High-Throughput Machine Learning and Modeling for Protein Function Landscapes

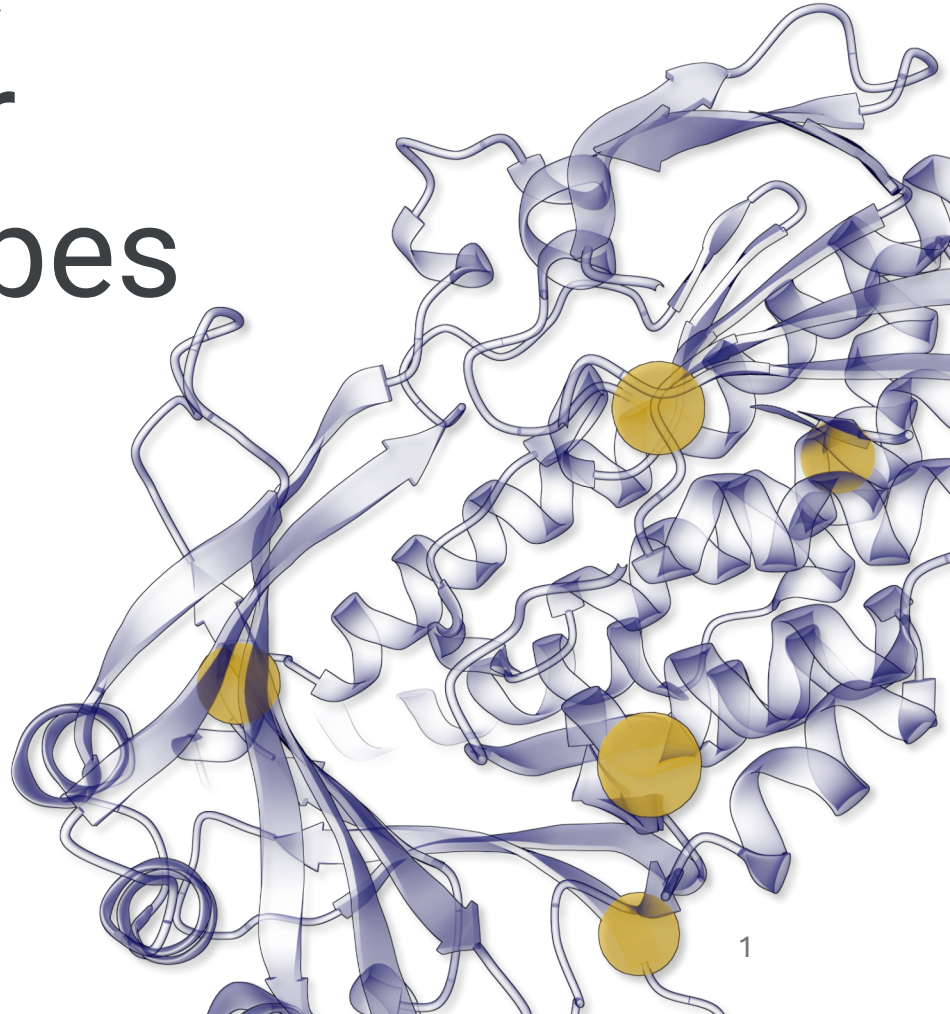
Azam Hussain

Brooks Group

University of Michigan

UofM DCMB Tools and Technology Seminar Series

Sep 19, 2024



Tools and Technology for **Protein Modelling**

- There are too many computational tools and techniques in protein modelling at different scales
- These tools tend to be used individually or in frameworks by domain experts
- The goal of this talk will be to make **general high throughput state of the art tools in protein modelling** beginner friendly
- Highlights of **code snippets** for each tool used in the **paper / github**
  - https://academic.oup.com/bioinformatics/article/40/1/btae002/7513688
  - https://github.com/BrooksResearchGroup-UM/seq_struct_func/tree/main

Hussain, A. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** (2024) doi:10.1093/bioinformatics/btae002.

Paper Background

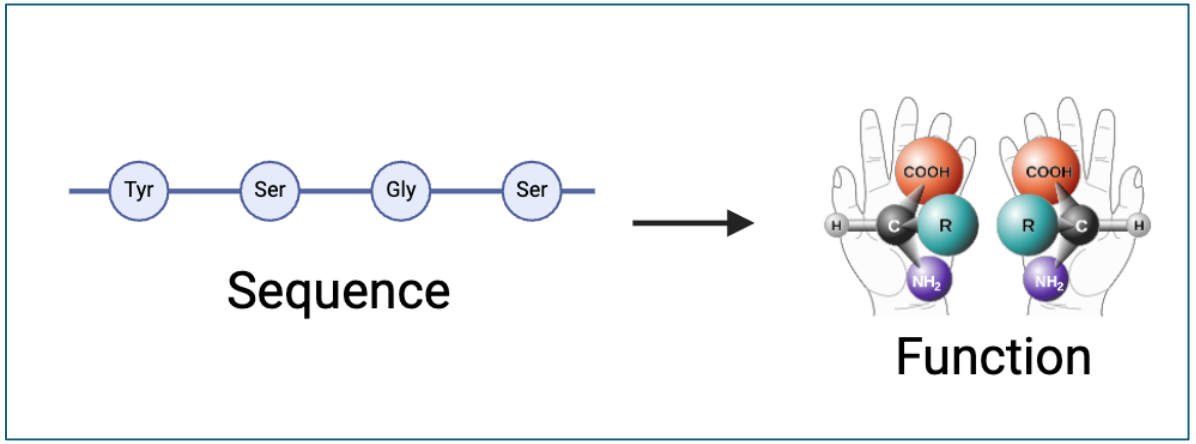
Hussain, **A**. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** **(2024)** doi:10.1093/bioinformatics/btae002.

Protein Sequence-Function Landscapes and the Curse of Dimensionality

- Any protein can be defined by:
  - **Sequence**: 100-1000 amino acids: (n choose 20)
  - **Structure**: 1000s of atoms ($x \in \mathbb{R}^{3n}$)
  - **Function**: 1000s+ classifications
- Traditional exploration of protein function entails:
  - Natural protein (sequence from DNA) (1 sequence)
  - Evolutionary analysis: 100-1000s of homologs
  - Crystal structure (1+ structures)
  - Molecular modelling (dynamics, docking, QM/MM)
  - Binding / functional assaying (cell lysate / purification)
  - Rational design / directed evolution (months)
- Can we do this exploration faster?

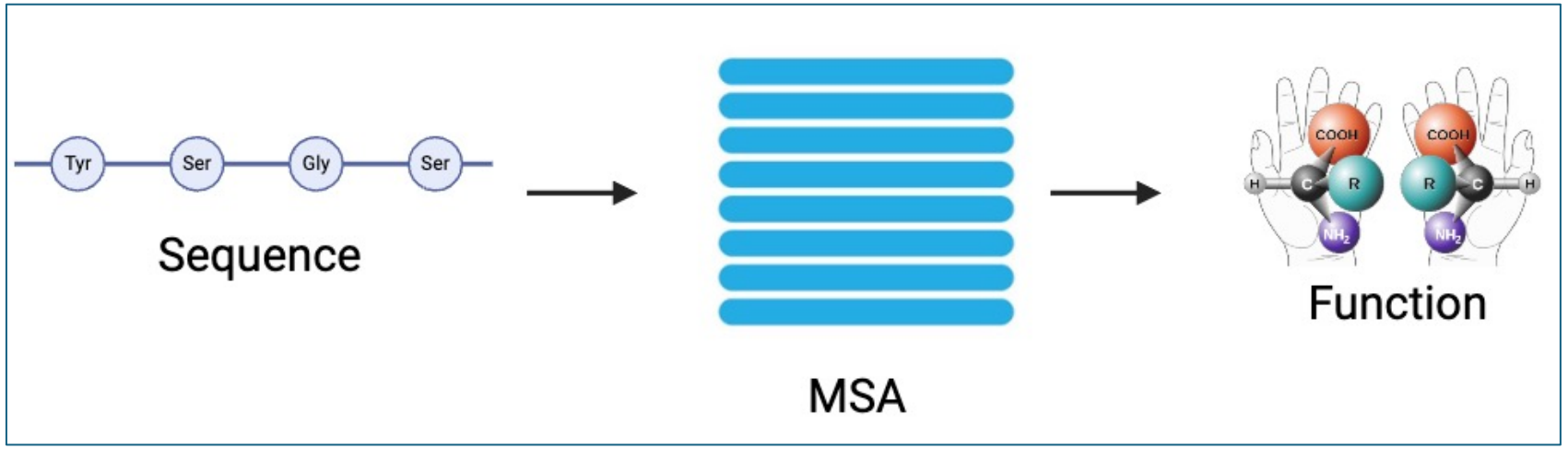
Designing a Workflow for Efficient Exploration

- Can we design a workflow that takes advantage of recent advancements to help protein modeling?
- If I was in a biocatalysis lab:
  - **Input**: **sequence** of biocatalyst (400 aa known enzyme)
  - **Output**: **mutations** to enhance function (enantioselectivity / activity)
  - Training data: 10-100 sequence-function points



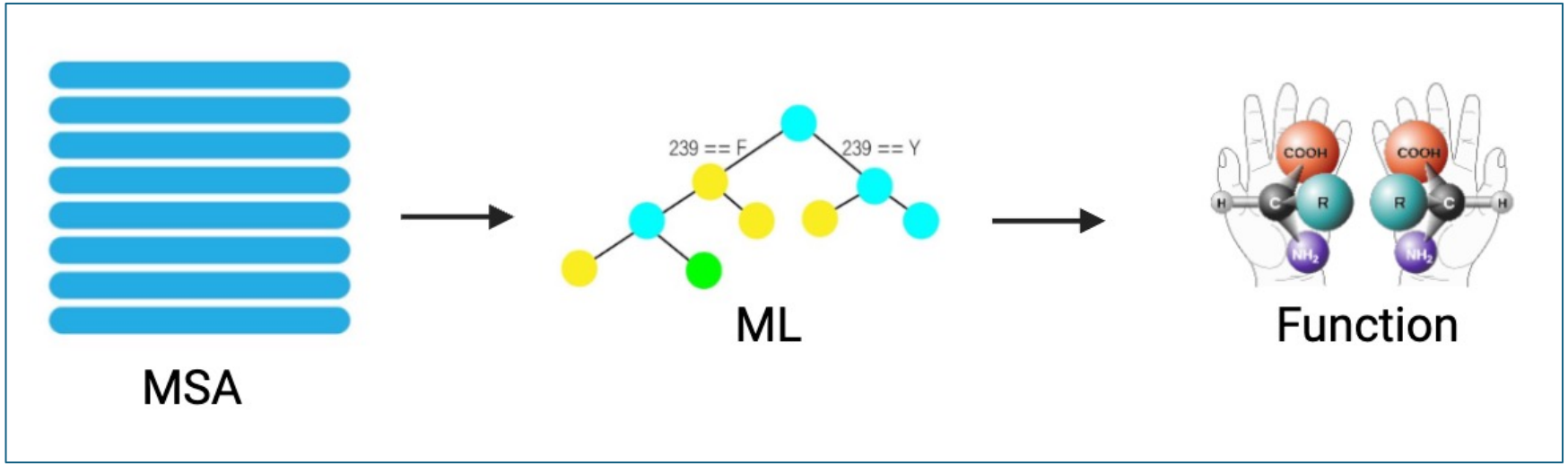
Adding an MSA for Evolutionary Constraints

- Sequence to function directly is impossible
- Let's add in the multiple sequence alignment built from a hypothetical BLAST search to restrict the sequence space



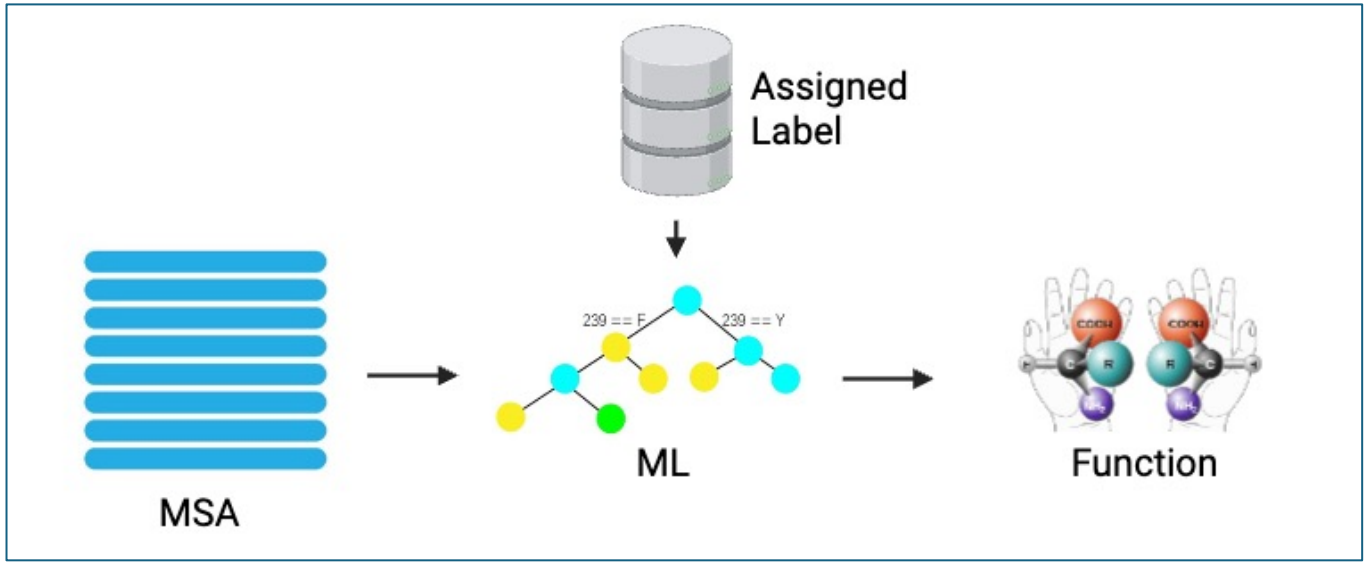
Train an ML model to map MSA to function

- Let's add a random forest model that takes in an aligned sequence and outputs the stereochemistry of that aligned sequence
- But can you train a model on only 10-100 experimental data points? Generalizable?



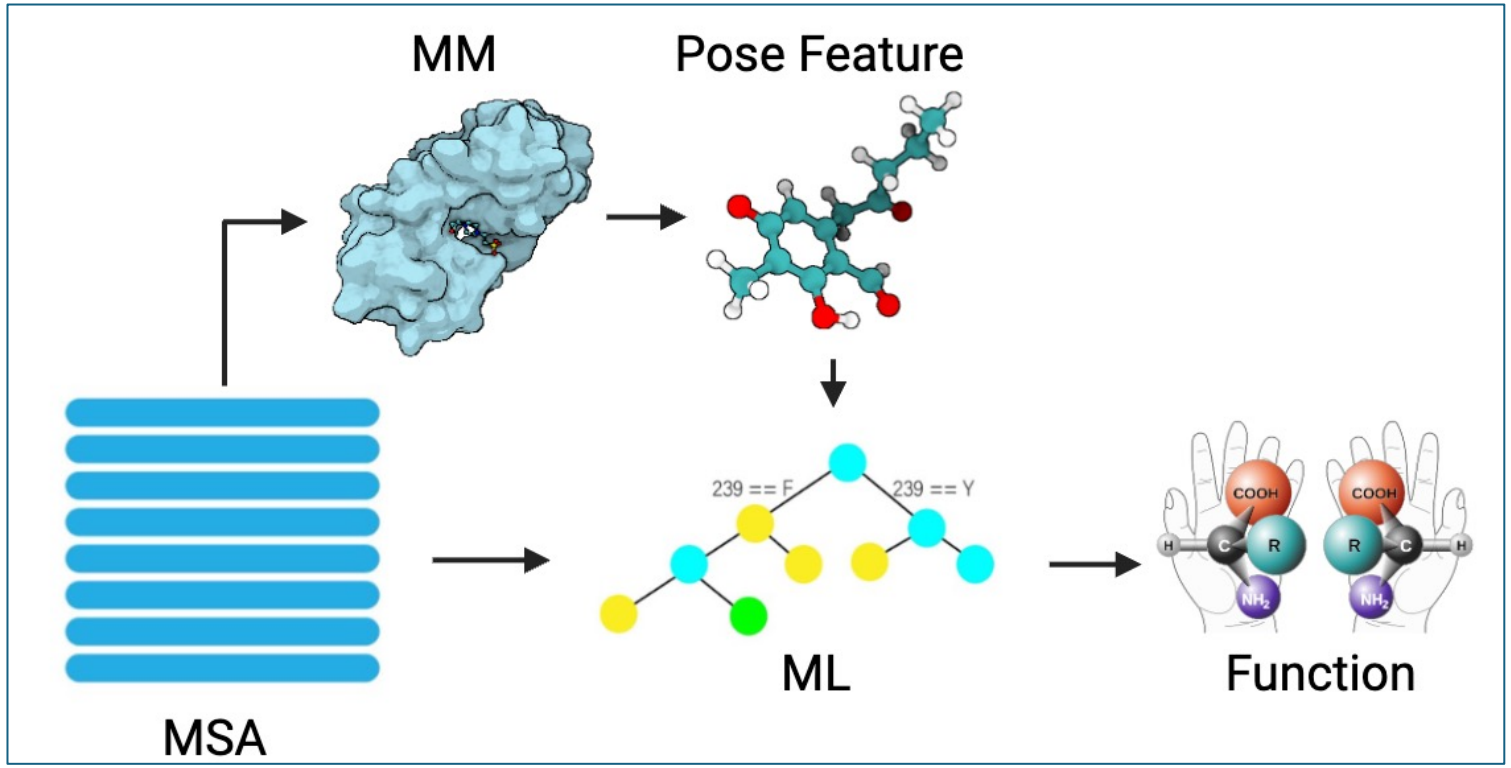
Generalized Training on Predicted Labels

- Let's train our random forest model with approximate assigned labels to the sequence
- The assigned label should correlate with the final function of the enzyme, and the ML model will fit labels to sequence space



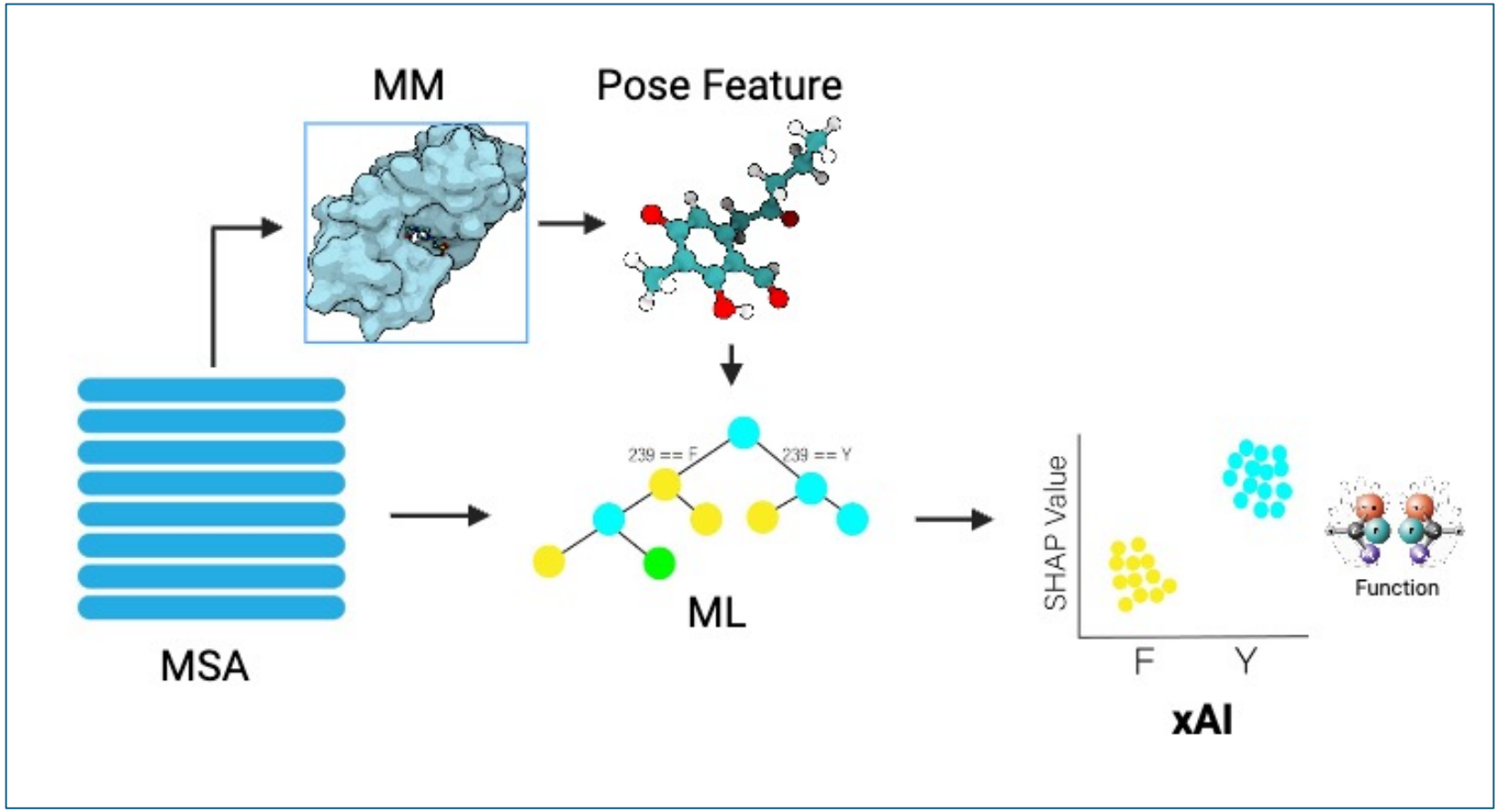
Assigning Labels with Structural Features and Molecular Modeling

- Lots of previous rational engineering studies to help pick structural feature to train on
- Modern forcefields are now fast and accurate



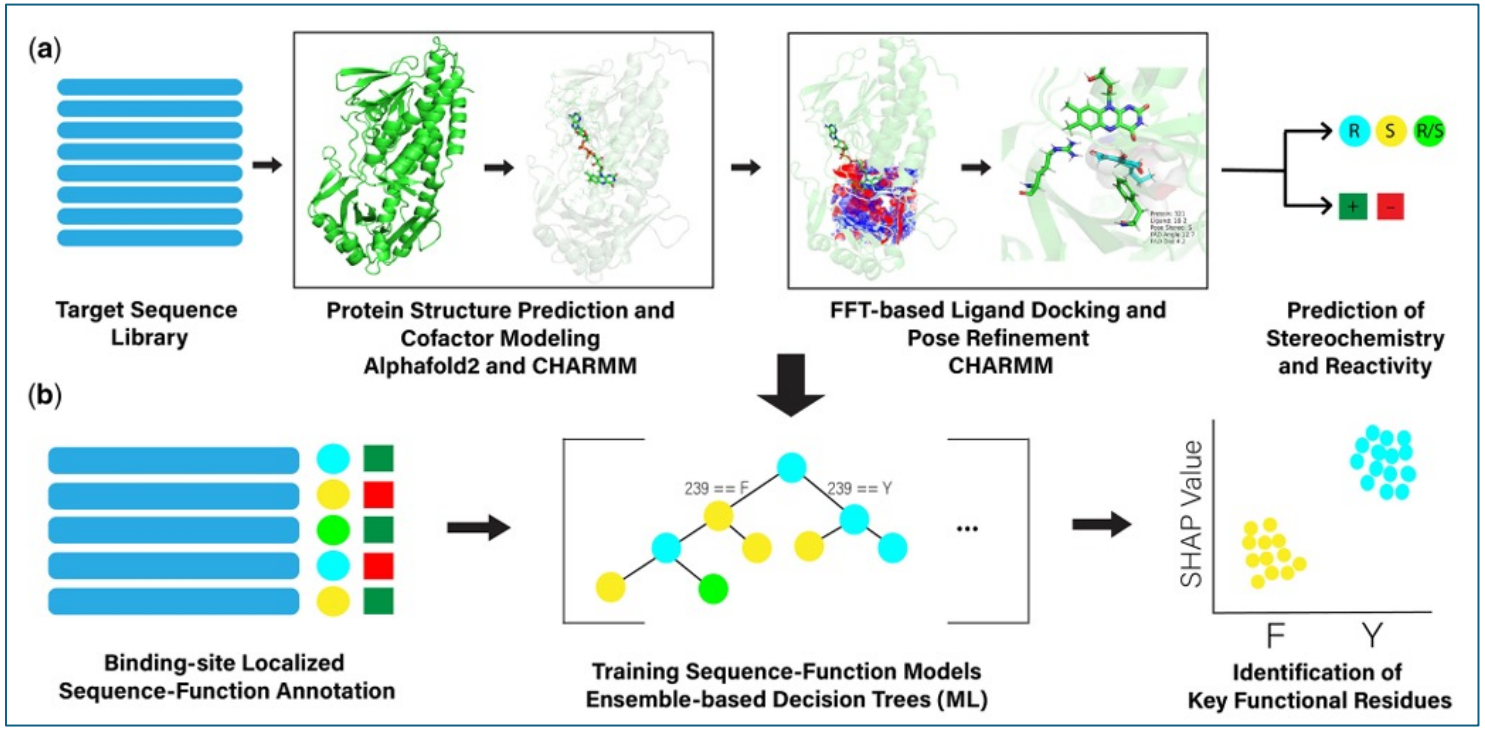
Explaining the Final Result with Explainable AI

- We can use **XAI** to explain our ML model
- Since the model features are amino acids that means we can ask how **specific residues** in a MSA column **map to the function** label



Final Pipeline: High Throughput ML and MM

- MSA with clustal omega
- Structure with **AlphaFold2**
- Cofactor with **CHARMM**
- Docking with **FFTDock**
- Prediction of Function
- **AutoML** on MSA + Labels
- **SHAP** for Key Residues
- **~30 cpu/gpu min per seq**



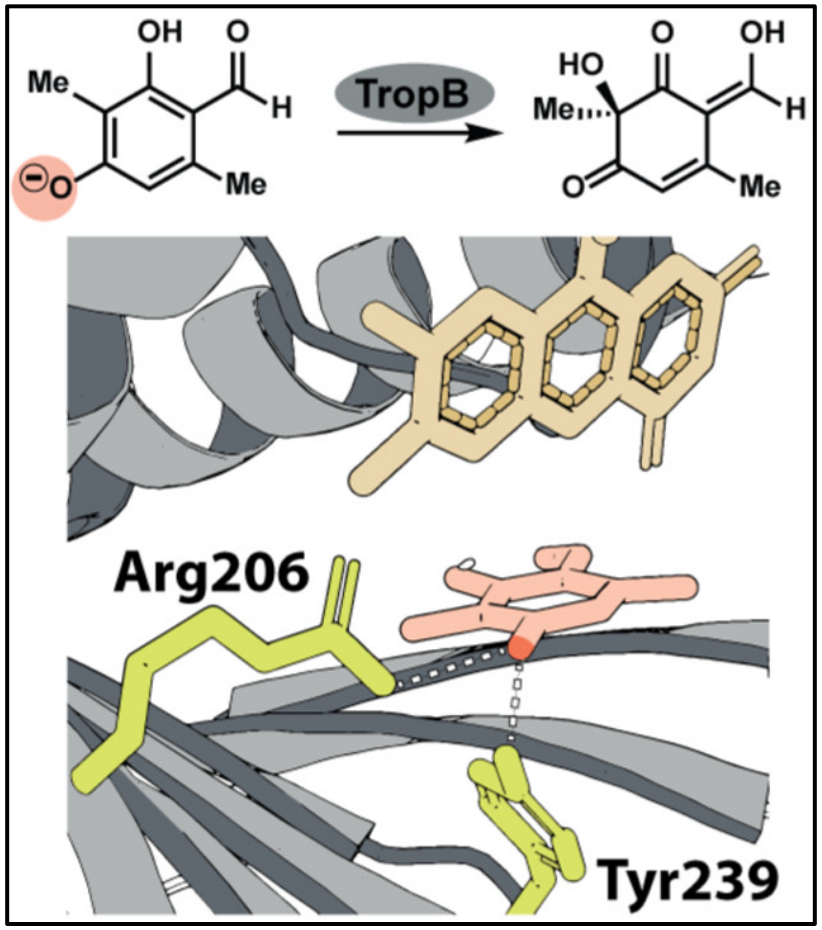
Hussain, A. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** (2024) doi:10.1093/bioinformatics/btae002.

Pipeline Walkthrough

BrooksResearchGroup-UM / **seq_struc_func**

Model System: Stereoselective Flavin Dependent Monooxygenases

- Despite thorough sequence and structural analysis, **only binding site** determinants of stereoselectivity / activity are known
  - ACS Catalysis 2019
    - Rational SDM, Crystal Structure, MD, QM/MM, Flexible Docking
  - JACS 2019
    - Sequence Similarity Networks
  - PNAS 2023
    - Ancestral Sequence Reconstruction + AlphaFold
- Previous success with modelling and rich untapped evolutionary space (ASR)



Chiang, Chang-Hwa, Troy Wymore, Attabey Rodríguez Benítez**, Azam Hussain**, Janet L. Smith, Charles L. Brooks III, and Alison R. H. Narayan. 2023. "Deciphering the Evolution of Flavin-Dependent Monooxygenase Stereoselectivity Using Ancestral Sequence Reconstruction." ***PNAS*** 120 (15): e2218248120.

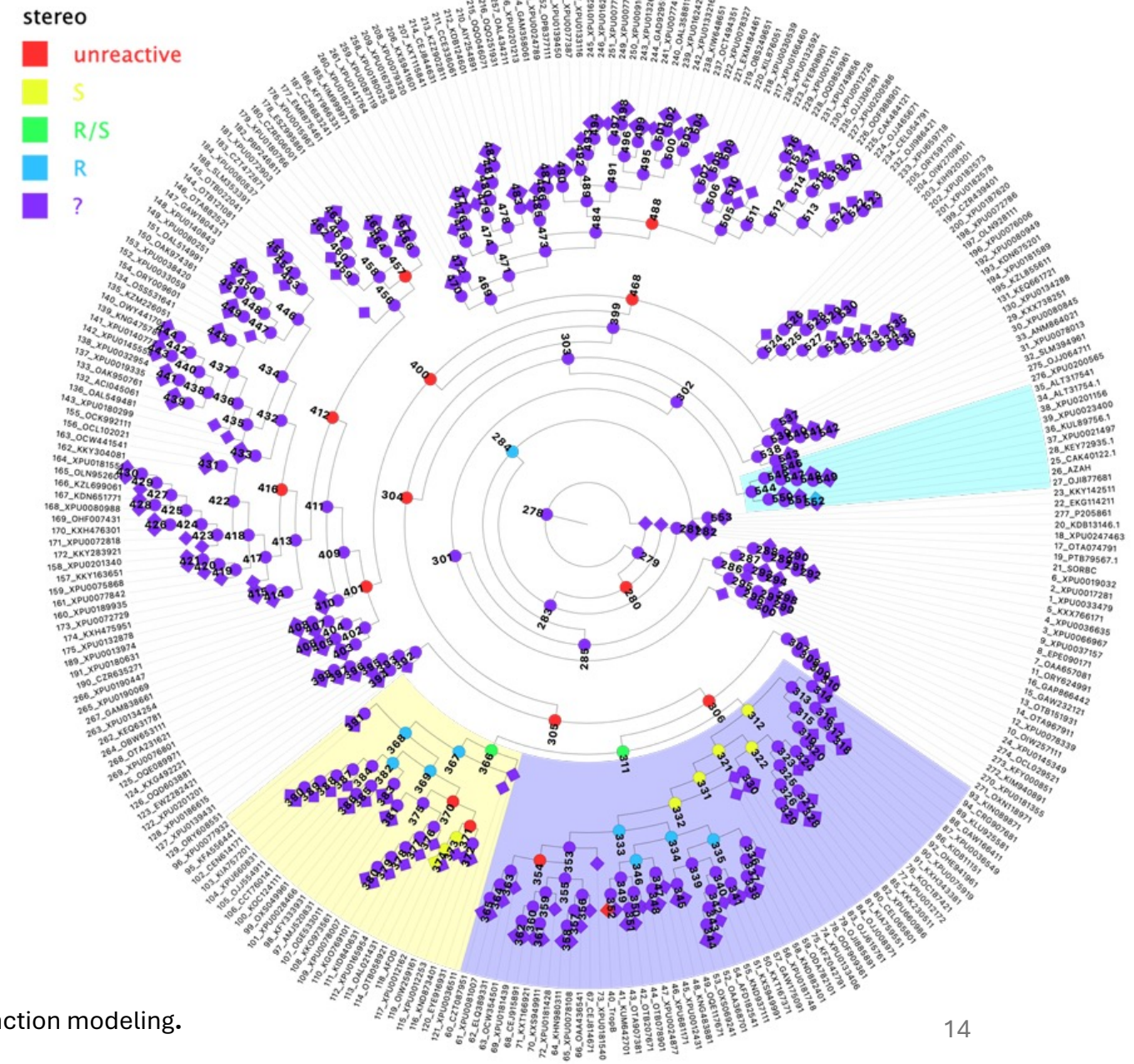
Model System Scope

19 R enzymes

15 S enzymes

33 inactive

838 total



Hussain, A. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** (2024) doi:10.1093/bioinformatics/btae002.

Step 1: (consensus/gen_consensus_db.ipynb)

- The MSA generation of AlphaFold is really slow.
- Since the sequences are homologous, lets run AF2 on the consensus sequence
  - **Reuse MSA hits and generate dataset**

In [4]:

```
#Generate consensus sequence (consensus.fasta)
os.system('rm ../consensus.fasta')
os.system(f'conda run -n seq_struct_func hhconsensus -M 50 -i {extant_msa_path} -s ../c
os.system('sed -i "1s/.*/>consensus/" ../consensus.fasta')
print(open('../consensus.fasta', 'r').read())
```

```
>consensus
MEMSQGFKSKPESMGLVDMSPSRSPVQTLRSTGYNRPEPAAAGKPQSTHPDSTSSSSRLRSMSSMSTSPPPQSPSMMMMASMMMMMMS
PKKPFEVAIVGGGIAGLTLAIGLLRRGIPVTIYEQAPAFREIGAGVAFTPNAVRAMKLIDPRIYEAFEKVATRNVWPSKPNDWFRWVD
GYNPDTKADDEEQELLFKLYAGERGFEGCHRAHFLDELVKLLPEGVVKFGKRLESIEEPEDSGKVVLHFADGTTAEADAVIGCDGIKS
RVRQLLLGEDHPAAYPSYTHKYAYRGLIPMDKAVEALGEEKARNSCMHLGPGAHVLTFPVANGTLLNVVAFVSDPEEWPDEDKMTAPA
TREEALRDFAGWGPTVRKIISLLPEKLDKWAIFDLGDHPAPTYAKGRVCLAGDAAHASSPHHGAGAGMGIEDALVLATLLEQVSQDAD
GRVSKAEALEAAFQAYDAVRRERSQWLVQSSREAGDLYEWRDPGVGRDFEKIFEEIKERSHKIWDFDIEAMLREAREEYEKRLSASSG
ASQGRGVAKAGAGAGNANAAAGAVLVVECRAWSVDREIPLNETPQVVEKSPQPVYKKNDMVS
```

In [6]:

```
#MSAs generated by AlphaFold2:
consensus_dir = f'../consensus_af2_results'
os.system(f'tar -xf {consensus_dir}/msas.tar.gz --directory {consensus_dir}')
bfd_uniclust_msa_file = os.path.join(consensus_dir, 'msas/bfd_uniclust_hits.a3m')
bfd_uniclust_fasta_msa_file = bfd_uniclust_msa_file.replace("a3m", "fas")
mgnify_msa_file = os.path.join(consensus_dir, 'msas/mgnify_hits.sto')
uniref90_msa_file = os.path.join(consensus_dir, 'msas/uniref90_hits.sto')
```

In [7]:

```
#Use reformat.pl from HH-suite3 to reformat all a3m to fasta for biopython
!./reformat.pl
```


https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/consensus/script/gen_consensus_db.ipynb

Step 2: (model/run_alphafold_consensus.ipynb)

- Slight modification to AlphaFold2 inference script with consensus sequences for MSA generation
- One RTX2080 takes ~ 5-10 min

In [12]:
```
#Perform example alphafold pipeline:
for fasta_path, model_dir, msa_dir in zip(fasta_paths, model_dirs, msa_dirs):
    t_0 = time.time()
    run_alphafold(fasta_path, model_dir, msa_dir)
    logging.info(f'TOTAL ALPHAFOLD2 PIPELINE TIME: {time.time() - t_0}s')
```

```
INFO:absl:Running AlphaFold for ../../si_data/fasta/tropb.fasta
INFO:absl:Constructing feature dictionary
INFO:absl:Generating MSA
INFO:absl:Launching subprocess "/home/azamh/miniconda3/envs/alphafold2/bin/jackhmmer -o /dev/null -A /tmp/tmp46z9z
cwj/output.sto --noali --F1 0.0005 --F2 5e-05 --F3 5e-07 --incE 0.0001 -E 0.0001 --cpu 8 -N 1 ../../si_data/fasta/
tropb.fasta ../../consensus/consensus_db.fasta"
INFO:absl:Started Jackhmmer (consensus_db.fasta) query
INFO:absl:Finished Jackhmmer (consensus_db.fasta) query in 86.013 seconds
INFO:absl:Searching for templates
INFO:absl:Launching subprocess "/home/azamh/miniconda3/envs/alphafold2/bin/hhsearch -i /tmp/tmp2jp27vk7/query.a3m
-o /tmp/tmp2jp27vk7/output.hhr -maxseq 1000000 -d /data/alphafold/data/pdb70/pdb70"
INFO:absl:Started HHsearch query
INFO:absl:Finished HHsearch query in 121.897 seconds
```


https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/model/script/run_alphafold_consensus.ipynb

Fast High Quality Apo Structures

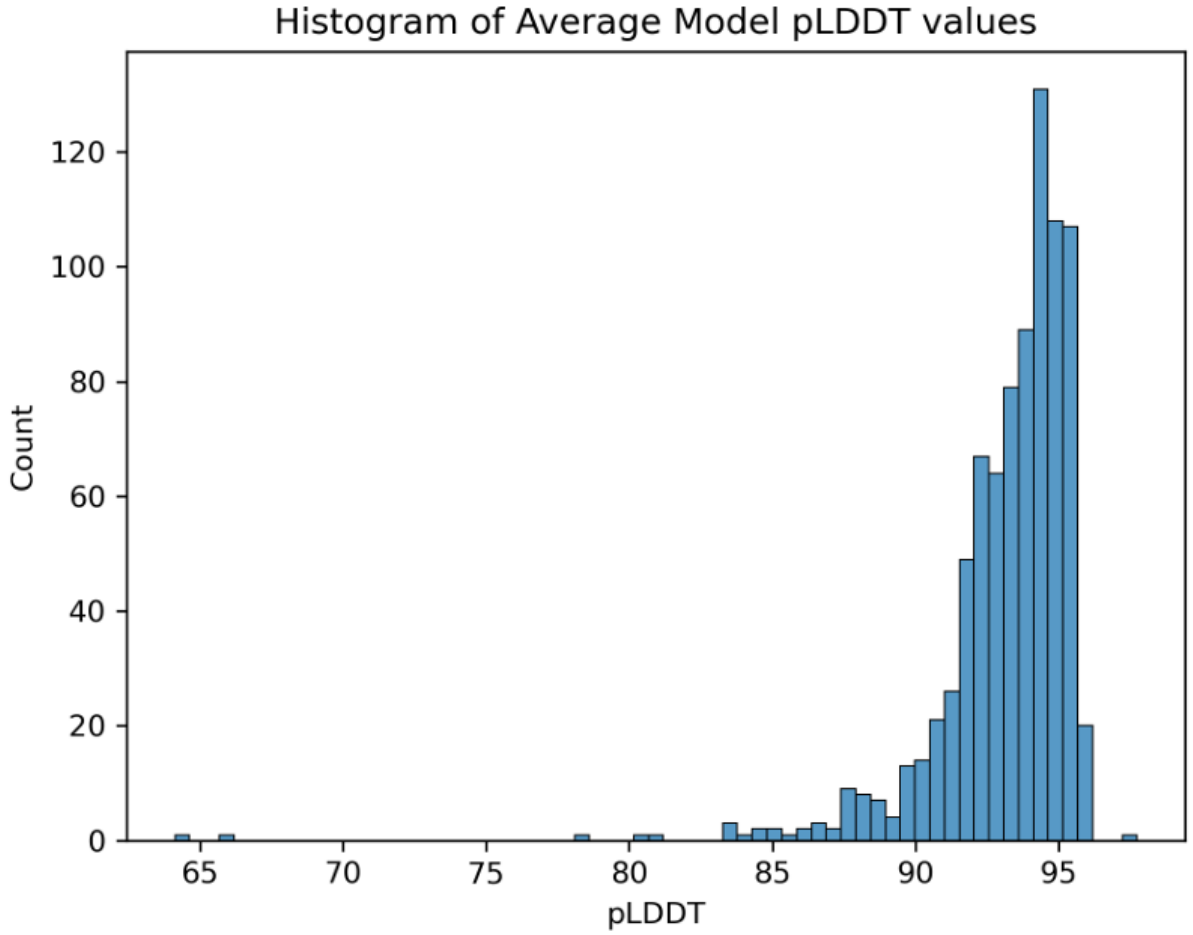
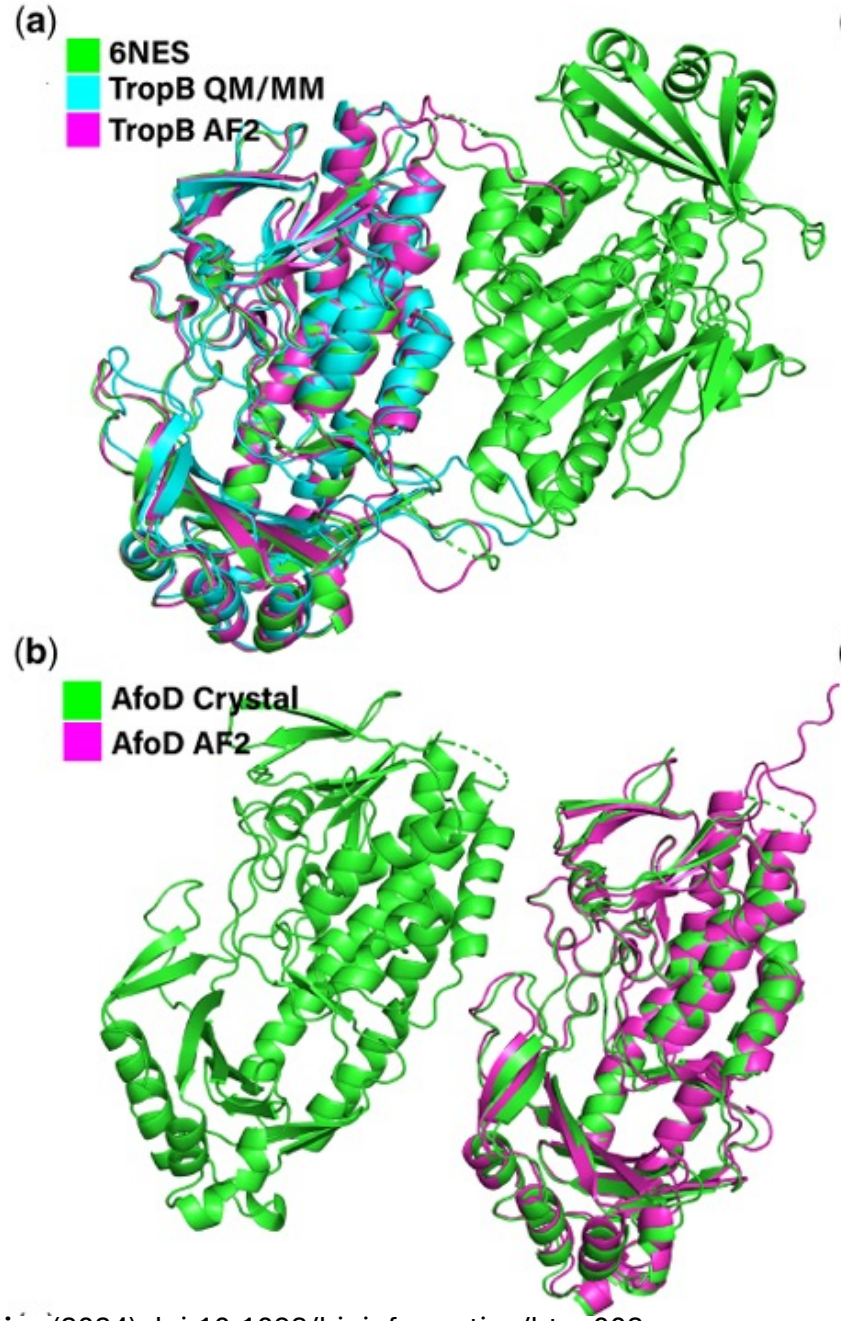


Figure S2: Histogram of average pLDDT[16] values across models. pLDDT values were averaged across all residues.

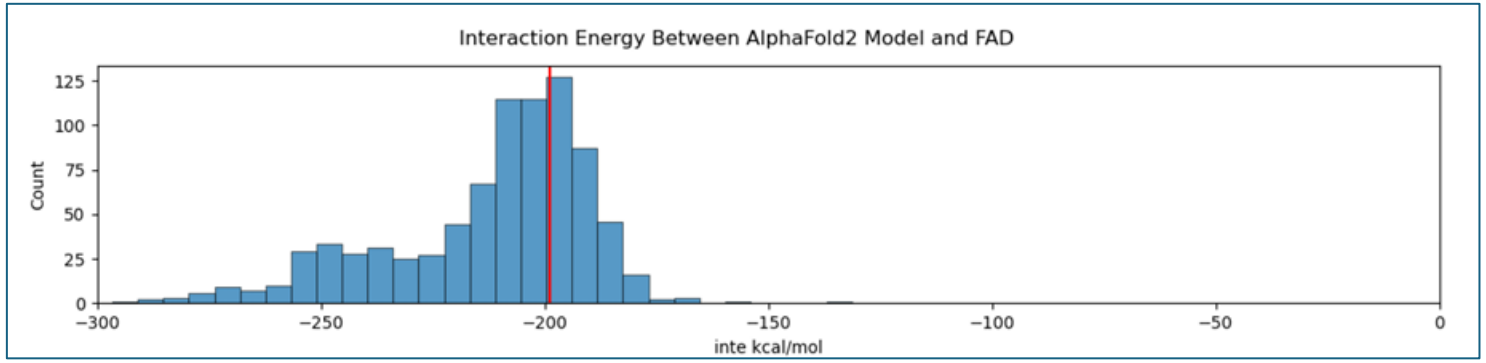


Hussain, A. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** (2024) doi:10.1093/bioinformatics/btae002.

Step 3: (cofactor/fad_pycharmm.ipynb)

- Our enzyme uses FAD as a cofactor to hydroxylate its substrate
- Minimize FAD from TropB crystal structure into AF2 predictions
  - CHARMM forcefield in python (pyCHARMM)
  - Similar to AlphaFill

```
#restrain everything except fad
cons_harm.setup_absolute(selection=protein_sel,force_constant=50)
minimize.run_sd(nstep=200, tolenr=1e-3, tolgrd=1e-4)
minimize.run_abnr(nstep=1000, tolenr=1e-3, tolgrd=1e-4)
cons_harm.turn_off()
```



Red line: crystal interaction energy

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/cofactor/script/fad_pycharmm.ipynb

Step 4.1: (dock/fftdock.ipynb)

- How can we get fast initial placements for our enzyme substrate in the pocket?
- Use FFTDock (CHARMM + cuFFT) on protein ligand grids

In [11]:

```
#Perform FFTDock
nsave = 500
charmm_script(f"fftg lcon ncon 1 icon 1  nrok {nsave} quau 31 sizb 100 select segid LIGA end")
charmm_script("close unit 31")
```

```
 CHARMM>     fftg lcon ncon 1 icon 1  nrok 500 quau 31 sizb 100 select segid LIGA end
 allocate gpu
Num of GPU Devices: 1
The device 0 is used.
  GPU Devices Name: NVIDIA GeForce GTX 1080 Ti
  total global devices memory: 11172 MB
 flag_init_quaternions_from_file  F
FFTDOCK - num of orientations: 36864
FFTDOCK - batch_size:   100
FFTDOCK - num_batch:   369
 flag_init_lig_conformers  F
 SELRPN>     32 atoms have been selected out of     32
<FFTDock> Initializing ligand grid
Total number of grids =    27
Types of VDW grid used  =     9
BatchId / Total =     1 /   369
Batch size is 10
CuFFT result is 0
Estimated grid memory is 0
Batch size is 1000
CuFFT result is 0
Estimated grid memory is 72
BatchId / Total =     2 /   369
```

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/dock/script/fftdock.ipynb

Step 4.2: (dock/prot_min.ipynb)

- Minimize FFTDock predictions in CHARMM **all atom forcefield** for more accurate final poses and energies

In [12]:

```
#Minimize all fftdock poses
nsave = 500
pose_energy_dfs = [initial_energy_df]

#Hide large output
settings.set_verbosity(0)
settings.set_warn_level(-2)
for i in tqdm(range(1, nsave + 1)):

    #Read FFTDock pose
    fftdock_pose = os.path.join(fftdockdir, f'{protein}_{ligand}_{i}.crd')
    read.pdb(fftdock_pose, resid=True)
    energy.show()

    #Perform minimization in explicit protein
    minimize.run_sd(nstep=50)
    minimize.run_abnr(nstep=1000, tolenr = 1e-3)

    #Get refined energy
    pose_energy_df = get_energy_df(i)
    pose_energy_dfs.append(pose_energy_df)

    #write pdb
    pose_pdb = os.path.join(dockdir, f'{protein}_{ligand}_{i}.pdb')
    write.coor_pdb(pose_pdb, sele = 'segid LIGA end')

settings.set_verbosity(5)
settings.set_warn_level(0)
```

```
100%|██████████| 500/500 [03:07<00:00,  2.67it/s]
```

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/dock/script/prot_min.ipynb

Step 4.3 (dock/cluster.ipynb)

- Cluster minimized poses for structural analysis
- Clusters with **representative pose** and **energies**

In [7]:

```
cluster_df = pd.DataFrame.from_records(cluster_dicts)
cluster_df = cluster_df.sort_values(by=['min_ener'])
cluster_df = cluster_df.reset_index(drop=True)
cluster_df.to_excel(f'../cluster/{protein}_{ligand}_prot.xlsx')
cluster_df.head(15)
```

Out[7]:

	cluster	size	min_ener	min_index	average energy
0	[111, 163, 166, 184, 195, 204, 209, 231, 268, ...	24	-48.82	393	-21.16
1	[126, 12, 143, 145, 162, 164, 167, 168, 169, 1...	31	-46.75	126	-14.53
2	[312, 313, 315, 332, 369, 394, 415, 453, 485]	9	-34.46	332	-13.83
3	[336, 439, 475, 494, 4]	5	-34.36	439	0.03

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/dock/script/cluster.ipynb

Step 5: (script/stereo.ipynb):

- If we assume little – no rearrangement the bound pose should reflect the preferred stereochemistry of the enzyme
- **Assign binary stereochemistry label for each cluster**
- **Assign protein stereochem from Boltzmann probabilities**

```
In [8]:  #Calculate overall R frac with ensemble of clusters
         R_frac, S_frac = partition_function(cluster_df)
         R_frac

Out[8]:  0.9999999999999999
```

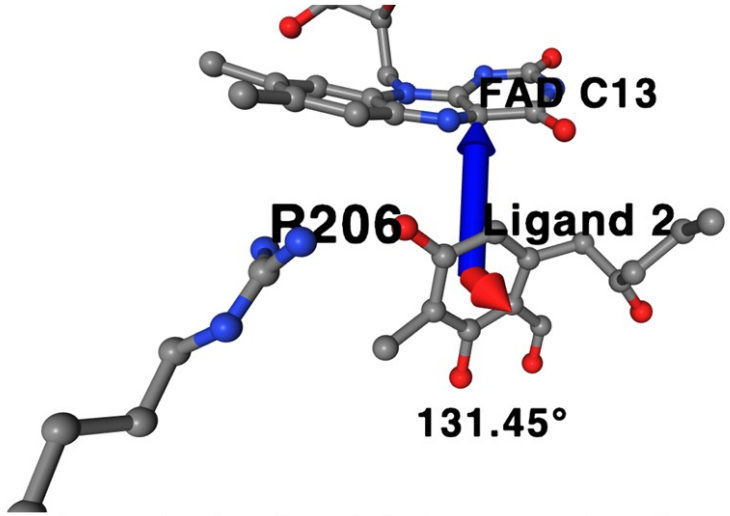
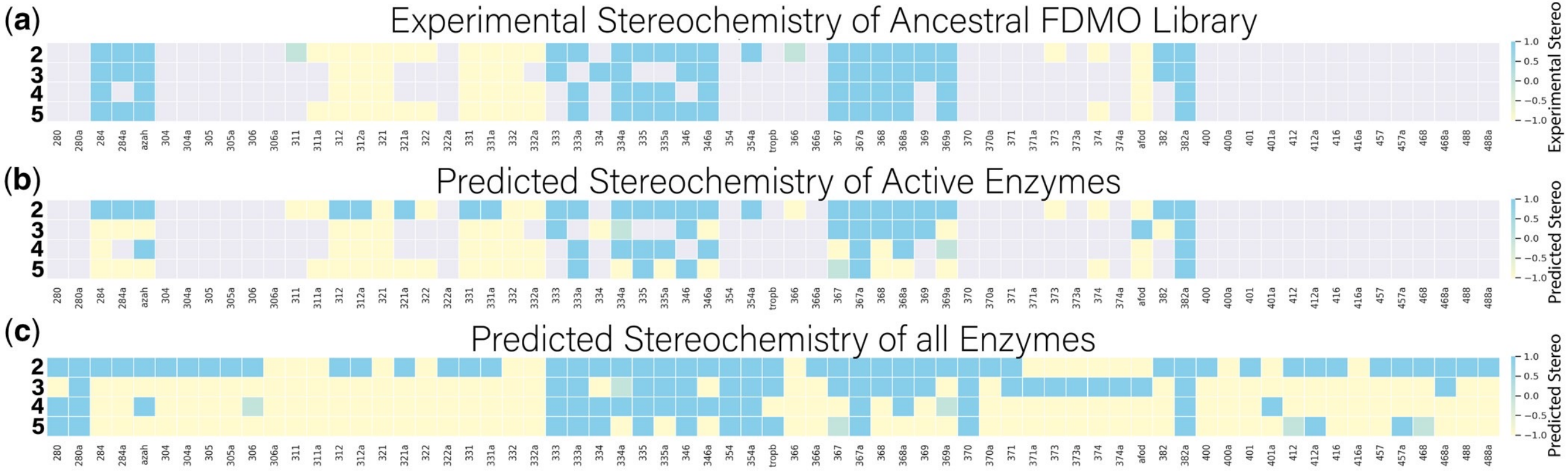


Figure S6: Prediction of stereochemistry from docked structures. The red vector indicates the normal vector the plane of the ligand resorcinol ring defined by three atoms. The blue vector indicates the vector from the ligand resorcinol ring and anion average coordinates to the FAD C13 atom. The angle between the blue and red vectors (131.45°) is used to classify the stereochemistry of a pose as R or S, with an angle greater than 90 ° as R, otherwise as S.

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/pred/script/stereo.ipynb

Validation Against Experimental Stereochemistry Data (R, S, Inactive)



Hussain, A. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** (2024) doi:10.1093/bioinformatics/btae002.

Step 5: (script/reactivity.ipynb):

- **Reactivity label** from logistic regressor trained features derived from protein-ligand complexes
  - Dock energy
  - FAD angle
  - Distance
  - CNN pred binding affinity
  - 80% accuracy

```
print('dock_energy_efficiency', dock_energy_efficiency, 'kcal/mol/HA')
print('angle', angle, 'degrees')
print('anion_dist', anion_dist, 'angstrom')
print('pred_pkd_efficiency', pred_pkd_efficiency, '1/HA')
```

```
dock_energy_efficiency -2.8717647058823528 kcal/mol/HA
angle 67.91 degrees
anion_dist 4.753537209279002 angstrom
pred_pkd_efficiency 0.3680809823529412 1/HA
```

```
#Predict with logistic regression model: 0 -> unreactive, 1 -> reactive
log_reg.predict([angle, anion_dist, dock_energy_efficiency, pred_pkd_efficiency])
```


https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/pred/script/reactivity.ipynb

Step 6: (msa/slice_msa.ipynb)

- For training ML model on assigned stereochemistries and reactivities we need MSA as input
- Generate and convert MSA to tabular format

```
#Prune columns with > 10% gaps
pruned_msa_df = prune_df(sliced_msa_df)
pruned_resi_df = prune_df(sliced_resi_df)
pruned_msa_df
```

	22I	50I	51G	52A	53G	54M	55A	56F	57T	58A	...
278	P	L	G	L	G	R	G	L	E	P	...
278a	P	L	G	L	G	V	A	F	E	P	...
279	M	L	G	V	G	I	H	F	T	P	...
279a	M	L	G	V	G	I	H	F	T	P	...
280	I	L	G	V	G	I	H	F	T	P	...
...	...	...	...	...	...	...	...	...	...	...	...
xp_659718	I	V	G	A	G	V	S	F	G	P	...
xp_660831	I	T	S	A	G	F	S	F	S	K	...
xp_660986	I	P	G	A	G	I	A	F	T	A	...
xp_681171	I	I	G	A	G	I	A	F	T	A	...
xp_749656	I	V	G	A	G	V	S	F	S	P	...

830 rows × 125 columns

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/msa/script/slice_msa.ipynb

Step 7: (seq_func/run_automl.ipynb)

- Fit multiple sequence alignment with assigned stereochemistries and reactivities
- Mljar AutoML explores hyperparameter variations of common ML models

```
#Create autoML object
automl_stereo = AutoML(
    mode = 'Perform',           #Do hyperparameter tuning
    total_time_limit=3600,      #1 hour
    results_path = resultsdir,
    explain_level = 2,          #Calculate SHAP values
    algorithms = [
                "Xgboost",
                "Random Forest",
                "CatBoost",
            ],
    golden_features=False)
```

```
#Fit automl (delete automl_stereo folder to rerun)
automl_stereo.fit(pruned_msa_df, asr_seq_annotations['average_pred_stereo'])
```

```
AutoML directory: ../automl_stereo
The task is multiclass_classification with evaluation metric logloss
AutoML will use algorithms: ['Xgboost', 'Random Forest', 'CatBoost']
AutoML will ensemble available models
AutoML steps: ['simple_algorithms', 'default_algorithms', 'not_so_random', 'inse
ction', 'hill_climbing_1', 'hill_climbing_2', 'ensemble']
Skip simple_algorithms because no parameters were generated.
* Step default_algorithms will try to check up to 3 models
```

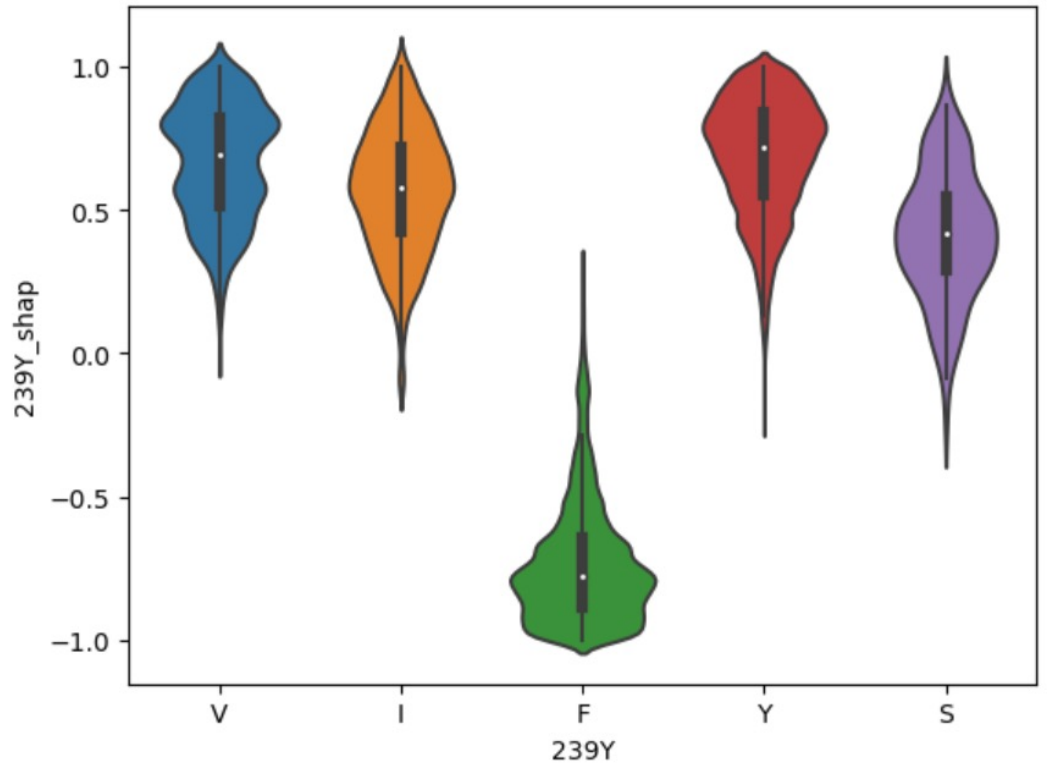

https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/seq_func/script/run_automl.ipynb

Step 7: (seq_func/shap_analysis.ipynb)

- Run SHAP Explainer on pretrained models
- Average normalized SHAP values across models and create **dependence plots for any residue**

In [11]:
```
#Create dependence plot for resi 239
#Positive shap value indicates R and negative indicates S
#Dependence plot is all shap values across all folds of all trained models (100,000 points)
sns.violinplot(data = all_fold_shap_df, y = f'239Y_shap', x = '239Y')
```

Out[11]: <AxesSubplot: xlabel='239Y', ylabel='239Y_shap'>



https://github.com/BrooksResearchGroup-UM/seq_struct_func/blob/main/seq_func/script/shap_analysis.ipynb

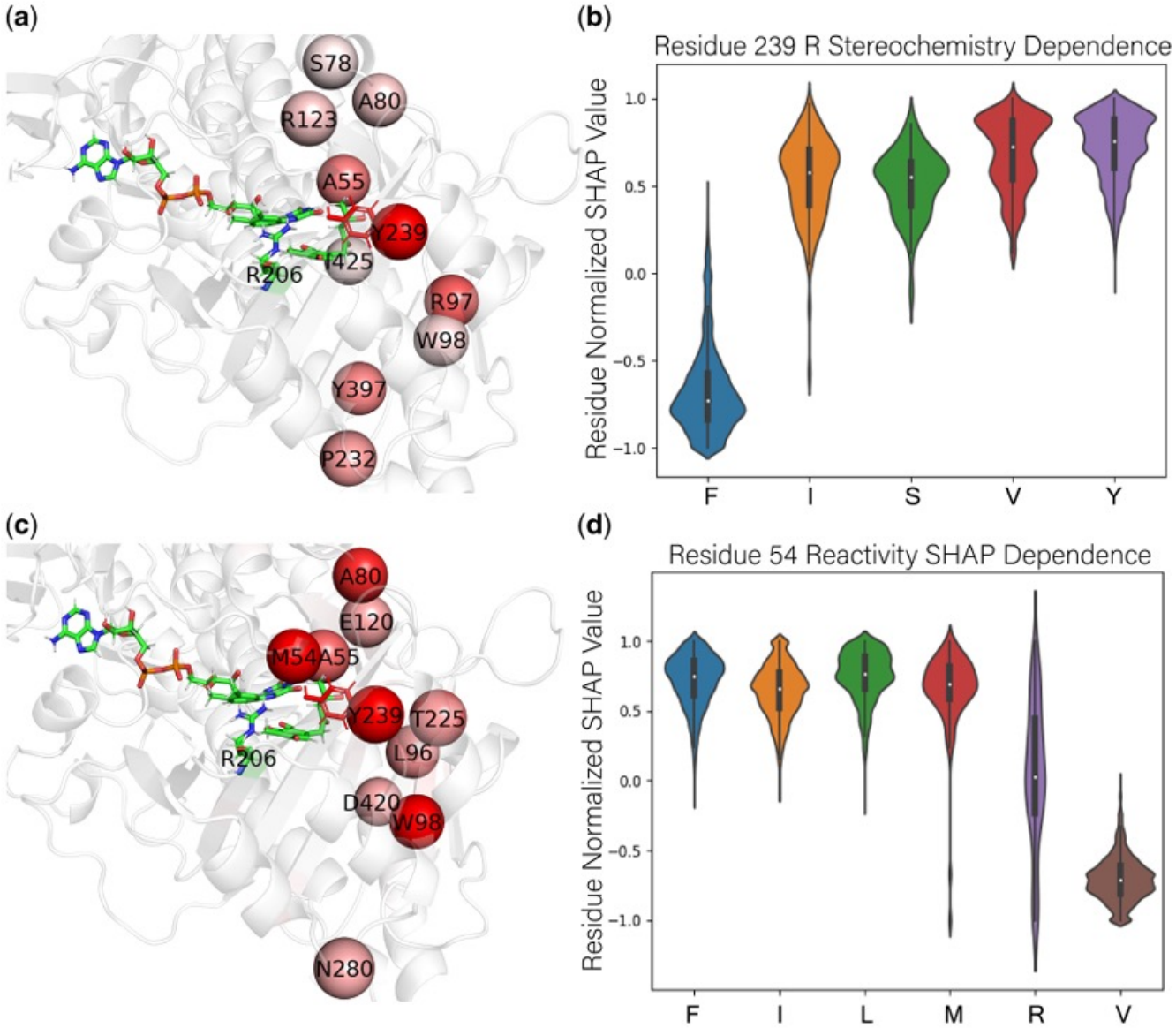
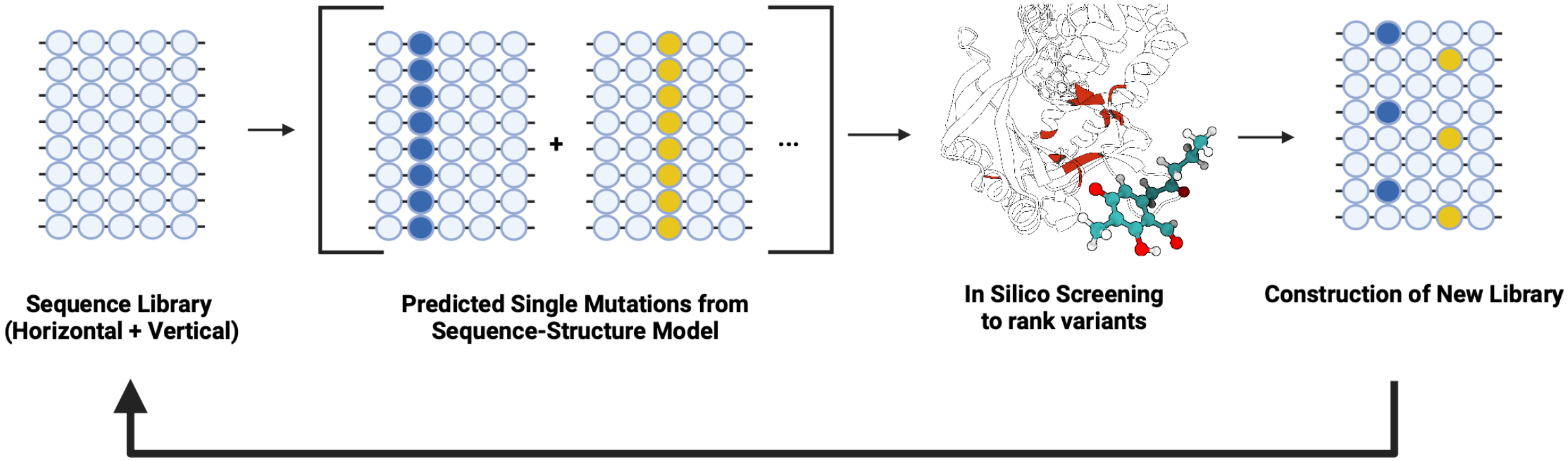


Figure 5. Key residues for stereoselectivity and reactivity control identified by SHAP analysis. a) Top 10 residues by mean absolute normalized SHAP value across all generated folds of all models for prediction of stereochemistry in TropB docked with **3**. The deep red spheres indicate a high mean absolute SHAP value, with Y239 having the largest value. b) SHAP dependence plot of residue 239 for the R class with normalized SHAP contribution of residue 239 across all folds of all models for full sequence library. c) Top 10 residues by mean absolute normalized SHAP value across all generated folds of all models for prediction of reactivity. M54 has the largest mean absolute SHAP value. d) SHAP dependence plot of residue 54 across all folds of all models for full sequence library for prediction of reactivity.

Hussain, A. & Brooks, C. L., 3rd. Guiding discovery of protein Sequence-Structure-Function modeling. ***Bioinformatics*** (2024) doi:10.1093/bioinformatics/btae002.

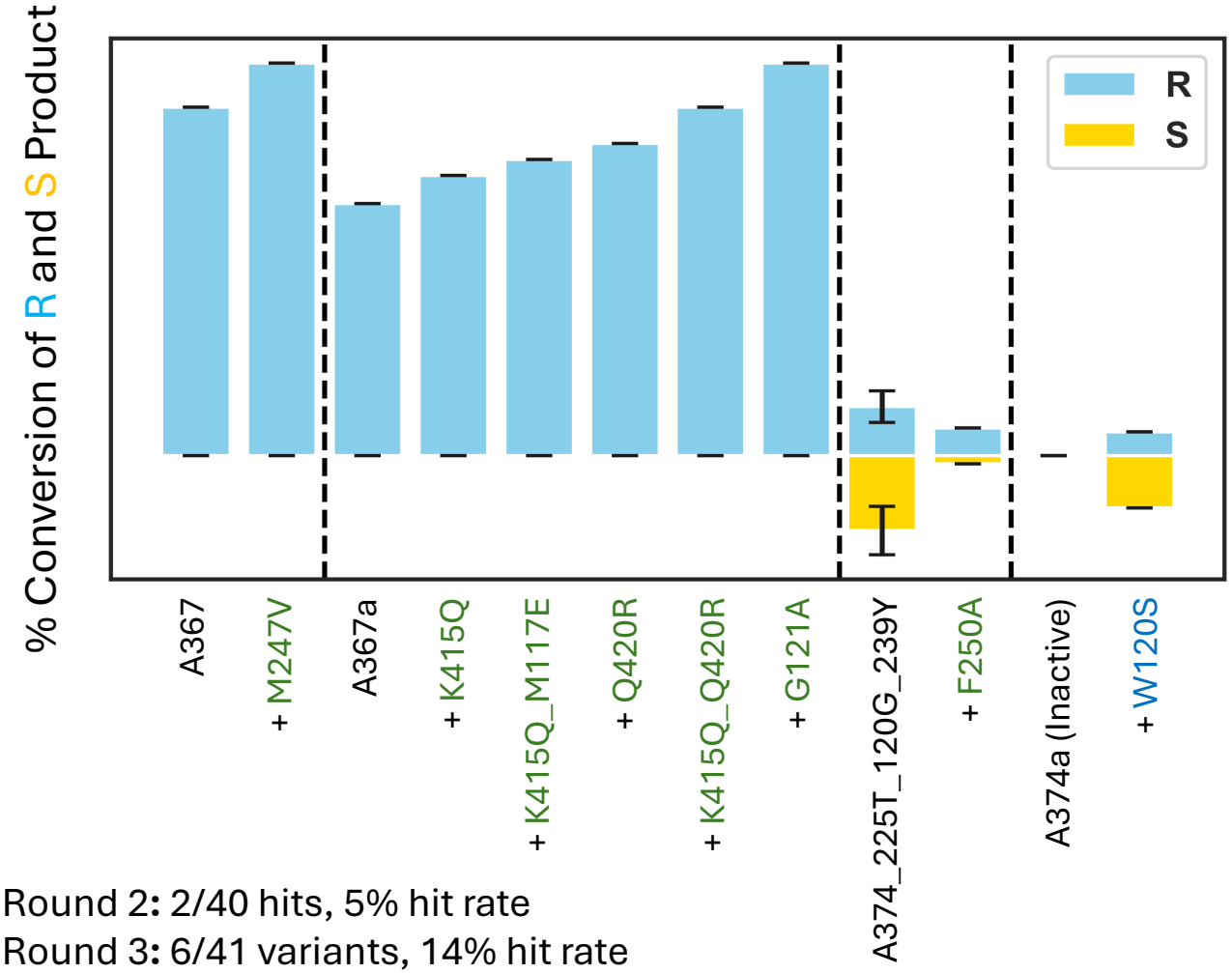
Future Directions and Conclusions

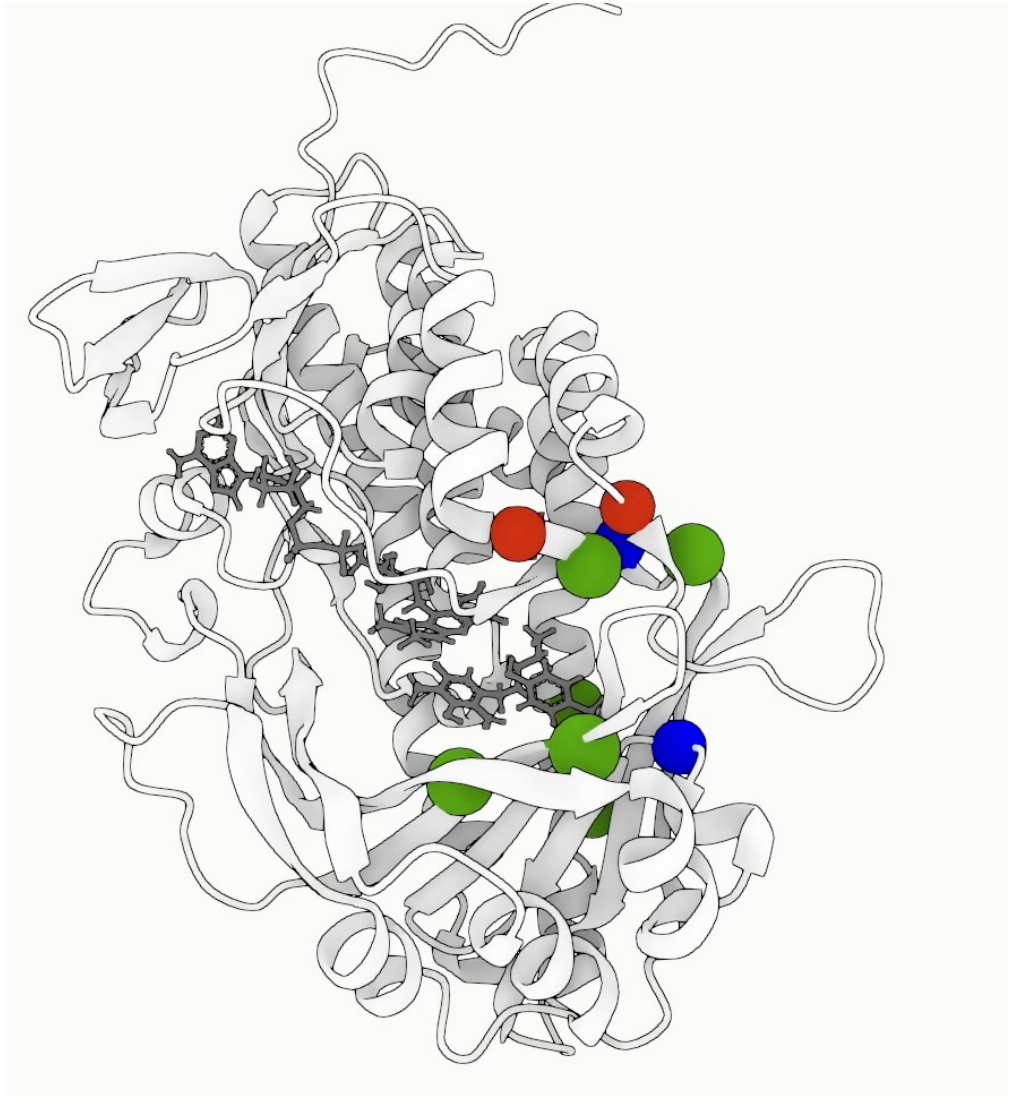
Next Project: Using Pipeline for In Silico Directed Evolution



Hussain et al. Integrating Molecular Modelling and Machine Learning for Low-N Protein Engineering (in preparation)

Initial Experimental Evidence for Low-N Discovery of Distal Residues with MM + ML





Round 2: 2/40 hits, 5% hit rate
Round 3: 6/41 variants, 14% hit rate
Almost all are second sphere + beyond

High throughput ML and MM for Protein Modeling

- Current approaches are now making modelling more and more accessible with limited resources
- By combining various SOTA approaches at different scales with domain knowledge, we can start to simply high dimensional problems like protein sequence-structure-function space
- Each part of the pipeline is from a generalized technique and the code can be repurposed for many different problems
  - Can steal snippets for other projects!
- Code, data, and paper are all open access

Acknowledgements

- Brooks Group (University of Michigan):
  - **Charles L. Brooks III**
  - **Chang-Hwa Chiang (Chad)**
- **Dr. Alison Narayan** (University of Michigan)
- Dr. Troy Wymore (Stony Brook University)

BrooksResearchGroup-UM/**seq_struct_func**

linkedin.com/in/**azamh**

Thank you!