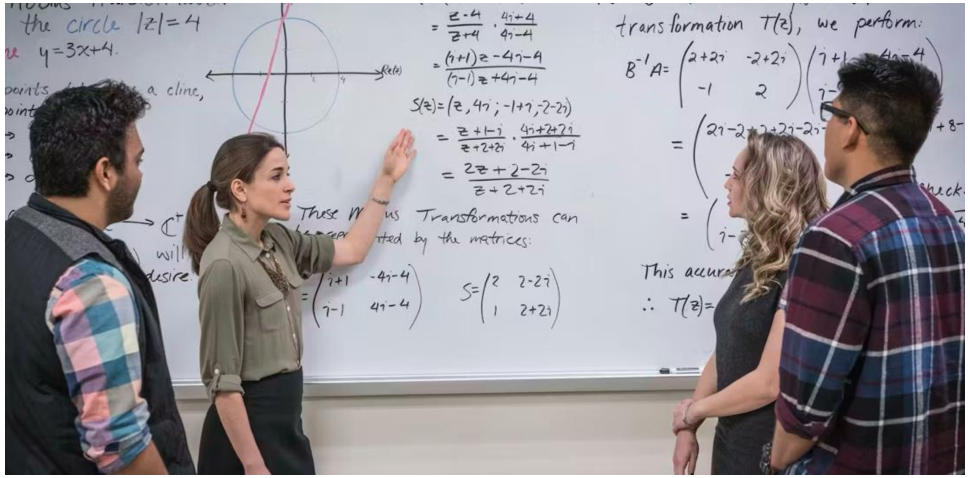

Useful when…

Data is scarce

Large volumes of proxy data is available

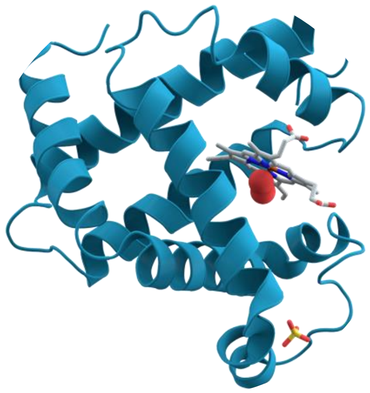
Application maybe very specialised

vs

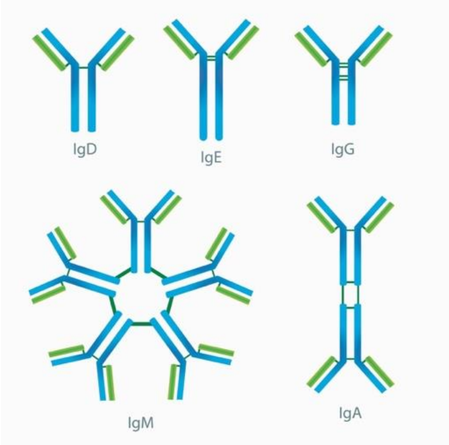


Rina Diane Caballar, IBM (https://www.ibm.com/think/topics/foundation-models#:~:text=Author,models%20are%20built%20via%20adaptation.)

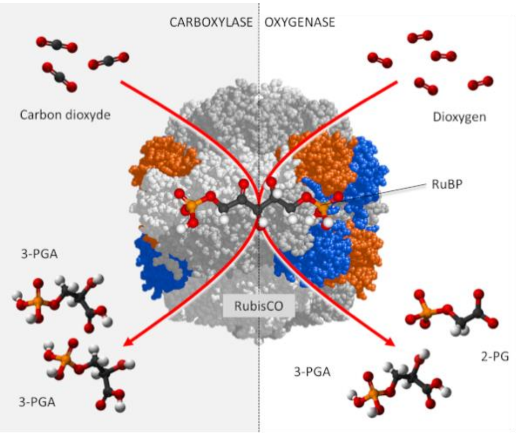
We will use protein foundation models to illustrate the usage of FMs in protein engineering problems



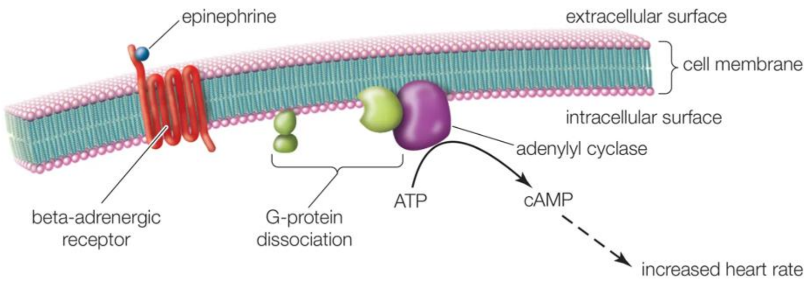
Myoglobin
(storage)



Immunoglobulin



Rubisco
(catalysis)



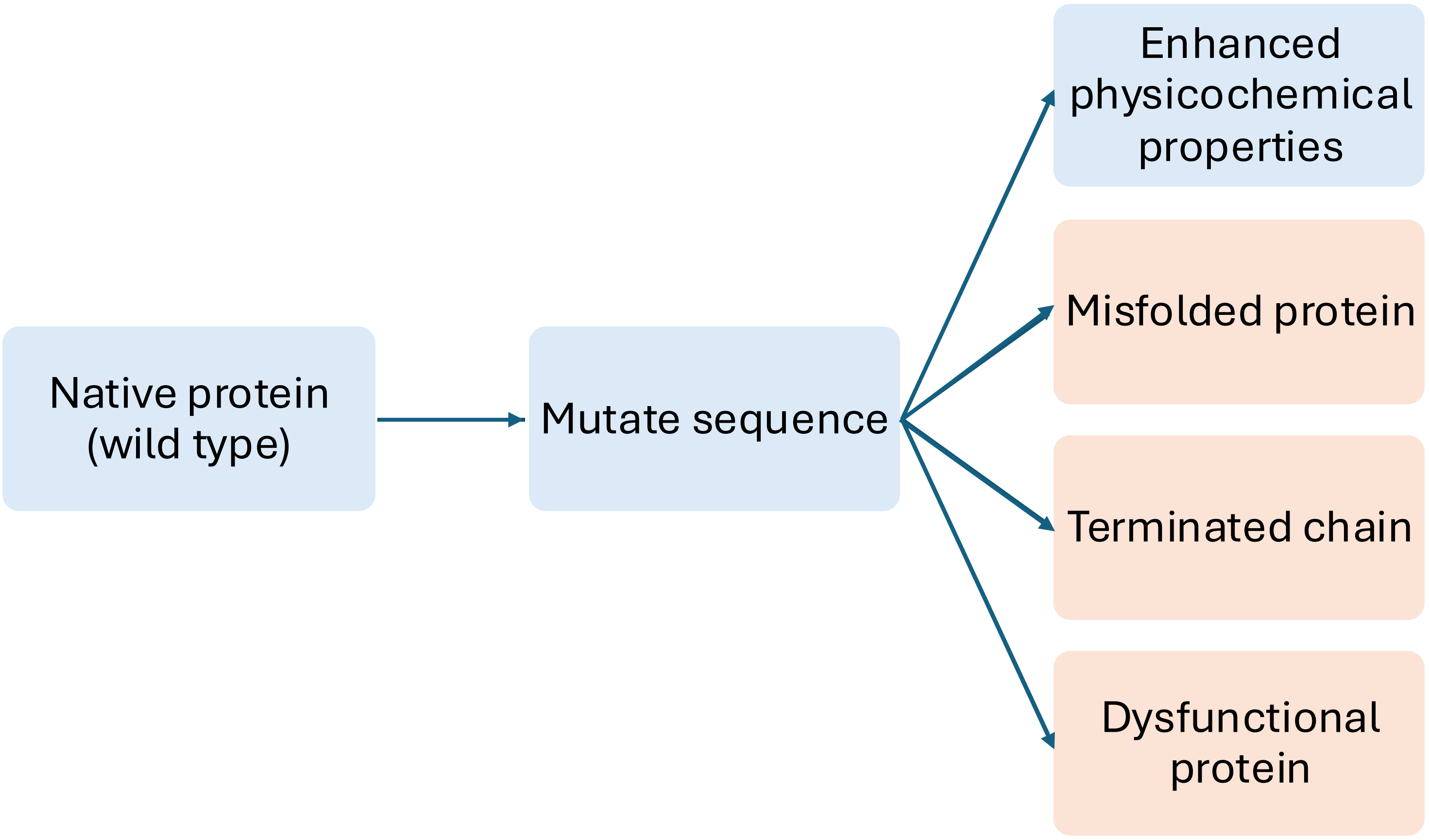
G-protein coupled receptors
(signaling)

Proteins are also **industrially valuable**:
Therapeutic agents
Brewing enzymes
Food preservants
Industrial catalysts

But wild type proteins do not function well outside their native conditions

Notin P et al., NeurIPS (2023) ; Pandi SV & Ransumdar B, (2024) ; A Bjerregaard et al., Current Opinion in Structural Biology **91**, 103004 (2025)

Wild-type proteins need to be modified or ‘mutated’ to adapt their function to the application we want



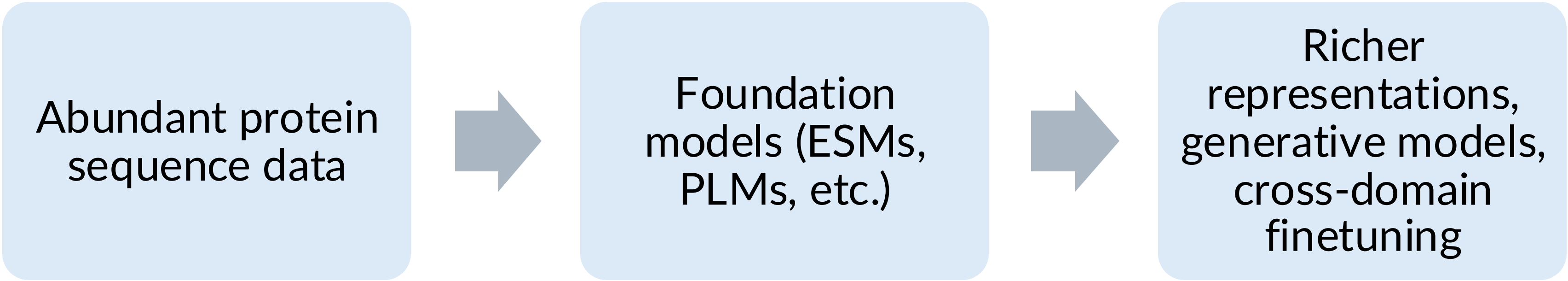
Iterative cycles of mutation, screening and selection are time and labour-intensive

Computational models like Rosetta and FoldX can predict protein stability fairly well

But fall short in predicting biological activity…

Y Xia et al., Applied Microbiology and Biotechnology **105** (19), 7309 (2021) ; S Lutz and SM Iamurri, in *Protein Engineering: Methods and Protocols*, edited by Uwe T. Bornscheuer and Matthias Höhne (Springer New York, New York, NY, 2018), pp. 1.

AI and deep learning models perform better at relating protein sequence to activity and physiochemical properties



Abundant protein sequence data

Foundation models (ESMs, PLMs, etc.)

Richer representations, generative models, cross-domain finetuning

When you have a pre-trained FM, you can then use it for downstream tasks as is, or use transfer learning or fine tuning to refine it further

Notin P et al., NeurIPS (2023) ; Pandi SV & Ransumdar B, (2024) ; A Bjerregaard et al., Current Opinion in Structural Biology **91**, 103004 (2025)

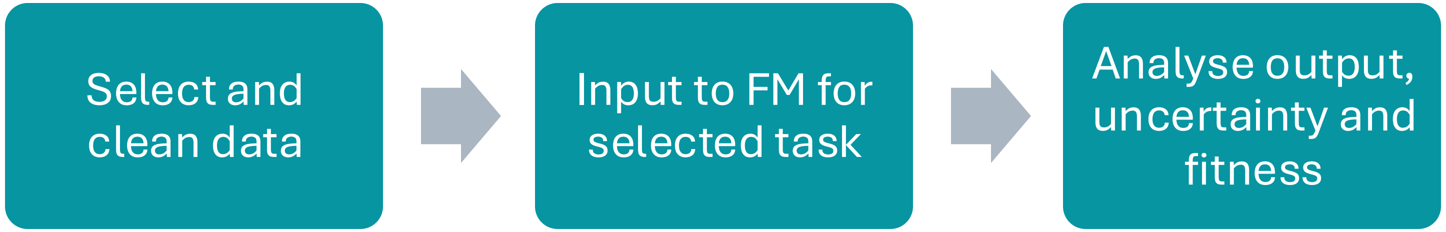
ESM-2 is a transformer protein language model that can predict the structure of a protein from its primary sequence

Masked language model

Trained on 65M unique protein sequences from UniRef DB

Model is trained to predict the identity of a random AA based on the rest of the sequence (context)

ESM-2 has ‘seen’ how protein sequence is related to structure



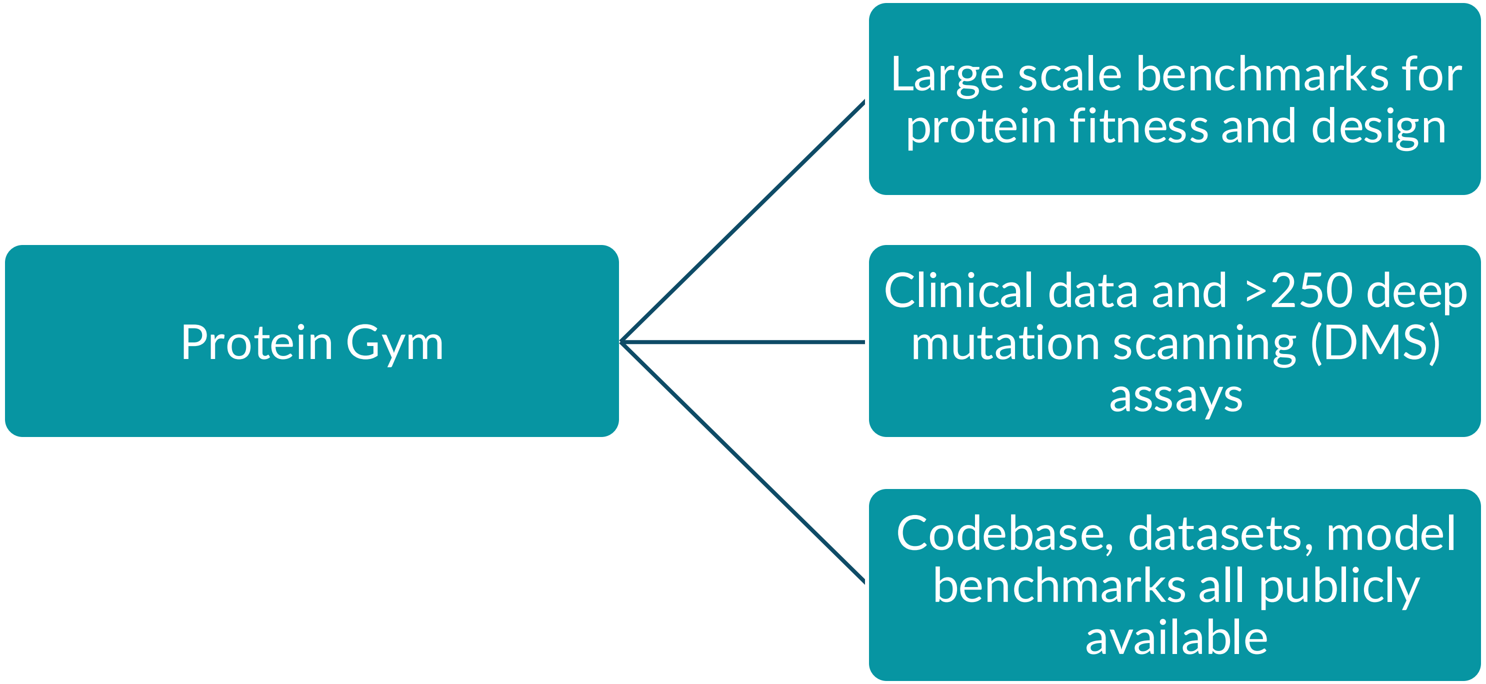
Select and clean data

Input to FM for selected task

Analyse output, uncertainty and fitness

Z Lin et al., Science **379** (6637), 1123 (2023) ; ESM: Evolutionary Scale language Model

Data from public databases need to be pre-processed before they can be input to your model for the most meaningful modelling



Protein Gym

Large scale benchmarks for protein fitness and design

Clinical data and >250 deep mutation scanning (DMS) assays

Codebase, datasets, model benchmarks all publicly available

Data can be found at: https://proteingym.org/download → DMS Assays → substitutions

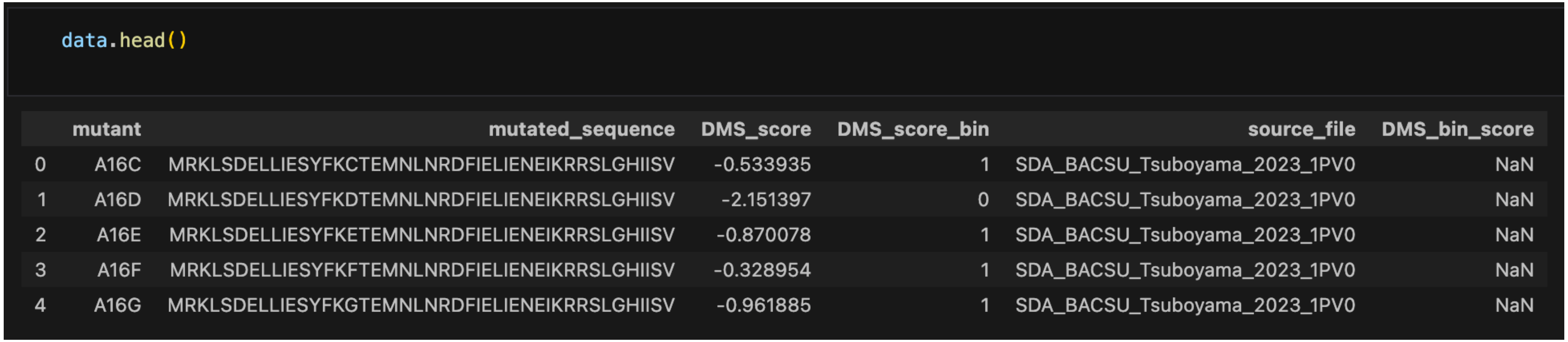
Each CSV = one assay from a substitution type mutation

DMS Score is a measure of the impact of mutation on the protein fitness, function, or stability.

Higher score = higher impact
-ve score = -ve impact
+ve score = +ve impact

Notin P et al., NeurIPS (2023)

Probing the raw data and domain knowledge allows us to decide the best steps for data cleaning and standardization



```
data.head()
```

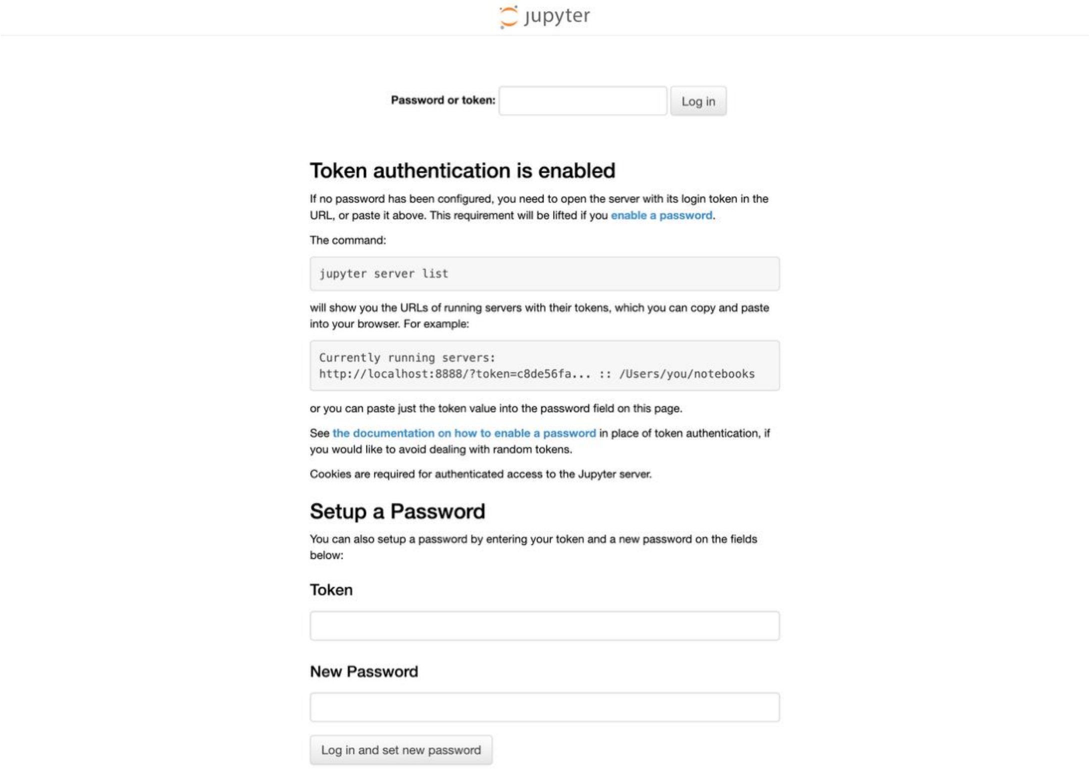
	mutant	mutated_sequence	DMS_score	DMS_score_bin	source_file	DMS_bin_score
0	A16C	MRKLSDELLIESYFKCTEMNLNRDFIELIENEIKRRSLGHIISV	-0.533935	1	SDA_BACSU_Tsuboyama_2023_1PV0	NaN
1	A16D	MRKLSDELLIESYFKDTEMNLNRDFIELIENEIKRRSLGHIISV	-2.151397	0	SDA_BACSU_Tsuboyama_2023_1PV0	NaN
2	A16E	MRKLSDELLIESYFKETEMNLNRDFIELIENEIKRRSLGHIISV	-0.870078	1	SDA_BACSU_Tsuboyama_2023_1PV0	NaN
3	A16F	MRKLSDELLIESYFKFTEMNLNRDFIELIENEIKRRSLGHIISV	-0.328954	1	SDA_BACSU_Tsuboyama_2023_1PV0	NaN
4	A16G	MRKLSDELLIESYFKGTEMNLNRDFIELIENEIKRRSLGHIISV	-0.961885	1	SDA_BACSU_Tsuboyama_2023_1PV0	NaN

- What are the columns? What type of information is there?
- Can the data be used as is for a downstream task?
- [‘mutant’] contains information about the wild type, mutation position, and type of mutation. Can we extract this data?
- Do we have any duplicates or null entries?
- Do the DMS scores need to be scaled or standardized?

What is the impact of having nulls and duplicates in your data?

Connecting to the Server

- You will receive a link and password like this:
- Environment URL: https://<instance-name>.cloud.denvrdata.com/
- Password: Shared during the session
- Enter password into *password or token* box at top of screen



Token authentication is enabled

If no password has been configured, you need to open the server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

```
jupyter server list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens.

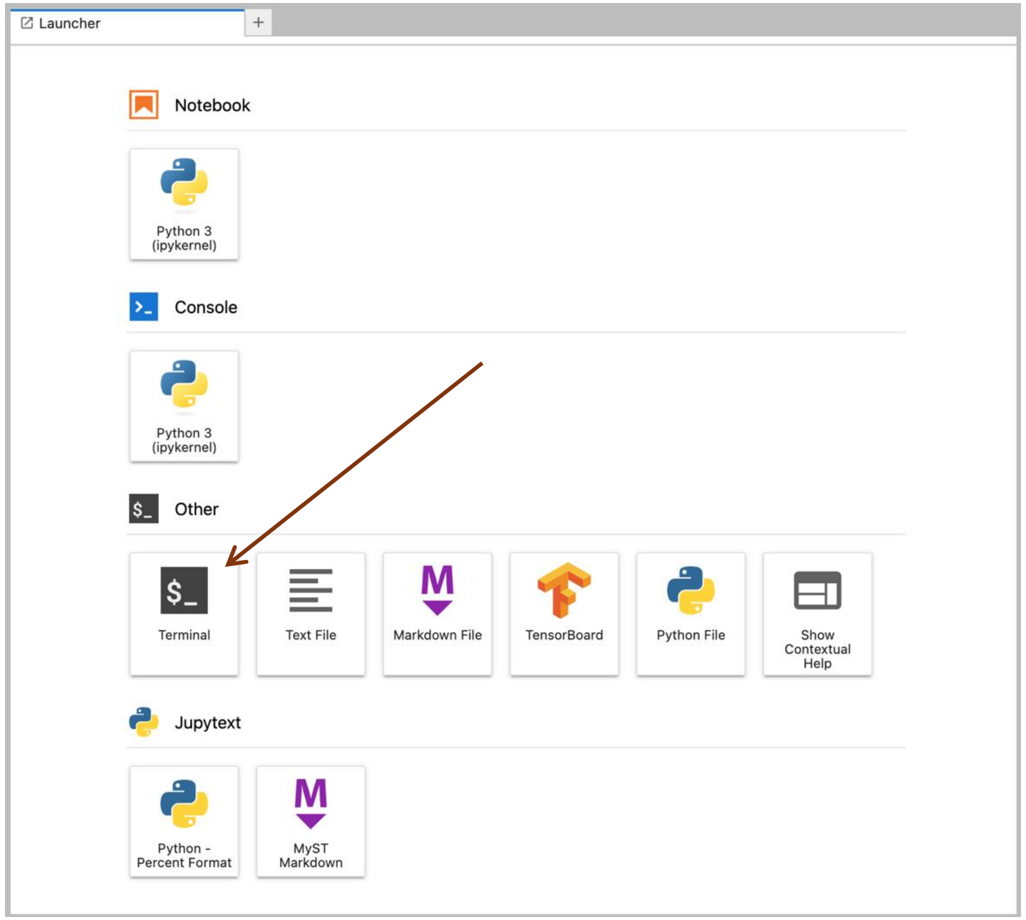
Cookies are required for authenticated access to the Jupyter server.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

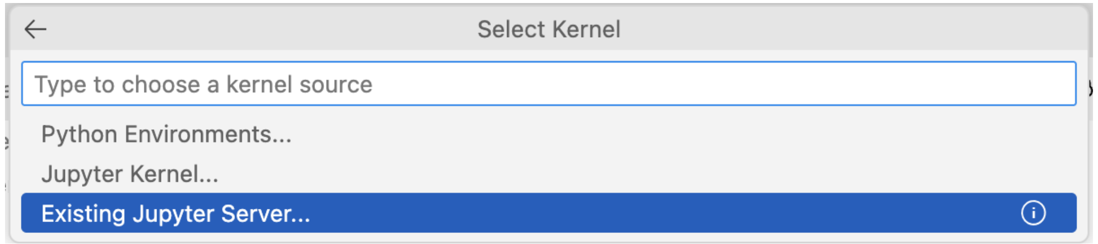
Forking repo

- Open new terminal window
- Create personal directory
- Clone git repo



Using VS Code

- Clone repository locally
- Connect to remote server using 'existing jupyter server' option



Data pre-processing is very domain and context specific. Having strong knowledge about the problem you are solving as well as your data helps!

- Handling empty/missing values
- Handling conflicting values
- Ensuring all units are uniform (especially if combining datasets)
- One-hot encoding/embedding
- Filtering out sequence length (if interested only in particular lengths)
- And many other steps!

- Remember to think of the problem you want to solve and the data you have at hand!

- Your data is now ready for the ML pipeline! It can be passed on to a foundation model like ESM or ProtBERT for your ML task.

FOUNDATION MODELS

for SCIENCE