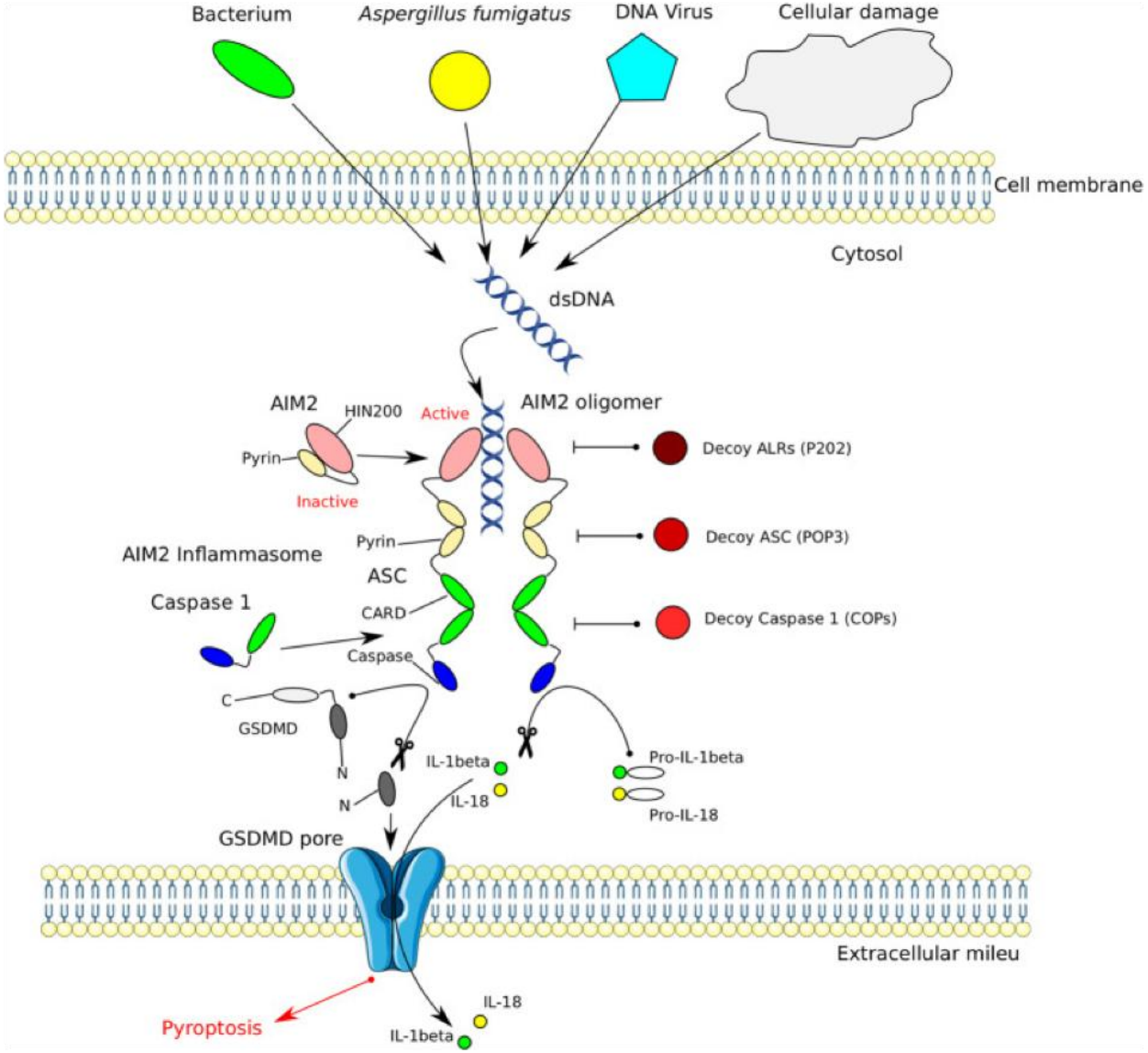
Using Alphafold-Multimer to predict the interaction of AIM2ΔPYD and IFI16β

Background

- Cells produce inflammasomes in response to internal or external threats
  - Inflammasomes contain sensor, adaptor, and effector proteins
  - An inflammasome will often lead to a cascade of signals that result in inflammatory cell death known as pyroptosis

- Two inflammasomes that respond to foreign dsDNA are Absent-in-Melanoma 2 (AIM2) and Interferon inducible protein 16 (IFI16)
  - P202 (shown in figure) is a notable mouse inhibitor of AIM2 with a human homologue called IFI16β



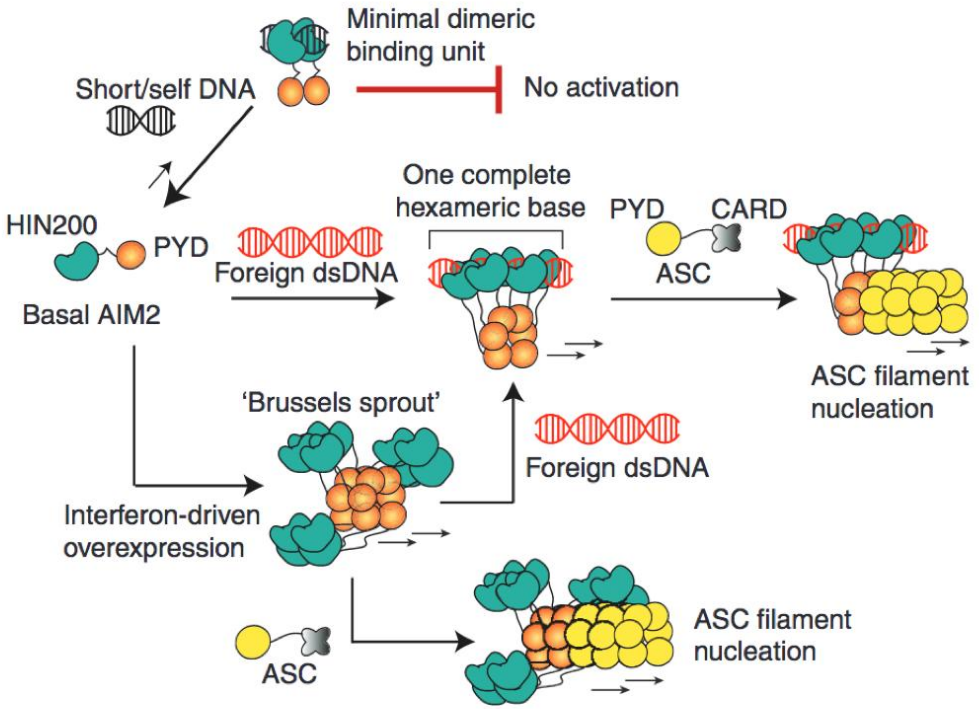
Schematic model of AIM2 inflammasome activation. Adapted from Zheng Danping et al. *Cell Discovery* 6 (June 9, 2020): 36. DOI: 10.1038

Background

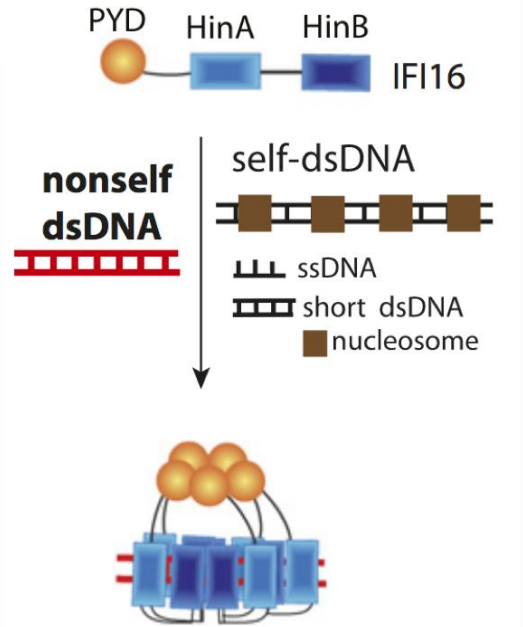
- Absent-in-Melanoma 2 (AIM2)
  - Produces an inflammasome in response to cytosolic dsDNA
  - Contains a Hematopoietic Interferon-inducible Nuclear domain (HIN) and Pyrin Domain (PYD)

- Interferon inducible protein 16 beta (IFI16β)
  - An isoform of IFI16 with only two HIN domains and a lacking a PYD
  - Coprecipitates with AIM2 from cell lysate and regulates AIM2 mediated inflammation

- The structural components of the interaction between AIM2 and IFI16β is poorly understood

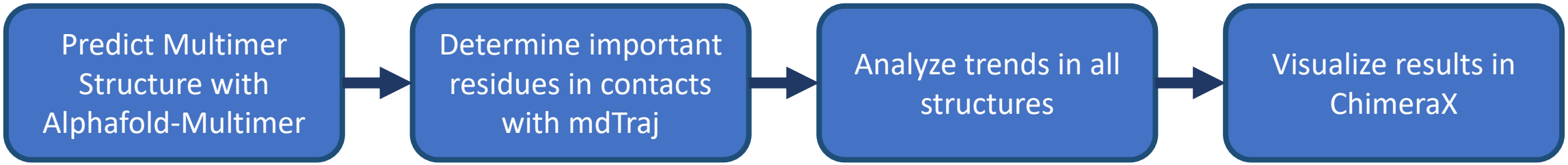


Model for AIM2 on dsDNA. The binding of AIM2 to dsDNA, with ASC filament formation. Adapted from Morrone, Seamus R. et al. *Nature Communications* 6 (July 22, 2015): 7827. DOI: 10.1038



Model for IFI16 on dsDNA. A model of IFI16 producing a platform of ASC filament formation. Adapted from Morrone, Seamus R. et al. *Proceedings of the National Academy of Sciences* 111, no. 1 (January 7, 2014): E62–71. DOI: 10.1073

Project Overview:



What is Alphafold?

- Alphafold2 is an artificial intelligence powered algorithm to solve the 3D structures of proteins using only the polypeptide sequence and the database of solved 3D structures

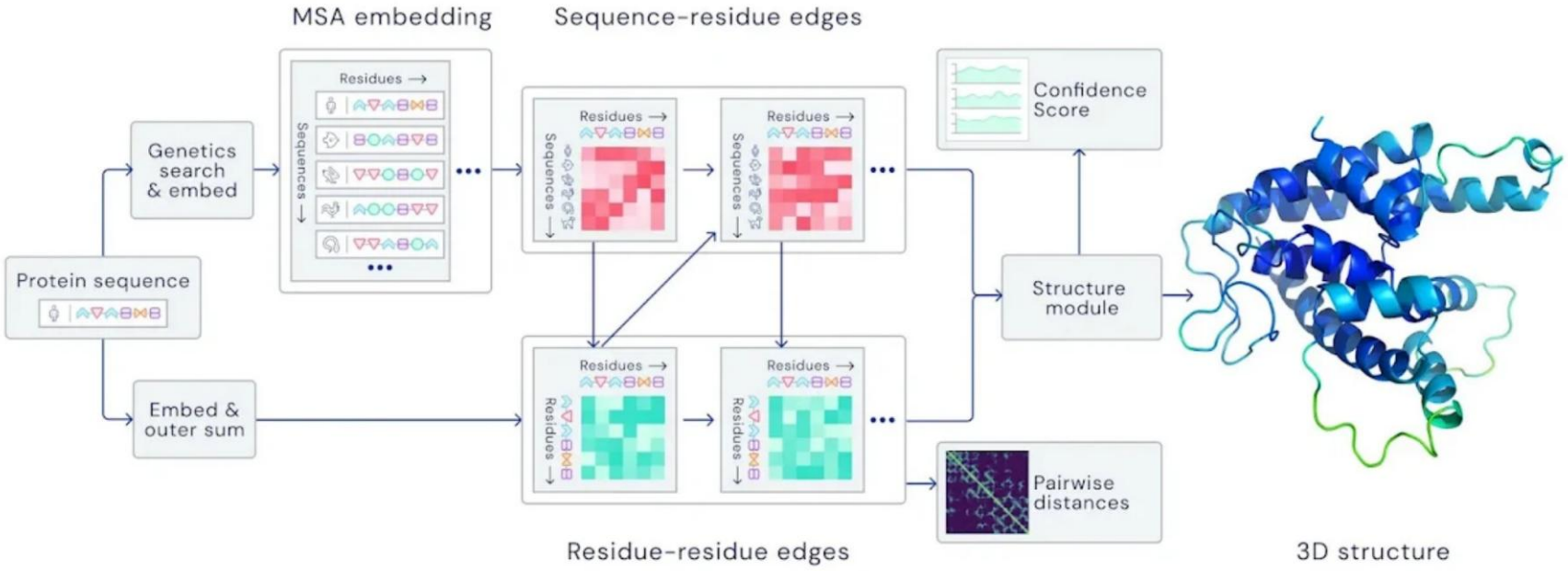


Image retrieved from: Google DeepMind. “AlphaFold: A Solution to a 50-Year-Old Grand Challenge in Biology,” November 30, 2020. https://deepmind.google/discover/blog/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology/.

Using the Computational Cluster

- The Merced Cluster can run a common use Alphafold program known as Colabfold
- Requires a fasta formatted protein sequence (use a colon to separate for multimers)

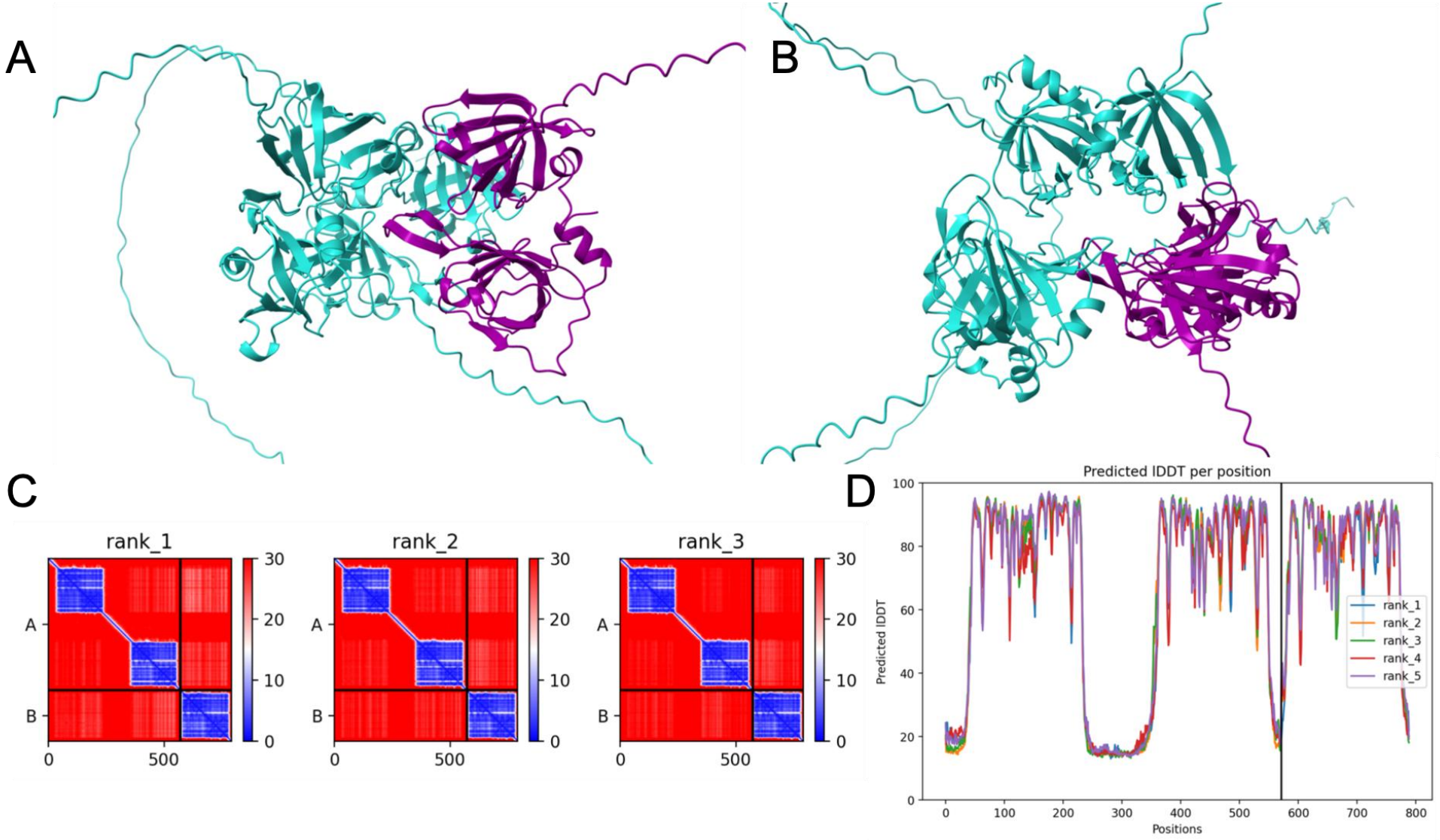
```
#!/bin/bash
#SBATCH --nodes=1  # requesting 1 node
#SBATCH --ntasks=28 # requesting 28 cores
#SBATCH --partition gpu  # submitting to the test partition
#SBATCH --gres=gpu:1 # requesting 1 gpu
#SBATCH --time=1-24:00:00 # setting a time limit of 1 day
#SBATCH --output=run_multi_new.qlog  # directing output to test1.qlog file
#SBATCH --job-name=AF2_SD_multi  # specifying the job name
#SBATCH --export=ALL

# Load necessary modules
module load cuda/11.8.0
module load localcolabfold

# Run colabfold with specific parameters
colabfold_batch --amber --num-relax 3 --model-type alphafold2_multimer_v3 IFI16AIM2_multi.fasta ./multi_new/
```

IFI16AIM2_multi.fasta

```
>IF16_beta_AIM2_HIN
MSTAMGRSPSPKTSLSAPPNSSSTENPKTVAKCQVTPRRNVLQKRPVIVKVLSTTKPFEY
ETPEMEKKIMFHATVATQTQFFHVKVLNTSLKEKFNGKKIIIISDYLEYDSLLEVNEEST
VSEAGPNQTFEVPNKIINRAKETLKIDILHKQASGNIVYGVFMLHKKTVNQKTTIYEIQD
DRGKMDVVGTGQCHNIPCEEGDKLQLFCFRLRKKNQMSKLISEMHSFIQIKKKTNPRNND
PKSMKLPQEQRQLPYPSEASTTFPESHLRTPQMPPTTPSSSFFTKKSEDTISKMNDFMRM
QILKEGSHFPGPFMTSIGPAESHPHTPQMPPSTPSSSFLTTLKPRLKTEPEEVSIEDSAQ
SDLKEVMVLNATESFVYEPKEQKKMFHATVATENEVFRVKVFNIDLKEKFTPKKIIAIAN
YVCRNGFLEVYPFTLVADVNADRNMEIPKGLIRSASVTPKINQLCSQTKGSFVNGVFEVH
KKNVRGEFTYYEIQDNTGKMEVVVHGRLTTINCEEGDKLKLTCFELAPKSGNTGELRSVI
HSHIKVIKTRKNKKDILNPDSSMETSPDFFF:MVAQQESIREGFQKRCLPVMVLKAKKPFTFETQEGKQEMFHAT
VATEKEFFFVKVFNTLLKDKFIPKRIIIIARYYRHSGFLEVNSASRVLDAESDQKVNVPL
NIIRKAGETPKINTLQTQPLGTIVNGLFVVQKVTEKKKNILFDLSDNTGKMEVLGVRNED
TMKCKEGDKVRLTFFTLSKNGEKLQLTSGVHSTIKVIKAKKKTGGGGSLPETGGG
```

Alphafold Results

Actual AIM2 HIN for reference. The dsDNA binding site is on the right face. (PDB 3RN2)



Colabfold-solved multimer structure of AIM2 HIN and IFI16β. (A-B) Ribbon representation of predicted Rank_1 AIM2 HIN (purple) and IFI16β (cyan) interaction from side and top view. (C) Predicted Aligned Error (PAE) for top three predicted structures. (D) Predicted lDDT for each structure. Average lDDT between 66-68 for all.

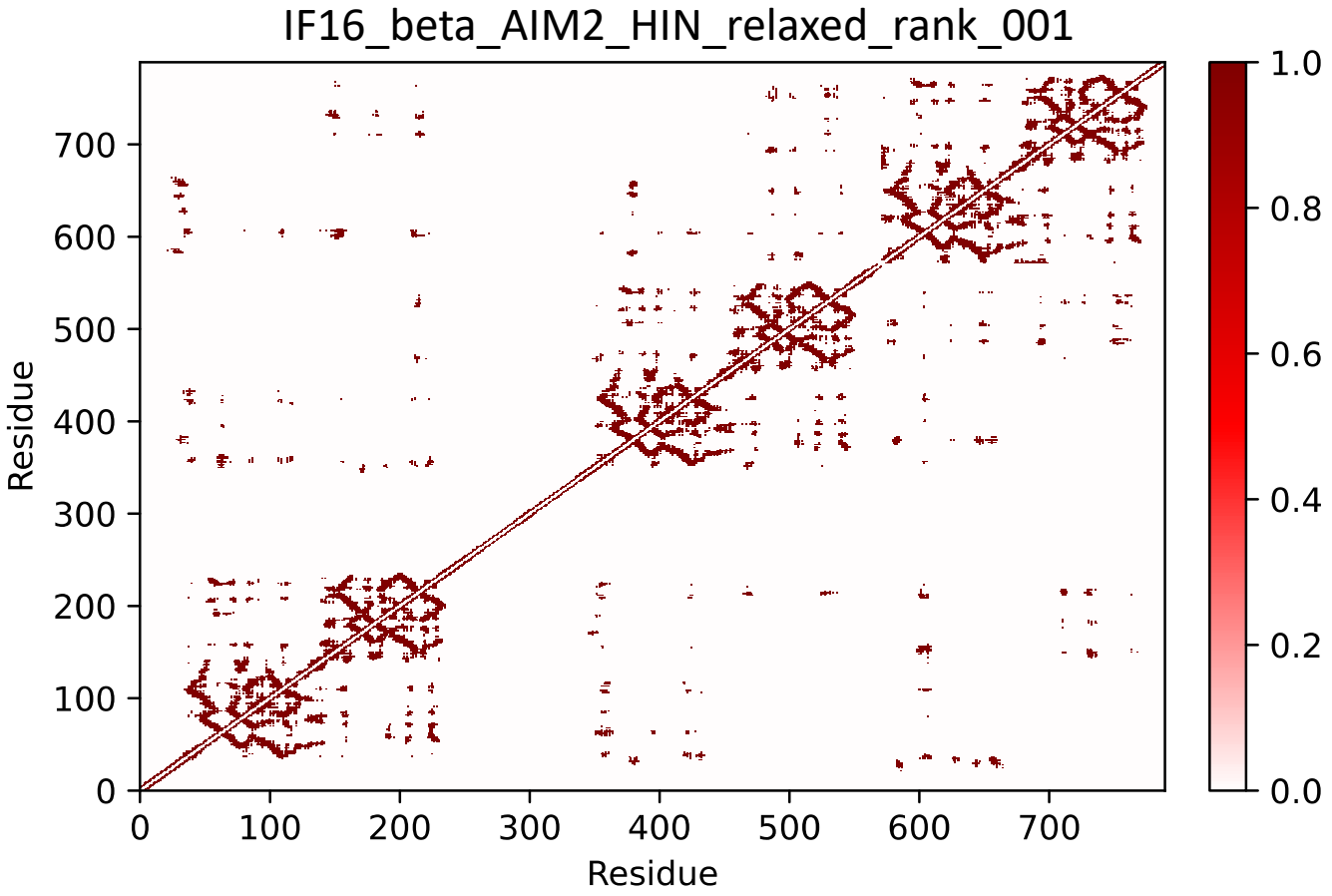
Interactions calculated with mdTraj library

- mdTraj is a python library to analyze molecular dynamics trajectories produced from a pdb file
- The md.compute_contacts() function calculates the distances between residues
  - Here it calculates by ‘closest-heavy’ or the closest non-hydrogen atom present
- Outputs all pairs below 1 nm in distance
- Calculates Solvent Accessible Surface Area for each important residue

```
import mdtraj as md
from contact_map import ContactFrequency, ContactDifference
import itertools as itertools
import matplotlib.pyplot as plt
import numpy as np
from Bio.SeqUtils import seq1
import pandas as pd
#%% begin function to take a file and run contact analysis
def contacts(file):
    #file = 'IF16_beta_AIM2_HIN_unrelaxed_rank_001' #if running as non-function
    t = md.load('multi_new/'+file + '.pdb') #create trajectory
    residues = list(t.topology.residues)
    residue_indices = [residue.index for residue in t.topology.residues]
    index1 = residue_indices[0:570] #residues for IFI16b
    index2 = residue_indices[571:788] #residues for AIM2 HIN
    pairs = list(itertools.product(index1,index2)) #create all possible pairs
    #%% extract sequence from residue info
    fullreslist=np.array([])
    for ind in residue_indices:
        fullres = t.topology.residue(ind)
        fullresname =fullres.name
        fullreslist=np.append(fullreslist,fullresname)
    #%% produce a contact map for the protein
    frame_contacts = ContactFrequency(t[0],cutoff=1)
    fig, ax = frame_contacts.residue_contacts.plot()
    plt.xlabel("Residue")
    _ = plt.ylabel("Residue")
    plt.savefig('multi_new/'+file+'.pdf', format='pdf')
    #%% calculate contacts and significant contacts
    distrow,respair = md.compute_contacts(t,pairs,scheme='closest-heavy')
    distcount = distrow.shape[1]
    dist = distrow.reshape(distcount,1)
    sigres_ind = np.where(dist[:,0]<1)[0] #find index for distances below 1 nm
    sigpair = respair[sigres_ind]
    siguniq=np.unique(sigpair) #significant residues
    distsig=dist[sigres_ind] #collect data for significant residues
    distgrid=np.hstack((sigpair,distsig))
    distdf=pd.DataFrame(distgrid,columns=['Residue1','Residue2','Distance (nm)'])
    distdf.to_csv('multi_new/'+file+'distances.csv',index=None)
```

Contact Map

- Contact Map Explorer is a helpful package that uses mdTraj to produce a contact map “fingerprint” for a protein
- Using a cutoff of 1 nm, a contact map was made for each protein
- Observable clusters of residues between HIN domains



Common Residues between Structures

- Between all 5 unrelaxed structures, there should be some overlap, indicating that those residues are most important for the interaction
- A new python script takes the first script as a function and iterates through all 5 unrelaxed structures to determine residue counts
- This script then provides a ChimeraX command to select residues in the following categories for visualization:
  - Appears in all 5 structures (red), appears in 3 to 4 structures (orange), appears in 1 to 2 structures (yellow).

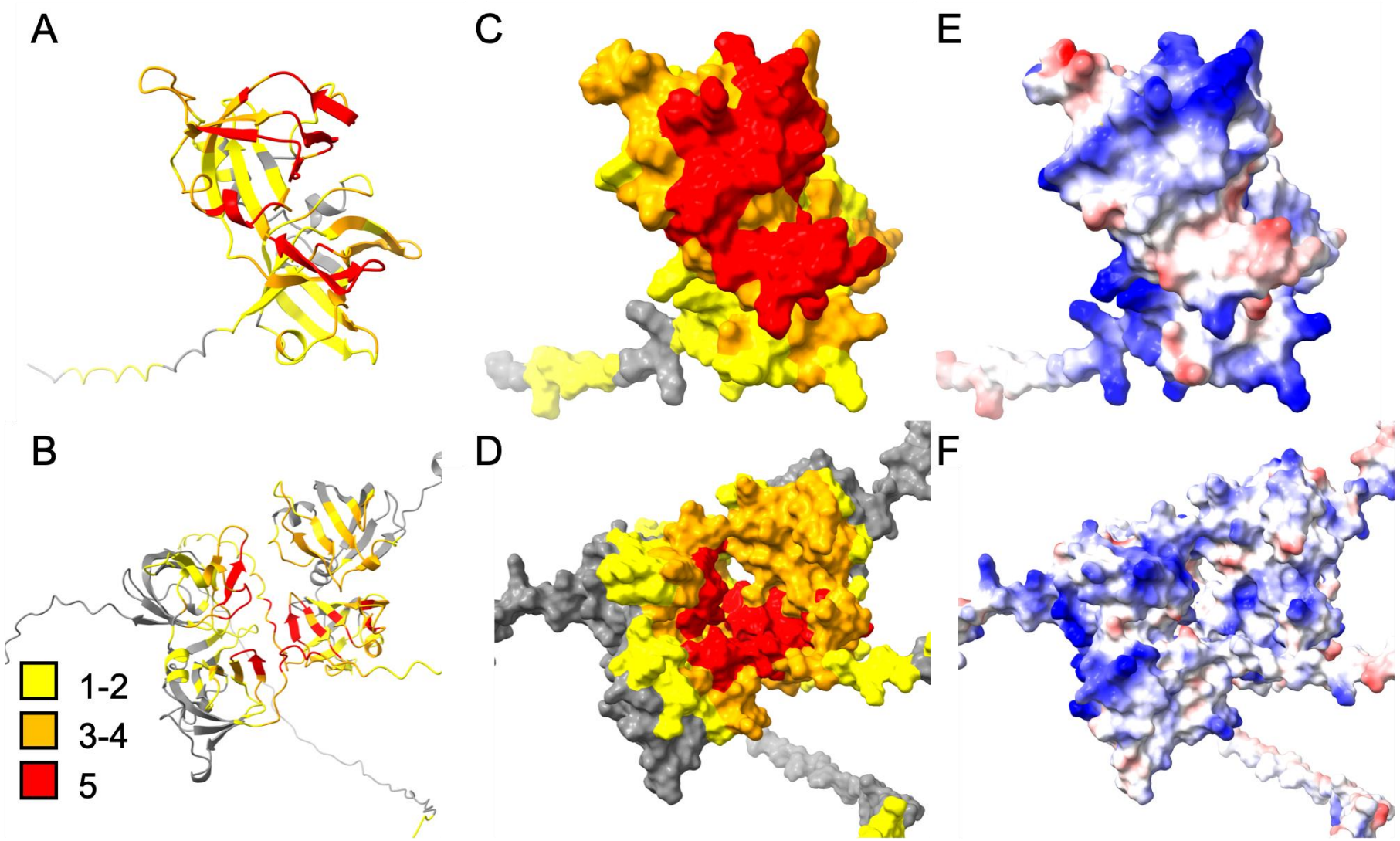
```
import numpy as np
import pandas as pd
from IFIAIM2contact import contacts
```

```
#%% Take a list of all pdb file names (minus .pdb) and run IFIAIM2contact.py on them all
path = 'multi_new/IFIAIM2_filenames.txt'
names = []
with open(path, 'r') as file:
    for line in file:
        clean = line.strip()
        names.append(clean)
for n in range (0,len(names)):
    contacts(names[n])
```

```
#%%
df = {}
dfn = {}
totalres = np.array([])
totalres1 = np.array([])
totalres2 = np.array([])
for i in range(0,len(names)):
    df[f'df_{i+1}'] = pd.read_csv(('multi_new/'+names[i]+'distances.csv')) #import all csvs into df
    dfn[f'df_{i+1}'] = pd.read_csv(('multi_new/'+names[i]+'.csv'))
    totalres = np.append(totalres,dfn[f'df_{i+1}']['Unique Residue']) #list of all important residues
    totalres1 = np.append(totalres1,df[f'df_{i+1}']['Residue1']) #all improtant residues from one protein
    totalres2 = np.append(totalres2,df[f'df_{i+1}']['Residue2']) #all from other
```

```
#%% count how many times each unique residue appears in total within all csvs uniquely
uniq1 = np.unique(totalres1) #unique residues in IFI
uniq2 = np.unique(totalres2) #unique residues in AIM2
count1 = np.array([])
count2 = np.array([])
for u in uniq1:
    count1 = np.append(count1, np.sum(totalres == u))
for u in uniq2:
    count2 = np.append(count2, np.sum(totalres == u))
```

```
#%% make dataframe of correct residue values and occurances
aim_uniq= uniq2-569
ifi_uniq = uniq1+1
data1 = {'Unique Residue IFI': ifi_uniq, 'Occurances': count1}
data2 = {'Unique Residue AIM2': aim_uniq, 'Occurances': count2}
uniq_count1 = pd.DataFrame(data1)
uniq_count2 = pd.DataFrame(data2)
```



Important residues in the interaction of AIM2 HIN and IFI16β. (A-B) A ribbon diagram and (C-D) surface model of AIM2 HIN and IFI16β, respectively. Important residues are colored by their occurances in all solved multimer structures as shown in the key. (E-F) Model of electrostatic surfaces for each protein. Structures from highest ranked colabfold-multimer model solved.

Conclusions

- Over 30 residues important in every predicted structure
- Electrostatic residues in notable regions
- IFI16β interacts using both HIN domains by creating a pocket for AIM2 HIN to interact with
- AIM2 HIN IFI16β interaction surface should not interfere with the dsDNA binding surface

Future Plans

- Confirm that the interaction of AIM2 and IFI16β does not inhibit dsDNA binding as the structure suggests
- Use molecular docking calculation software to determine the energetics of the predicted interaction and if it is realistic
- Analyze the interaction further through bulk molecular techniques and NMR

Questions?

Citations and Acknowledgements

- Mirdita, Milot, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. “ColabFold: Making Protein Folding Accessible to All.” *Nature Methods* 19, no. 6 (June 2022): 679–82. https://doi.org/10.1038/s41592-022-01488-1.
- Evans, Richard, Michael O’Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, et al. “Protein Complex Prediction with AlphaFold-Multimer.” bioRxiv, March 10, 2022. https://doi.org/10.1101/2021.10.04.463034.
- McGibbon, Robert T., Kyle A. Beauchamp, Matthew P. Harrigan, Christoph Klein, Jason M. Swails, Carlos X. Hernández, Christian R. Schwantes, Lee-Ping Wang, Thomas J. Lane, and Vijay S. Pande. “MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories.” *Biophysical Journal* 109, no. 8 (October 20, 2015): 1528–32. https://doi.org/10.1016/j.bpj.2015.08.015.
- https://github.com/dwhswenson/contact_map
- Google DeepMind. “AlphaFold: A Solution to a 50-Year-Old Grand Challenge in Biology,” November 30, 2020. https://deepmind.google/discover/blog/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology/.
- Molecular graphics and analyses performed with UCSF ChimeraX, developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco, with support from National Institutes of Health R01-GM129325 and the Office of Cyber Infrastructure and Computational Biology, National Institute of Allergy and Infectious Diseases.
- UCSF ChimeraX: Tools for structure building and analysis. Meng EC, Goddard TD, Pettersen EF, Couch GS, Pearson ZJ, Morris JH, Ferrin TE. Protein Sci. 2023 Nov;32(11):e4792.
- The UC Merced IT for putting up with my many requests for help with the computational cluster