Chapter 3: Computational Methodology

Table of Contents

3.1 Introduction 2

3.2 Structure prediction tools 2

3.2.1 Rosetta 2

3.2.2 Molecular Software Library 3

3.2.3 Docking Algorithms 4

3.2.4 Structure Prediction using Maching Learning 4

3.3 Protein Design Algorithm 5

3.3.1 Analysis of membrane protein PDBs 5

3.3.2 Defining the Interface 5

3.3.3 Developing the energy terms 5

3.3.4 Sequence Search 6

3.3.5 Backbone Refinement 6

3.4 Analysis 6

3.5 Conclusion 6

3.1 Introduction

* Talk about the importance of computation through my time in grad school, catching up, and how there are good tools to do so
* Reference papers that seem to do a good job mixing in computation with the biology simplifying for learners
* Emphasize the importance of computation due to high-throughput data analysis and AI

3.2 Structure prediction tools

In conjunction with structural determination, computational methods have been invented to evaluate previously solved MP structures. Many methods look to further understand MP folding by establishing energetic terms that aim to estimate the thermodynamics of association. Below I review some of the current computational methods that are used to predict protein structures.

3.2.1 Rosetta

David Baker’s Lab at University of Washington are experts in this area, utilizing known energetic and structural information to develop the software suite Rosetta. Rosetta houses a variety of energy functions and prediction tools for soluble environments, including the ability to dock or design proteins *de novo*. The original Rosetta energy functions have been adapted to accommodate predicting helical TMs within the membrane environment. The updated functions include an energy term that models the membrane into layers classified defining atoms as water-exposed, polar, interface, and hydrophobic (White & Wimley, 1999; Yarov-Yarovoy, Schonbrun, et al., 2006). Successful predictions determined structural details in MPs associated with voltage sensing and gating mechanisms (Vargas et al., 2012; Yarov-Yarovoy, Baker, et al., 2006). Recently, RosettaMP was developed to enhance the functionality of MP prediction which includes modeling TM helices *de novo* from sequence, a representation of the membrane bilayer, and the ability to transform a protein into the membrane (Koehler Leman et al., 2017). These tools increase the accessibility to MP structure, improving the ability to visualize and predict structures of MPs that have not yet been solved. Simultaneously, energetic predictions enhance the knowledge in the field by allowing researchers to assess structural mutations *in silico* prior to testing with *in vitro* or *in vivo* experiments.

3.2.2 Molecular Software Library

Another computational tool for membrane proteins is the Molecular Software Library, or MSL (Kulp et al., 2012). Similar to Rosetta, MSL houses a variety of tools to perform MP structure prediction. These tools include the ability to transform proteins in space, mutate specific residues, extract geometric information from known structures, and predict the structure and energetics of an input sequence. Additionally, it has been used to develop the CATM algorithm using two energy terms parameterized by CHARMM: hydrogen bonding and van der Waals packing. The CATM algorithm has been applied to predict the structures of both known and unknown TM dimers, and experimental studies have shown that it accurately predicts the energetics of association (Anderson et al., 2017; Díaz Vázquez et al., 2023; Mueller et al., 2014). My research adapts the CATM algorithm to design structures with strong van der Waals packing in the absence of hydrogen bonding to assess the extent at which packing can drive MP association. I further detail how I used MSL to design TM homodimers in section 3.3.

3.2.3 Docking Algorithms

Other methods for predicting interactions between MPs focus on predicting the topology or docking of individual TM helices. OCTOPUS predicts TM topology using a combination of Markov models and neural networks (Viklund & Elofsson, 2008). HADDOCK can apply experimental knowledge of the interface region between proteins to refine docking (de Vries et al., 2010; Dominguez et al., 2003). PREDDIMER utilizes a novel surface-based modeling approach to predict and screen TM dimers for conformation heterogeneity (Polyansky et al., 2012). EVFold uses evolutionary based structural restraints to refine their docked structures (Braun T et al., 2015). TMDOCK uses an all-atom model for helices, inserting them in the membrane and outputting a structure alongside a predicted ΔG of insertion and ΔG of association (Lomize & Pogozheva, 2017). Each of these methods was adapted into a webserver for online use, where users can input the sequence and additional information to guide the process.

3.2.4 Machine learning tools

Most notably, at the 2020 Critical Assessment of Structure Prediction (CASP) conference, Google’s Deepmind introduced the machine learning model AlphaFold. AlphaFold utilizes a combination of neural networks and training on multiple sequence alignments (MSAs) and solved protein structures to predict unknown structures to near atomic precision with a 95% confidence interval (Jumper et al., 2021). Shortly afterward, David Baker’s group introduced RoseTTAFold, improving on the Rosetta prediction by incorporating a similar architecture to AlphaFold, with the addition of a third track network that connects sequence, residue-residue distances, and atomic coordinates (Baek et al., 2021). Each of these methods drastically improved the ability to predict unknown protein structures using information from previously studied and solved proteins. With increasing interest in using these technologies, multiple free webservers have been established to enhance access to these advanced protein prediction algorithms (Mirdita et al., 2022; Roberts et al., 2024). However, there are limits to how well these prediction algorithms work. AlphaFold struggles to predict proteins with <30 homologs in their MSAs, and accuracy decreases for protein structures dependent on other protein-protein interactions, while RoseTTAFold has difficulty predicting higher-order oligomers (Agard et al., 2022). The disparity between soluble proteins and MPs in the PDB contributes to these limitations, making small TM proteins difficult to predict.

3.3 Protein Design Algorithm

To investigate the impact of van der Waals packing on membrane protein association, I opted for a high-throughput design approach. I developed an algorithm that designs protein sequences based on an input geometry and a corresponding interface. Using the tools available in MSL, I developed a sequence search algorithm coupled with structural backbone refinement. Below, I detail how the algorithm works alongside the experiments, tests, and the inspiration from other research that helped me develop it.

3.3.1 Analysis of membrane protein PDBs

Using MSL, I extracted backbone geometries from all unique MPs found in OPM (date). The list of MPs from PDB website was used to trim for sequence based on 30% sequence similarity, allowing me to search only non-redundant protein structures. Helices of length L are identified and the distance measured between each helix by their centroid (explain). Any two helices with a distance less than # are extracted as individual protein pairs and the geometric information measured. (explain more?) The geometry as well as the sequence information is then stored for analysis.

I compiled this geometric data and (how did I trim the data…?) I need to redo that part for my analysis actually, so might as well do that soon. The angle and xShift are plotted against each other for each protein and analyzed using kernel density estimation. (explain).

3.3.2 Defining the Interface

In my first protein design run, I implemented a method to identify the interface of the given geometry using Solvent Accessible Surface Area (SASA). I calculated the SASA for each position on the protein and defined the interface as the 8 positions with the least amount of access to the solvent, or the most buried positions. However, when after sorting my designs and looking at the results, the variations in the interface increased the complexity of the data (figure). In my future design runs, I standardized the interface for proteins designed in each region through visual inspection of a multitude of geometries (figure). The GASright and right designs used the same interface, while the left designs were composed of 3 unique interfaces to account for known motifs such as the leucine zipper and coiled-coil (cite). Standardizing the interfaces also allowed me to come up with a consistent mutational strategy to assess my proteins for their association at the given interface that is discussed in detail in section 3.3.6.

3.3.3 Developing the energy terms

To assess my designed proteins, I used the same set of energy terms used previously by my lab in CATM. These terms include a term that measures the van der Waals packing (CHARMM\_VDW), hydrogen bonding (SCWRL4\_HBOND), and implicit solvation of the membrane (CHARMM\_IMM1 and CHARMM\_IMM1REF). These terms are then used to assess the energetics of the dimer-monomer during our sequence design search to find the best sequence for the structure. However, computing the monomer energy for each sequence was initially found to create a bottleneck in our design algorithm. To account for this, I developed a new BASELINE\_MONOMER term that estimates the monomer energy of each sequence.

The BASELINE\_MONOMER term was created by first measuring the contributions from each individual amino acid. I ran 1000s of iterations of random sequences and measured the self and pair energies for each amino acid. The self energy is the energy contribution for each amino acid to the protein stability alone, while the pair energies are the energy contribution between any two interacting amino acids. From each of my iterations, I calculated the average of all self and pair energies and saw a strong correlation between the measured monomer energy and the BASELINE\_MONOMER term (figure). However, the BASELINE\_MONOMER term was only developed for a subset of amino acids that were used for sequence design and would need to be rerun to establish a term for any additional amino acids. This term was helpful in decreasing computational time.

Another issue that I encountered while designing my proteins was my designed proteins having too many of a single amino acid. This result could impact our protein expression, where designs with many of a single amino acid do not express similarly to designs with a variety of AAs. To combat this, I developed a SEQUENCE\_ENTROPY term that outputs a score based on how similar the output sequence is to a typical membrane protein sequence. This term was developed by using the sequences extracted from membrane protein PDBs and defining the frequency of each AA. The frequency is then used to … (add in equations here).

Each of these terms was used in our sequence search algorithm.

* Show the image of the comparison between distributions (both from extracted MPs and the liu 2002 paper)

3.3.4 Sequence Search

For each input geometry, the algorithm first defines the best sequence using a Self Consistent Mean Field (SCMF). Add more detail here…

The sequence from the SCMF is then run through a Monte Carlo sequence optimization: A random position on the interface is switched to a random AA from the given set of AAs, and the energy is calculated using the combination of defined energies in 3.3.3. This energy is used for the acceptance criteria, where early on in the MC more leeway is given on the search for the sequence with the energetic minimum. The sequences are saved in an output file to show the trajectory of x sequences that were chosen along with their corresponding energy terms.

3.3.5 Backbone Refinement

Another improvement from my initial design run was to use backbone iterative refinement, inspired by …Rosetta. After the best sequence is found, I implemented a MC based structural refinement procedure to find the best structure for the sequence. During the refinement process, a random geometric term is chosen and shifted by a given …. The energy for the altered structure is again used as the acceptance criteria, with the monomer energy being calculated prior to the procedure. As an option, this new geometry can then repeat the sequence search and backbone refinement process to determine if there are other sequences that are stable.

After my protein design for my second run, I also wanted to have a way to measure the energies of the chosen mutants for each of my structures. To do so, I created a more intensive structural refinement program. This program takes in the given structure and repeats the MC based structural refinement, but this time procedurally decreases the given … by the MC acceptance ratio. For example: As the geometric distance between helices is made shorter, the … is adjusted to allows for shorter distance movements to be made. This more intensive refinement can be repeated as many times as desired, and the energy output is what I used to evaluate my structures against their reconstructed fluorescence in sort-seq.

3.3.6 Mutating the interface

For each protein, I made point mutations on the interface corresponding to interfacial positions expected to decrease association. I created a program that accepts the structure and interfacial positions as inputs, individually mutates the positions to a given amino acid, and outputs the energy. The first mutation is a clash mutant, where the interface was mutated to an Ile to disrupt the ability for the protein to associate. The second mutation is a void mutant, where the interface was mutated to an Ala to assess the loss of steric packing on association. When comparing the clash mutations to the fluorescence of the designed structure, there is a significant decrease in their association, suggesting that many of my proteins are associating by the designed interface. However, the void mutations did not show this same decrease, suggesting that mutating for loss of packing may have instead resulted in the formation of an alternate interface for association.

3.4 Analysis

To analyze my design data, I created a program that compiles the data from all of the design runs and …

After running sort-seq, the sequencing data is run through a pipeline that reconstructs the fluorescence profile for each sequence found in the dataset. ….

* Python scripts built for ease of use and how to use them all detailed here, probably with images
  + ngsReconstruction
  + runAllAnalysis
  + toxgreenAnalyzer

3.5 Conclusion

* Talk about ways to improve the molecular software library and/or to transition some of these scripts to something more usable like Rosetta/Alphafold/how to make this easy to use
  + Creating a webserver was a pipe dream I had but never got the skill to do it

3.6 References

Agard, D. A., Bowman, G. R., DeGrado, W., Dokholyan, N. V., & Zhou, H. X. (2022). Solution of the protein structure prediction problem at last: crucial innovations and next frontiers. *Fac Rev*, *11*, 38. <https://doi.org/10.12703/r-01-0000020>

Anderson, S. M., Mueller, B. K., Lange, E. J., & Senes, A. (2017). Combination of Cα-H Hydrogen Bonds and van der Waals Packing Modulates the Stability of GxxxG-Mediated Dimers in Membranes. *J Am Chem Soc*, *139*(44), 15774-15783. <https://doi.org/10.1021/jacs.7b07505>

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., . . . Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, *373*(6557), 871-876. <https://doi.org/10.1126/science.abj8754>

Braun T, Koehler Leman J, & OF, L. (2015). **Combining Evolutionary Information and an Iterative Sampling Strategy for Accurate Protein Structure Prediction**. In: PLOS Computational Biology.

de Vries, S. J., van Dijk, M., & Bonvin, A. M. (2010). The HADDOCK web server for data-driven biomolecular docking. *Nat Protoc*, *5*(5), 883-897. <https://doi.org/10.1038/nprot.2010.32>

Dominguez, C., Boelens, R., & Bonvin, A. M. (2003). HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *J Am Chem Soc*, *125*(7), 1731-1737. <https://doi.org/10.1021/ja026939x>

Díaz Vázquez, G., Cui, Q., & Senes, A. (2023). Thermodynamic analysis of the GAS. *Biophys J*, *122*(1), 143-155. <https://doi.org/10.1016/j.bpj.2022.11.018>

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., . . . Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583-589. <https://doi.org/10.1038/s41586-021-03819-2>

Koehler Leman, J., Mueller, B. K., & Gray, J. J. (2017). Expanding the toolkit for membrane protein modeling in Rosetta. *Bioinformatics*, *33*(5), 754-756. <https://doi.org/10.1093/bioinformatics/btw716>

Kulp, D. W., Subramaniam, S., Donald, J. E., Hannigan, B. T., Mueller, B. K., Grigoryan, G., & Senes, A. (2012). Structural informatics, modeling, and design with an open-source Molecular Software Library (MSL). *J Comput Chem*, *33*(20), 1645-1661. <https://doi.org/10.1002/jcc.22968>

Lomize, A. L., & Pogozheva, I. D. (2017). TMDOCK: An Energy-Based Method for Modeling α-Helical Dimers in Membranes. *J Mol Biol*, *429*(3), 390-398. <https://doi.org/10.1016/j.jmb.2016.09.005>

Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., & Steinegger, M. (2022). ColabFold: making protein folding accessible to all. *Nat Methods*, *19*(6), 679-682. <https://doi.org/10.1038/s41592-022-01488-1>

Mueller, B. K., Subramaniam, S., & Senes, A. (2014). A frequent, GxxxG-mediated, transmembrane association motif is optimized for the formation of interhelical Cα-H hydrogen bonds. *Proc Natl Acad Sci U S A*, *111*(10), E888-895. <https://doi.org/10.1073/pnas.1319944111>

Polyansky, A. A., Volynsky, P. E., & Efremov, R. G. (2012). Multistate organization of transmembrane helical protein dimers governed by the host membrane. *J Am Chem Soc*, *134*(35), 14390-14400. <https://doi.org/10.1021/ja303483k>

Roberts, J. B., Nava, A. A., Pearson, A. N., Incha, M. R., Valencia, L. E., Ma, M., . . . Keasling, J. D. (2024). Foldy: An open-source web application for interactive protein structure analysis. *PLoS Comput Biol*, *20*(2), e1011171. <https://doi.org/10.1371/journal.pcbi.1011171>

Vargas, E., Yarov-Yarovoy, V., Khalili-Araghi, F., Catterall, W. A., Klein, M. L., Tarek, M., . . . Roux, B. (2012). An emerging consensus on voltage-dependent gating from computational modeling and molecular dynamics simulations. *J Gen Physiol*, *140*(6), 587-594. <https://doi.org/10.1085/jgp.201210873>

Viklund, H., & Elofsson, A. (2008). OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. *Bioinformatics*, *24*(15), 1662-1668. <https://doi.org/10.1093/bioinformatics/btn221>

White, S. H., & Wimley, W. C. (1999). Membrane protein folding and stability: physical principles. *Annu Rev Biophys Biomol Struct*, *28*, 319-365. <https://doi.org/10.1146/annurev.biophys.28.1.319>

Yarov-Yarovoy, V., Baker, D., & Catterall, W. A. (2006). Voltage sensor conformations in the open and closed states in ROSETTA structural models of K(+) channels. *Proc Natl Acad Sci U S A*, *103*(19), 7292-7297. <https://doi.org/10.1073/pnas.0602350103>

Yarov-Yarovoy, V., Schonbrun, J., & Baker, D. (2006). Multipass membrane protein structure prediction using Rosetta. *Proteins*, *62*(4), 1010-1025. <https://doi.org/10.1002/prot.20817>