Analysis of local neighbourhoods in proteins

विद्या वाचस्पति की
उपाधि की अपेक्षाओं की आंशिक पूर्ति में प्रस्तुत शोध प्रबंध

A thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy

द्वारा / By

मुकुंदन/ Mukundan S)

पंजीकरण सं. / Registration No.: 20193650

शोध प्रबंध पर्यवेक्षक / Thesis Supervisor:
Prof. M. S. Madhusudhan

भारतीय विज्ञान शिक्षा एवं अनुसंधान संस्थान पुणे
INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH PUNE
2025

CERTIFICATE

Certified that the work incorporated in the thesis entitled ("**Analysis of local neighbourhoods in proteins**") Submitted by **Mukundan S** was carried out by the candidate, under my supervision. The work presented here or any part of it has not been included in any other thesis submitted previously for the award of any degree or diploma from any other University or institution.

डॉ. एम्. एस. मधुसूदन/Dr. M. S. Madhusudhan
प्राध्यापक, जीवशास्त्र विभाग/Professor, Biology Department
भारतीय विज्ञान शिक्षा एवं अनुसंधान संस्थान
Indian Institute of Science Education & Research
पुणे / Pune- 411 008, भारत / India

Prof. M. S. Madhusudhan

Date: 14th January 2025

Declaration by Student

Name of Student: **Mukundan S**

Reg. No.: **20193650**

Thesis Supervisor(s): **Prof. M. S. Madhusudhan**

Department: **Biology**

Date of joining program: **1st Aug 2019**

Date of Pre-Synopsis Seminar: **13th Dec 2024**

Title of Thesis: **Analysis of local neighbourhoods in proteins**

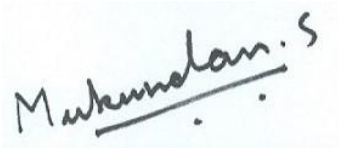
I declare that this written submission represents my idea in my own words and where others' ideas have been included; I have adequately cited and referenced the original sources. I declare that I have acknowledged collaborative work and discussions wherever such work has been included. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

The work reported in this thesis is the original work done by me under the guidance of

Dr./Prof. **M. S. Madhusudhan**

Date: **14.01.2025**

Signature of the student



Acknowledgements

Six years is a lot of time to spend on a degree. However, during these years, I believe that I have gained a wealth of valuable skills, lessons, and experience. Many people have facilitated this, and they deserve mention in this thesis.

First and foremost, I would like to thank my supervisor, Prof. M. S. Madhusudhan. He is a master at navigating the delicate balance between warmth and professionalism. His positivity is an optimist, an idealist, and a critical thinker. Moreover, he is a very good human being. It is unfair to ask for a better supervisor than him. He has been instrumental in shaping this thesis and my academic development. Over the years, he has built an excellent lab, which I was lucky to have joined.

I want to thank the members of this lab, both past and present. Neelesh, Neeladri, Sanjana, Tejashree and Atreyi were already part of the lab when I joined the lab. All of them played crucial roles in my academic life. Shourya and Somanath joined the lab recently, and I thank them for their many fun memories. I also thank all the numerous project students who joined that lab (special thanks to the ones who had the misfortune of working with me). The very dull task of proofreading the thesis was done by Asita, Deep, Atreyi and Somanath; thank you. Special thanks to all the undergraduates who worked with me, Kritika, Misal, Surya, Aditya, Mahima, Ruchir, Parth, Aneri, Deep and Medha for tolerating me.

Special thanks to Aparna Sinha the lab manager of the IISER Bioinformatics Center (of which our lab is a part of) who makes the mundane bureaucratic tasks significantly easier. If you are reading this, I would like to repeat that it was NOT I who threw dart at the monitor like you suspect. I cannot lie in this thesis (see the attached affidavit).

I am grateful to my research advisory committee members Dr. Gayathri Pananghat, Dr. Siddhesh S Kamat and Dr. Durba Sengupta for their feedback and rigorous evaluations.

I want to thank my parents, Asha G. and Shaju R., and family for their support during this duration. I would also like to thank my friends (in no specific order) Saunri, Karishma, Jashaswi, Kritika, Manisha, Kunal, Bijesh, Somya, Titir, Shivani, Somashree, Debadutta, Arjun, Vijay, SB and Milan for all their sunshine, kindness, and affection. I would also like to thank my batchmates for their support.

Table of Contents

Chapter 1: Overview

Synopsis

Proteins, through their activity and interactions, are the predominant class of molecules enabling the functioning of the living cell. They are linear polymers of amino acids that fold based on covalent and non-covalent interactions (Anfinsen 1973). The amino acids in proteins form local structural and chemical neighbourhoods responsible for stabilising the proteins and the myriad of functions they execute.

Analysing these local neighbourhoods should help us understand various aspects of proteins such as their stability, functionality, and structural similarity. To this end, we have formulated an approach to arrange the three-dimensional data that describes these neighbourhoods as a tree database. This database clusters and superimposes the local neighbourhoods from a database of protein structures based only on their spatial similarity. Moreover, the data is indexed in the process, i.e., arranged to make for efficient search, even for large databases.

We make use of this tree to create a topology-independent (sequence and sequence-order independent) structural database search program that is methodologically analogous to BLAST (Altschul et al. 1990). Our method mines structural data in a superior manner than state-of-the-art methods such as FoldSeek (van Kempen et al. 2024) or TM-Align (Zhang and Skolnick 2005). We demonstrate this by an example query consisting of pockets of discontiguous small peptide fragments involved in binding phosphoinositide. We retrieve multiple high-quality structural matches using such queries, while FoldSeek and TM-align return zero matches. We can achieve this because our method only evaluates the similarity in the spatial arrangement of atoms.

One of the applications of the tree database described above is to engineer proteins where we attempt to engineer new functions into existing proteins. We aim to achieve this by creating a program that transfers the local chemical and structural neighbourhoods responsible for the function via predicting mutations. Our program first checks if the structural motifs required for the function exist in the query protein. It then tries to reproduce the chemical environment (interaction and catalytic residues) required for the function. Using the green fluorescence function from Green fluorescent protein (Shimomura, Johnson, and Saiga 1962) as our function of interest, we searched the 200199 structures from the Protein Data Bank (Berman et al. 2002) and found 12573 hits. Each of these outputs consists of a mutational library, where a

subset of the mutation would make the protein fluorescent. We filtered and selected three out of the 12573 hits for experiments based on size, number of mutations and MD simulations. The experiments on two of the tree hits show no fluorescence; the experiments on the third protein are underway.

During the process of protein engineering, we have also developed a dynamic programming approach to predict the optimal set of overhangs for efficient and accurate DNA assembly by the Golden Gate approach (Engler, Kandzia, and Marillonnet 2008; Pryor et al. 2020). The Golden Gate approach assembles DNA fragments in an order specified by four base pair overhangs. However, all overhangs are not equally efficient nor accurate (fidelity) (Potapov et al. 2018). Thus, predicting high efficiency and high fidelity overhangs (fragmentation sites) in a DNA sequence is crucial for creating large mutational libraries. Our method simultaneously optimises number of fragments and the fragmentation sites for with that for both efficiency and fidelity of assembly. We then benchmarked our method against NEB SplitSet (Pryor et al. 2022), a popular method to predict overhangs which only optimises fidelity. Our method outperforms SplitSet when only optimising fidelity. We match SplitSet on fidelity and outperforms SplitSet on efficiency when optimising both fidelity and efficiency.

In another study, we analysed the physical chemistry of biomolecular association reactions in situations where reactants in the system share such local environments involved in the interactions. We specifically explored cases where one such reactant has comparatively weaker binding strength. The strength of molecular interactions is characterised by their dissociation constants (K_D). Only high-affinity interactions ($K_D \leq 1e{-}8$ M) are extensively investigated and support binary on/off switches. However, such analyses have discounted the presence of low-affinity binders ($K_D > 1e{-}5$ M) in the cellular environment that might share similar local environments with a high-affinity interactor. We assess the potential influence of such low-affinity binders on high-affinity interactions. By employing Gillespie stochastic simulations and continuous methods, we demonstrate that the presence of low-affinity binders can alter the kinetics and the steady state of high-affinity interactions. We refer to this effect as ‘herd regulation’ and have evaluated its possible impact in two different contexts, including sex determination in Drosophila melanogaster and modelling signalling systems that employ molecular thresholds. We have also suggested experiments to

validate herd regulation in vitro. We speculate that low-affinity binders are prevalent in biological contexts where the outcomes depend on molecular thresholds impacting homoeostatic regulation.

Thesis organisation

Chapter 1: Overview

This chapter provides a synopsis of the entire thesis and a thesis organisation.

Chapter 2: Tree data structure to index local similarities of 3D coordinates

In this chapter, we discovered a tree data structure that clusters, superimposes, and indexes discontinuous local structural motifs. In addition, we describe a program that we created that uses the tree to efficiently search large structural databases based on cartesian coordinates of a query protein or a discontinuous fragment of a protein.

Chapter 3: Engineering functional chemical neighbourhoods into existing protein structures

In this chapter, we have created a program that grafts the function of one protein to another. This program was used to search the PDB database using the green fluorescence of GFP to predict 12573 mutational libraries. We are testing these libraries experimentally.

Chapter 4: Overhang Optimizer for Golden Gate Assembly (OOGGA)

This chapter describes a dynamic programming algorithm that we discovered that predicts the optimal set of overhangs for Golden Gate assembly. We outperformed the current state-of-the-art method NEB SplitSet in terms of both accuracy and efficiency of assembly.

Chapter 5: High-affinity biomolecular interactions are modulated by low-affinity binders

In this chapter, we model the effect of low-affinity binders ($K_D > 10^{-5}$) on high-affinity biomolecular interaction. We found that low-affinity binders likely play crucial roles in biological systems. We then use these models to mechanistically explain sex determination in *Drosophila melanogaster* and binary switches in biological systems.

Chapter 6: Conclusions and Future Perspectives

This concluding chapter discusses the future perspectives of the various studies listed in the thesis.

References

Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. 1990. “Basic Local Alignment Search Tool.” *Journal of Molecular Biology* 215(3):403–10. doi: 10.1016/S0022-2836(05)80360-2.

Anfinsen, Christian B. 1973. “Principles That Govern the Folding of Protein Chains.” *Science* 181(4096).

Berman, Helen M., Tammy Battistuz, T. N. Bhat, Wolfgang F. Bluhm, Philip E. Bourne, Kyle Burkhardt, Zukang Feng, Gary L. Gilliland, Lisa Iype, Shri Jain, Phoebe Fagan, Jessica Marvin, David Padilla, Veerasamy Ravichandran, Bohdan Schneider, Narmada Thanki, Helge Weissig, John D. Westbrook, and Christine Zardecki. 2002. “The Protein Data Bank.” *Acta Crystallographica Section D: Biological Crystallography*. doi: 10.1107/S0907444902003451.

Engler, Carola, Romy Kandzia, and Sylvestre Marillonnet. 2008. “A One Pot, One Step, Precision Cloning Method with High Throughput Capability.” *PLOS ONE* 3(11):e3647.

van Kempen, Michel, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. 2024. “Fast and Accurate Protein Structure Search with Foldseek.” *Nature Biotechnology* 42(2):243–46. doi: 10.1038/s41587-023-01773-0.

Potapov, Vladimir, Jennifer L. Ong, Rebecca B. Kucera, Bradley W. Langhorst, Katharina Bilotti, John M. Pryor, Eric J. Cantor, Barry Canton, Thomas F. Knight, Thomas C. Evans, and Gregory J. S. Lohman. 2018. “Comprehensive Profiling of Four Base Overhang Ligation Fidelity by T4 DNA Ligase and Application to DNA Assembly.” *ACS Synthetic Biology* 7(11):2665–74. doi: 10.1021/acssynbio.8b00333.

Pryor, John M., Vladimir Potapov, Katharina Bilotti, Nilisha Pokhrel, and Gregory J. S. Lohman. 2022. “Rapid 40 Kb Genome Construction from 52 Parts through Data-Optimized Assembly Design.” *ACS Synthetic Biology* 11(6):2036–42. doi: 10.1021/acssynbio.1c00525.

Pryor, John M., Vladimir Potapov, Rebecca B. Kucera, Katharina Bilotti, Eric J. Cantor, and Gregory J. S. Lohman. 2020. “Enabling One-Pot Golden Gate Assemblies of Unprecedented Complexity Using Data-Optimized Assembly Design.” *PLOS ONE* 15(9):e0238592.

Shimomura, Osamu, Frank H. Johnson, and Yo Saiga. 1962. “Extraction, Purification and Properties of Aequorin, a Bioluminescent Protein from the Luminous Hydromedusan,

Aequorea." *Journal of Cellular and Comparative Physiology* 59(3):223–39. doi: https://doi.org/10.1002/jcp.1030590302.

Zhang, Yang, and Jeffrey Skolnick. 2005. "TM-Align: A Protein Structure Alignment Algorithm Based on the TM-Score." *Nucleic Acids Research* 33(7):2302–9. doi: 10.1093/nar/gki524.

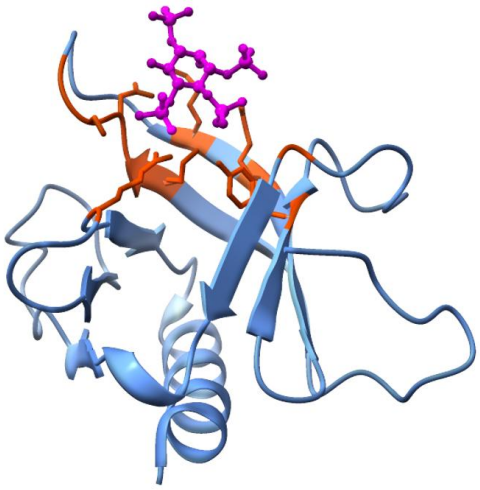
Chapter 2: Tree data structure to index local similarities of 3D coordinates.

Introduction

Biological systems rely on complex molecular pathways to carry out various functions. Most of these functions are executed by proteins. Proteins are polymers of amino acids that fold on themselves based on various non-covalent interactions to form specific structures (Anfinsen 1973). This process of protein folding creates a three-dimensional shape called a protein structure. This process of folding also brings various amino acids in proximity that are far away in their sequential order (amino acid sequence).

Many of these three-dimensional ‘neighbourhoods’ are responsible for various functions of proteins. Such neighbourhoods can be represented as collections of residues and can be considered as ‘structural motifs’. Such a neighbourhood that binds to a phosphoinositide is shown in Figure 2.1 (orange residues). Another protein that shares such structural motifs might share its phosphoinositide binding function. Searching a sequence database is irrational due to the discontinuous nature of the motif. Such applications require searching for structural motifs.

Existing structure search approaches such as TM-align and FoldSeek are designed to search larger protein structures (van Kempen et al. 2024; Zhang and Skolnick 2005). However, they fail to search for small discontiguous structural motifs similar to those shown in Figure 2.1. Exhaustive superimposition of the query to all neighbourhoods in a database is one approach to solve this problem. However, such operations are computationally prohibitive as they involve many pairwise superimpositions. CLICK solves this with a heuristic approach that searches a database primarily based on 3D coordinates by starting with small three-membered neighbourhoods and iteratively expanding to 7 members (M N Nguyen, Tan, and Madhusudhan 2011). However, CLICK can only perform pairwise superimpositions, and database searches remain computationally expensive. In this chapter, we suggest a solution to this problem by formulating a method to cluster, index and superimpose three dimensional coordinates.



PDB:1upr

NALRRDPNLPVHIRGWLHKQDSSGLRLWKRRWFVLSGHCLF
YYKDSREESVLGSVLLPSYNIRPDGPGAPRGRRFTFTAEHP
GMRTYVLAADTLEDLRGWLRALGRASRAEGDDYGQPRSPAR

Figure 2.1: Structure of pleckstrin homology domain (PDB id 1UPR) bound to phosphoinositide. The structure of the protein-ligand complex is represented in ribbons (protein) and ball and stick (Phosphoinositide ligand). The single-letter amino acid sequence of the protein is shown on the right side. The residues that interact with the phosphoinositide molecule are shown in orange in both the structure and the sequence.

Pairwise alignment of proteins is analogous to sequence pairwise alignment programs like the Needleman-Wunsch algorithm (Needleman and Wunsch 1970). Sequence database search algorithms such as BLAST can query millions of sequences in minutes. (Altschul et al. 1990). The speed of these algorithms can be attributed to various methods to index sequences, such as tries and suffix trees (De La Briandais 1959). An analogous approach to index for 3-dimensional data does not exist. This chapter describes a tree data structure to cluster and index 3D coordinates that is similar in functionality to a prefix tree.

Methods

Dataset curation

Protein structures in PDB format were downloaded from the RCSB Protein Data Bank (Berman et al. 2000). Lists of non-redundant chains from this dataset were obtained from PISCES webserver (Wang and Dunbrack 2003). We used three levels of non-redundancy: 15%, 30%, and 90% sequence identities. The parameters are shown in Table 2.1

Table 2.1: Parameters used to generate different subsets of PDB.

Maximum sequence identity	Minimum resolution	Minimum length	Maximum length	R-value cutoff	Date of retrieval	Total number of chains
15	2.5 Å	40	10000	0.3	14/03/2022	3988
30	2.5 Å	40	10000	0.3	14/03/2022	12079
90	2.5 Å	40	10000	0.3	14/03/2022	28745

Cliques: Group of associated atoms

Proteins are first deconvoluted into structural motifs called cliques. A clique is defined for each atom as its nearest N neighbours. The atom around which a clique is defined is called its central atom. The following steps are used to generate cliques from a PDB file.

1) PDB files from the curated datasets are parsed into coordinates, atom names, residue names, residue numbers, etc.
2) The coordinates are filtered for a set of atom types specific to the application.
3) A KD-tree is constructed from the coordinates using the KD-Tree module from SciPy (Virtanen et al. 2020). The 'query' function was used to find the nearest N neighbours for each coordinate. (The value of N is specified in respective sections)
4) This set of coordinates, along with its atom type, forms a clique.

An illustration of this process is shown in Figure 2.2. A part of the structure with PDB ID 1tig is first deconstructed into a set of C^{α} atoms. Cliques of atoms were then generated and are represented by red, blue, and yellow circles.

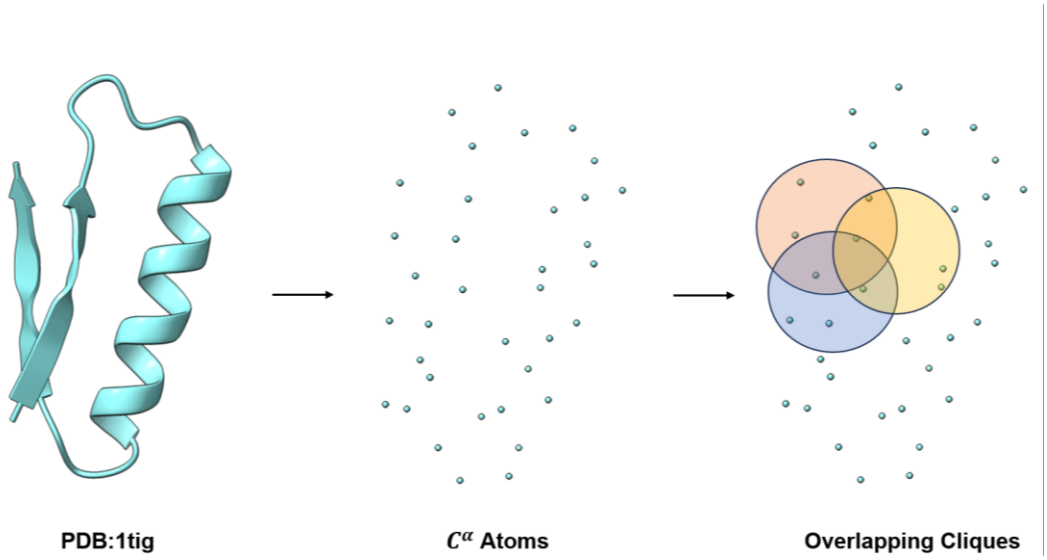


Figure 1.2: Example of formation of 5-membered cliques consisting of C^{α} atoms. The structure of a section of translational initiation factor IF3 is shown in blue ribbon diagram. The C^{α} atoms are shown in sphere representation. The red, blue and yellow coloured circles show three overlapping cliques containing five C^{α} atoms each.

In our study, we have used C^{α} and C^{β} atoms to form cliques. However, other combinations of atoms, including virtual coordinates (such as side chain centroid) representing interactions, can be used as long as they are specified using Cartesian coordinates assigned a type.

Hierarchical arrangement of structural motifs forms a tree

The tree we describe here is similar in arrangement to a sequence prefix tree. (De La Briandais 1959). The tree starts at the root and branches at points called nodes. Each node stores a structural motif. In the case of a structure motif, the prefix is a smaller motif containing one less atom. Based on this, nodes branch into child nodes representing distinct positions of the additional atom. This is similar to how nodes in a prefix tree branch into child nodes based on additional characters. The following sections explain various aspects and operations of the tree in more detail.

Attributes of the tree

The tree consists of nodes representing a category of cliques demarcated by its 3D shape. Each node can branch into multiple other nodes called child nodes. The category represented by the parent node is smaller by one atom, and the child node is one atom larger. The number of atoms represented by a node is called the level of the node. The set of coordinates that represent the node is its node clique. Since nodes represent all cliques that fall in the category, every node represents the subset

of cliques of its parent node and the superset of the child nodes. The information contained in the tree is stored in the following tables.

1) *cords*: Stores information of all the coordinates from the database.
2) *clique*: Stores a set of coordinate IDs that are part of the clique.
3) *nodes*: Information about nodes, i.e., its parent, children, level, and the number of cliques in this category.
4) *node_cliques*: Similar to the clique table (point 2), but stores cliques that describe the structural category that the node represents.
5) *node_cords*: Stores the coordinates that are part of node cliques.
6) *membership*: Stores the mapping between nodes and cliques. It also stores the order of coordinate IDs that match the node_clique (order of equivalences).
7) *info_id*: Stores parameters and inputs used for creating the tree.

Superimposing two cliques

Consider two cliques, A and B, consisting of atoms $A_1 \ldots A_n \ldots A_N$ and $B_1 \ldots B_n \ldots B_N$. Superimposition aims to find pairs of atoms (of the same type) between A and B such that the Euclidean distance between all the pairs is below a cutoff (SO_d). This implies that 100% of atoms are overlapping within SO_d. Let the set of such pairwise equivalences be ε. Since the difference between one node and its parent node is always one, at every level, we must only check if the additional atom superimposes with the existing superimposition. For this, one of the following three methods is used to evaluate the similarity of a clique to that of a node clique (clique representation of a node) depending on the level of the node.

At level 1, only one coordinate needs to be superimposed. For this, the atom type of the central atom is matched to the central atom of the node clique, i.e.

$$type(A_1) = type(B_1)$$

Note that this criterion is evaluated in all cases, even when the level is >1.

At level 2, it is implied that the pair of central atoms (A_1, B_1), are already superimposed. We now need to find out which pairs (A_x, B_y) for $x, y \in [2, N]$ has the least difference in Euclidian distance (d) to the central atoms, i.e.

$$argmin_{(A_x, B_y)} |d(A_1, A_x) - d(B_1, B_y)|$$

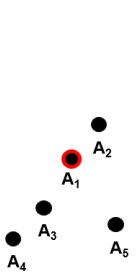
At levels (l) greater than two, $l-1$ pairs are already superimposed, and we only must find the l^{th} pair. The previously superimposed atoms, along with each possible pair are fitted using the least square fit in 3 dimensions. (Kearsley 1989), which analytically provides a rotation and translation matrix that orthogonally transforms B to A such that the root mean square deviation ($RMSD$) is minimised. The pair with the lowest $RMSD$ and 100% overlap is selected.

$$argmin_{(A_x,B_y)}\, RMSD\big(\varepsilon + (A_x, B_y)\big)\; where\; SO\big(\varepsilon + (A_x, B_y)\big) = 100$$

$$SO(E) = \frac{100}{|E|} \sum_{(a,b)\,\in E} \begin{cases} 1 & if\ d(a,b) \leq SO_d \\ 0 & otherwise \end{cases}$$

Indexing cliques in a tree (insertion)

The tree is initiated with a 'root' node. Consider two cliques A and B to be indexed in the tree. We start by indexing A. Since there are no nodes in the tree, the central atom of A forms the first node (Figure 2.3). However, each node needs at least two cliques to infer structural similarity. Since there is only one clique (A), we create a temporary 'hanging node'. This is shown under the column 'tmp' in the nodes table of Figure 2.3. The node clique of the temporary node contains all the coordinates in the clique. This is because we do not know which atom matches the next clique. We also cannot create any additional nodes for the same reason.



Root(0)

1

Membership

Node id	Clique id	Permutation

Nodes

Node id	tmp	Parent	Children	level	N
1	T	0		1	0

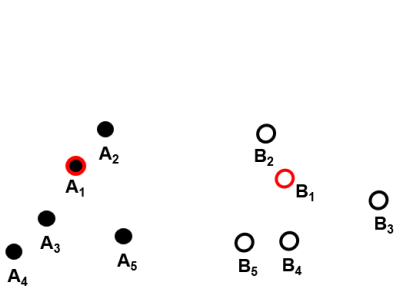
Node cliques

Node id	coordinates
1	A1,A2,A3,A4,A5

Figure 2.2: Formation of the first node as a temporary node from the clique A. The top left shows a representation of clique A in solid circles. The central group is highlighted in red. A representation of the tree with one node is shown at the top centre. Tables titled Membership, Nodes and Node cliques store information that describes the node. The membership table is empty in this case, as there is only one clique.

We now start the process of indexing clique B. We start at the root and reach node 1 (Figure 2.4). We check for similarities between cliques B and node 1 at level 1. They are similar in this example; consequently, node 1 has two matching cliques. Based on this similarity, we do the following

1) We change node 1 from temporary to permanent.
2) We modify the node clique to now contain only A_1(the matching coordinate).
3) We make two entries in the membership table storing the atom(s) that superimpose at the level from clique A and B.
4) We increment the number of cliques that match the node (population) (N).



Root(0)
1
2

Membership

Node id	Clique id	Permutation
1	1	A1
1	2	B1

Nodes

Node id	tmp	Parent	Children	level	N
1	F	0	2	1	2
2	T	1		2	0

Node cliques

Node id	coordinates
1	A1
2	A1,A2,A3,A4,A5

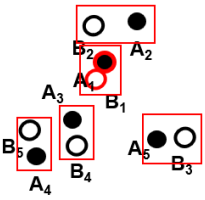
Figure 2.3: Conversion of node 1 and formation of node 2. The top left shows a representation of cliques A in solid circles and B in empty circles. The central groups are highlighted in red. A representation of the tree with node 1 and its child node 2 (a temporary node) is shown at the top centre. Tables titled Membership, Nodes and Node cliques store information that describes the node.

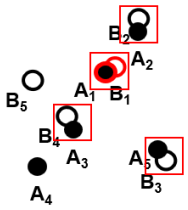
Since there are unindexed atoms in A (A_4), we have created a new hanging node (node 2) containing the remaining atoms (Figure 2.4Figure 2.3). This process is repeated till all the atoms in A and B are exhausted or till hanging nodes are formed.

However, there can be a state in which adding new pairs leads to non-superimposable structures. An example of such a case manifesting in A and B is shown in Figure 2.5Figure 2.4. The superimposition of A_4 to B_5 leads to Euclidian distances between the pairs of atoms between A and B that violate SO_d. In such cases A_4 and B_5 are used to create two separate temporary nodes (5 and 6) branching from node 4 (Figure 2.5Figure 2.4). This represents the divergence of the shape of the structural motifs beyond four atoms.

In addition, notice that at node 3, A_3 superimposes with B_4 (Figure 2.5Figure 2.4, table Membership). Please note that the clusters obtained from the tree are independent of the residues' position on the sequence, making them topology-independent. This is because the superimpositions only depend on the coordinates of the atom and its type.

The node clique coordinates are derived from the first clique that creates the node. In our example, nodes 1-5 are derived from clique A and node 6 is derived from clique B.





Membership

Node id	Clique id	Permutation
1	1	A1
1	2	B1
2	1	A1,A2
2	2	B1,B2
3	1	A1,A2,A3
3	2	B1,B2,B4
4	1	A1,A2,A3,A5
4	2	B1,B2,B4,B3

Nodes

Node id	tmp	Parent	Children	level	N
1	F	0	2	1	2
2	F	1	3	2	2
3	F	2	4	3	2
4	F	3	5	4	2
5	T	4		5	0
6	T	4		5	0

Node cliques

Node id	coordinates
1	A1
2	A1,A2
3	A1,A2,A3
4	A1,A2,A3,A5
5	A1,A2,A3,A5,A4
6	B1,B2,B4,B3,B5

Figure 2.4: Superimpositions are obtained until all atoms stop overlapping. Clique A is represented with solid circles and clique B with empty circles with the central groups highlighted in red. The top left shows cutoff breaking superimposition of all five atoms between the clique. The bottom left shows the optimal superimposition of A and B when only four atoms are superimposed, the red boxes show the equivalences used for superimposition. The representation of the complete tree formed from both cliques is shown at the top centre. Tables titled Membership, Nodes and Node cliques store information that describes the node.

Pseudocode that described the process of indexing

```
def index_clique(node,clique)
        level = get_node_level(node) # read from table
        children = get_children(node) # read child nodes from table
        best_child, best_atom_order = ‘’,‘’
        best_rmsd = inf
        for child in children:
                child_clique = get_node_clique(child) # read from table
                atom_order,rmsd = superimpose_cliques(clique,child_clique,level):
                        if rmsd < best_rmsd
                                best_child, best_atom_order = child,atom_order
        if best_child: # if there is a matching node
                if is_node_hanging(best_child) # read from table
                        convert_node(best_child,best_atom_order)
                add_to_tables(clique,best_child,best_atom_order)
                fit_node(best_child,clique)
        else:
                hang_the_child(clique) #create a child node, which is a temporary hanging node
index_clique(root,clique) # starting the search recursion
```

This process can be repeated using millions of cliques derived from the curated databases to form larger trees.

Querying a tree with a clique (search)

An existing tree can be queried with a query clique.

A pseudo-code that explains the algorithm for doing the same is provided below.

```
def fit_node(node,query_clique)
        level = get_node_level(node) # read from table
        children = get_children(node) # read child nodes from table
        fitting_childen = []
        for child in children:
                child_clique = get_node_clique(child) # read from table
                if superimpose_cliques(query_clique,child_clique,level):
                        fitting_childen.append(child)
        for child in fitting_childen:
                fit_node(child,query_clique)

fit_node(root,query_clique) # starting the search recursion
```

An example of this process using the tree made in the previous section and a query clique Q is shown in Figure 2.6Figure 2.5. The query superimposes with the node cliques of nodes 1-4 but fails to fit to either node 5 or 6. Since we know how cliques A and B map to nodes 1-4, and we can infer the mapping(equivalences) of the query Q to cliques A and B directly from the 'Membership' table and can form multiple structure alignments of the cliques.

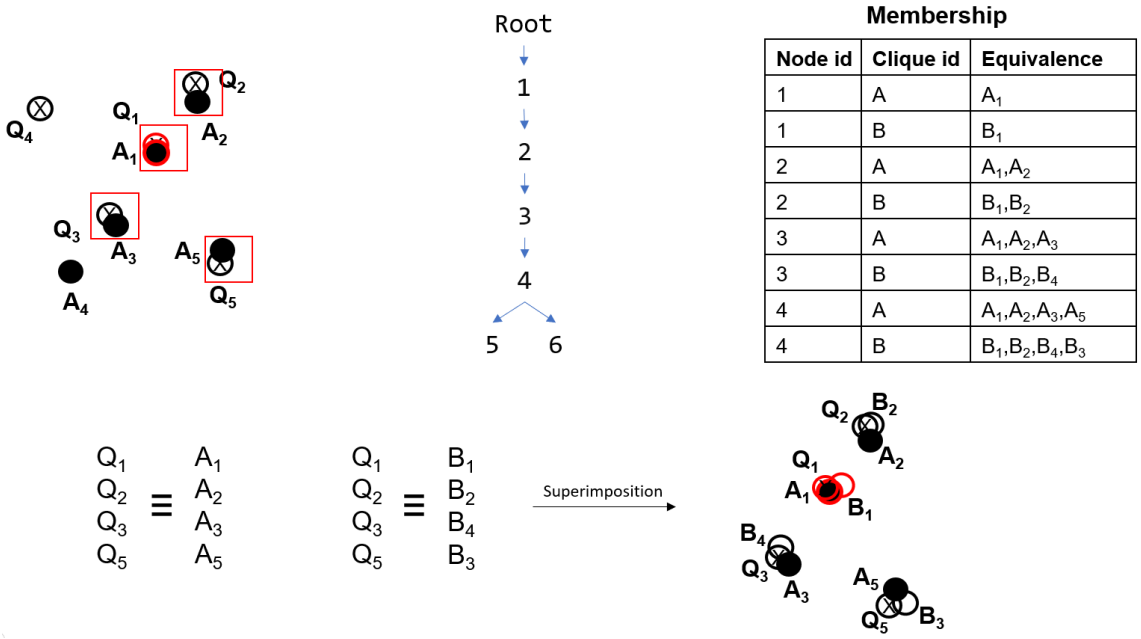


Figure 2.5: Querying tree with clique Q. Clique A is represented using solid black spheres; the crossed-out spheres show the query clique Q. The central atom of the clique is represented with a red highlight. Top left represents the superimposition of the query clique to clique A which represents nodes 1-5. The red box shows the atoms that superimpose to the tree. The bottom left shows the equivalence derived from the membership table for query clique Q to clique Q and B based on the superimposition to the tree. The bottom right shows the superimposition of the clique Q on A and B forming a multiple-structure alignment of cliques.

Thus, the tree enables us to match a query clique to all similar structural motifs/neighbourhoods from the indexed database simply by matching it to a node in the tree.

Implementation of the tree

The algorithms to index and search the tree are implemented in python3 along with numpy, scipy, numba packages (Harris et al. 2020; Lam, Pitrou, and Seibert 2015; Virtanen et al. 2020). The tables that form the tree data structure are implemented as part of a MariaDB SQL database. Our implementation takes approximately 20 minutes, 1 hour and 3 hours to index the NR15, NR30 and NR90 databases, respectively. Searching the tree with a clique takes approximately 100 milliseconds. Python scripts for the same can be made available on demand. Please note that this method is actively being developed.

Proteins Coordinate Basic Local Alignment Search Tool (ProCoBLAST)

Our protein structure database search approach is analogous to BLAST (Altschul et al. 1990). We first deconvolute a query into cliques (similar to sequence words in

BLAST) (Figure 2.7Figure 2.6 Step 1). We then query the cliques to a tree made with similar types of cliques of the same atom types and sizes (Figure 2.7 Step 2). This process identifies nodes from the tree that are similar to the query cliques. Since each node represents a collection of similar cliques from a database, we now have equivalences that map the query clique to various structures in the database.

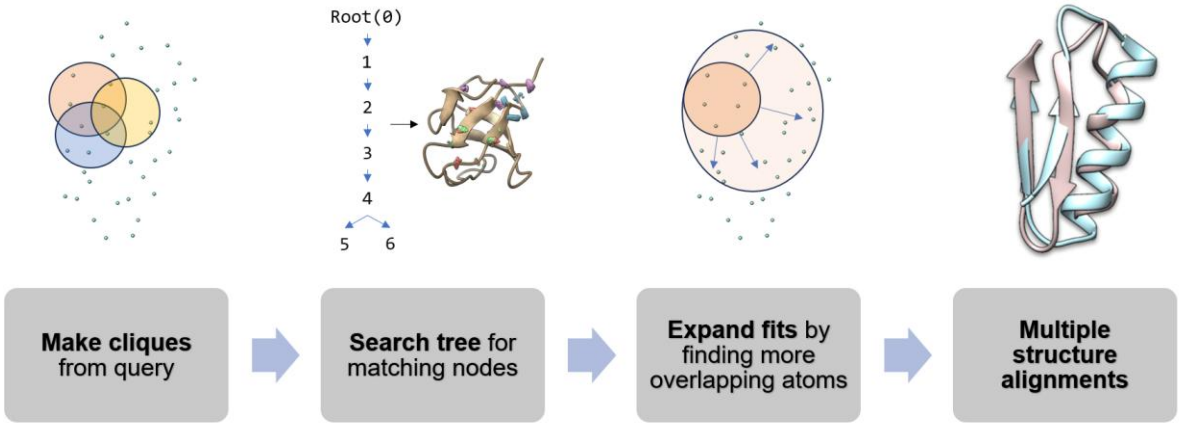


Figure 2.6: ProCoBLAST workflow. Step 1: Cliques are made from the query structure. Step 2: Structurally matching nodes are identified from the tree. Step 3: The matching cliques are expanded to identify larger sections of the match. Step 4: Report multiple structure alignments after rank ordering the hits.

We then quantify the percentage of atoms in the query with a unique counterpart in the database structure. To do this, we compute the superset of unique mappings based on all clique matches. A tentative query coverage in percentage is computed from this mapping. A cutoff is then used to filter structures from that database where a significant portion does not match. Here, we assume that the similarity of the superset implies the similarity of subsets (or at least a large percentage). This process ensures that only significant matches of structural motifs are used for further processing (analogous to BLAST), saving computing time.

Each of the shortlisted structures is first superimposed using every identified clique equivalences independently (Figure 2.7 Step 3). The set of equivalence in each case is then expanded (see next section) based on the overlapping atoms to identify hits from the database. A non-redundant list of equivalences is reported after culling all the equivalences obtained from each matching clique. TM-scores (Zhang and Skolnick 2004), GDT scores (Zemla 2003), RMSD, percentage coverage of query and template, and structure overlap is calculated for each hit. A double sort of coverage of the query and RMSD is used to rank order the hits. Query superimposed on the database

structure and vice versa are reported in PDB format (Berman et al. 2000) along with files containing a list of equivalent atoms and various matrices of superimposition(Figure 2.7 Step 4).

Algorithm to expand equivalences

Let us consider the case where a clique Q from the query structure S_Q maps to a clique A structure S_{DB} from the database based on the initial equivalence ε_i. Let the set of all coordinates in S_Q be C_Q and that in S_{DB} be C_{DB}. The method aims to obtain a final set of equivalence ε_f by adding additional superimposition pairs of atoms from C_Q and C_{DB}. We iteratively add pairs of overlapping atoms until the superimposition breaks a GDT or structure overlap (SO) cutoff to achieve this.

The following pseudocode describes our approach

```
# set a temporary equivalence as initial equivalence
```

ε_f = ε_i

N_ε = $|\varepsilon_f|$

gdt_d = range($d_{overlap}$,0,-0.5)

```
while True:
    # superimpose the atoms of the database structure to the query
```

$C_{DB\to Q}$ = superimpose($C_Q, C_{DB}, \varepsilon_f$)

SO = strucutre_overlap($C_Q, C_{DB\to Q}, d_{overlap}$) # compute structure overlap

GDT = GDT($C_Q, C_{DB\to Q}$, gdt_d)

if GDT < GDT_cutoff or SO < SO_cutoff or if N_ε == $|\varepsilon_f|$: # if the fit violates cutoffs stop iteration

```
        break
```

ε_f = find_more_overlapping_atoms($C_Q, C_{DB\to Q}, d_{overlap}$)

N_ε = $|\varepsilon_f|$

Here, superimpositions are obtained using Kearsley's 3D least square fit method (Kearsley 1989), SO_d is the lower limit of Euclidean distance for two atoms to be considered overlapping. $d_{overlap}$ is user-definable, a value in the range 1.5-3Å is suitable for searches based on C^α and C^β atoms. *SO_cutoff* is set at 100%. *GDT_cutoff* is user-defined, and values range between 0%-100%.

Results

Optimization of parameters

The Euclidian distance cutoff (SO_d), which is used to calculate the overlap between the atoms, and the total number of atoms (N) in a clique are optimised in this section. Databases were made for the NR15 subset of PDB at various distance cutoffs for cliques nearest 20 C^α and C^β atoms. One hundred most populous nodes were identified for each level above level 3 (the level after which the cutoff is required). All the nodes are selected if the number of nodes at the level is below 100. All the members of the node were then superimposed to the node representative. The Root Mean Square Fluctuation (RMSF) was calculated for each atom in the node representative, considering the superimposed member cliques as their alternate positions. The RMSF values for all such positions for every one of 100 nodes is plotted using box plots in Figure 2.8 across SO_d values from 0.5Å to 3Å at intervals of 0.25Å.

As expected, we see that the lower the cutoff, the lower the RMSF. Interestingly, level 3 has low RMSF regardless of the SO_d. We speculate that this is because 3-membered superimpositions formed at level 3 almost exclusively consist of a C^α and C^β pair, which is covalently bonded, and another C^α or C^β from the nearest residue. The superimpositions of the bonded pair are not dependent on SO_d as the bond length is invariant. In addition, the position of the nearest residue is also restricted by the Ramachandran Map (Ramachandran, Ramakrishnan, and Sasisekharan 1963; Ramakrishnan and Ramachandran 1965). This data was then evaluated to find the maximum value for SO_d that satisfies the condition that the 99th and 99.99th percentiles (different levels of permissiveness) of the RMSF distribution are below 0.5Å and 0.7Å. This value was computed at each level, providing efficient values for SO_d at different levels (Figure 2.8 bottom right plot).

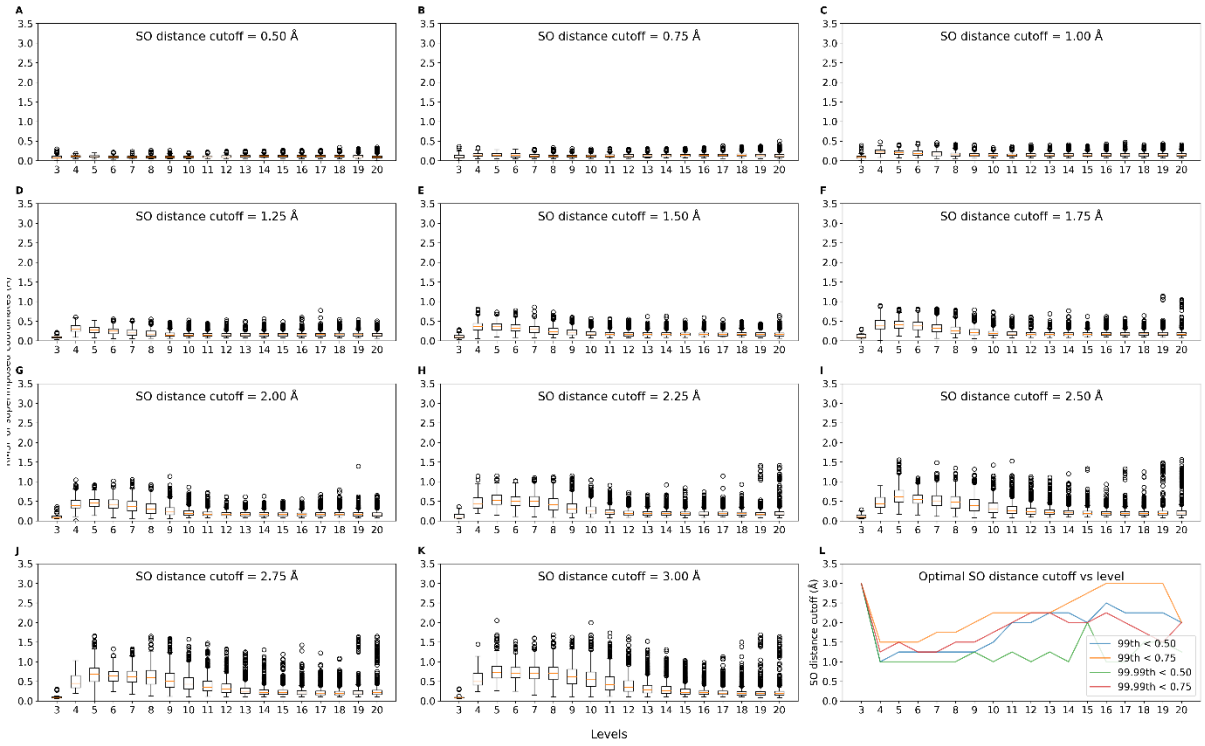


Figure 2.8: RMSF of coordinates of nodes at different levels and different SO_d. For panels A-K, the x-axis shows the level and the y-axis shows RMSF after superimpositions (except the bottom right where is SO_dshown). RMSF values are plotted as box plots with outliers shown in circles. The means of the distributions are shown with orange lines. Line traces in panel L show the optimal SO_d at different levels of permissiveness.

To optimise N, we evaluated the number of nodes (including temporary hanging nodes) at each level at different SO_d (Figure 2.8). An increase in the number of nodes implies the identification of new structural motifs by the tree. A decrease in the number of nodes means that the majority of the child nodes are hanging nodes, demarcating the end of structural similarity among the motifs. The number of nodes peaks between 9-13, a narrow range considering the change in SO_d (Figure 2.9 A-K). After level 12, we see a decline in the number of nodes in all SO_d. This implies that the structural information in C^α and C^βcliques saturate at level 12 or lower. Thus, we have set the cutoff of the number of levels in a clique to be 12. Note that even though the maximum level of a tree is set at 12, the cliques used to make the tree consist of 20 atoms. Thus, the tree selects the best-fitting 12 atoms from a superset of 20 for each clique.

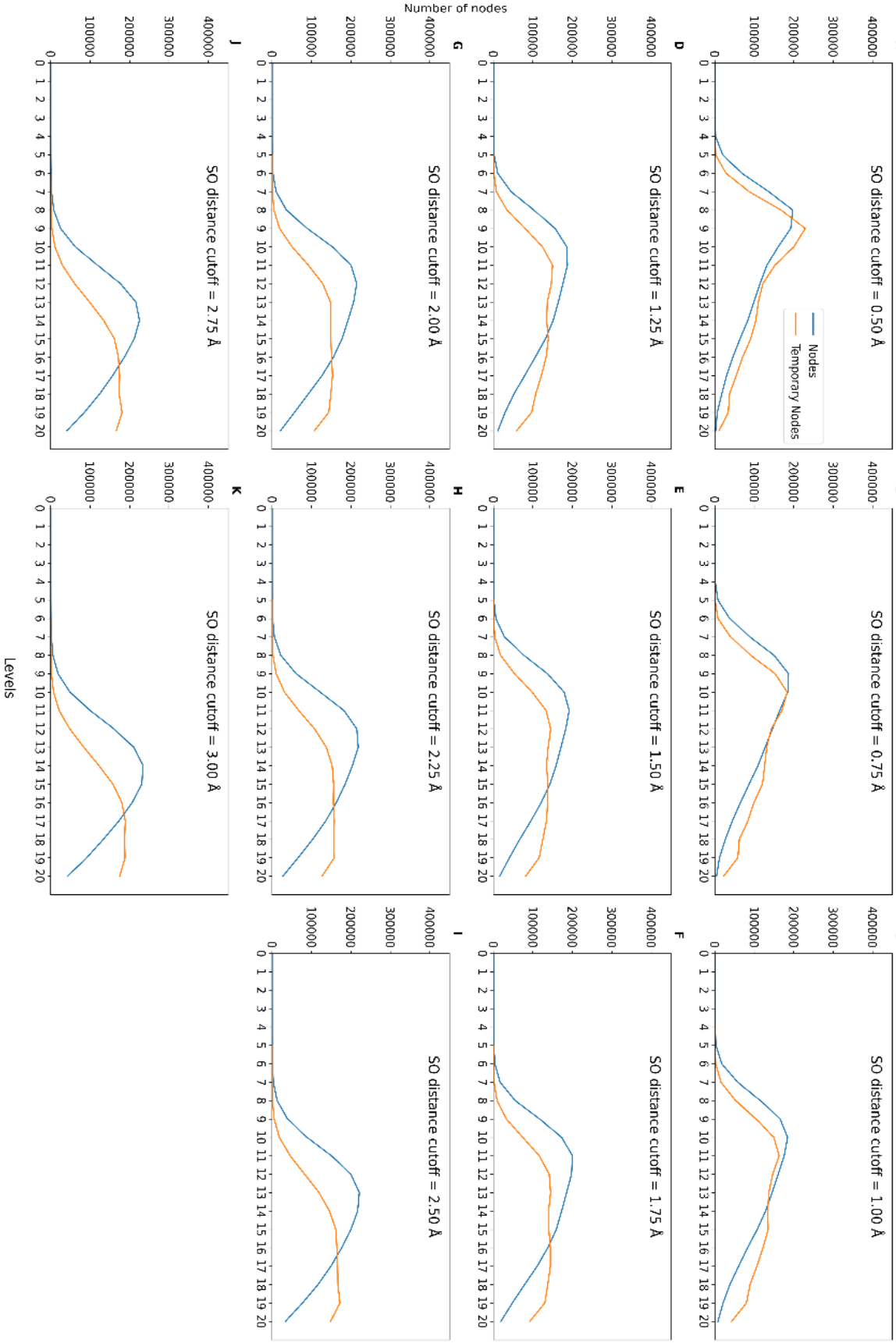


Figure 2.7: Panels show the number of nodes at different SO_dcutoff. The number of nodes are shown on the y-axis, and corresponding levels are shown in the x-axis. Blue traces show the number of regular nodes, and the red ones show the number of hanging nodes.

The predicted SO_d for various percentiles of RMSF along with cutoff obtained from the CLICK algorithm (M. N. Nguyen, Tan, and Madhusudhan 2011) and the number of nodes at different SO_d were used to generate a set of tight cutoffs (Table 2.2) and a set of weak cutoffs (Table 2.3).

Table 2.2: Stringent SO_d cutoffs determined for each level

Tight cutoffs	
level	SO_d (Å)
2	0.25
3-5	0.5
5-8	1
9-13	1.25
>13	1.5

Table 2.3: Permissive SO_d cutoffs determined for each level

Weak cutoffs	
level	SO_d (Å)
2	0.75
3-8	1.0
9-13	1.25
>13	1.5

Now that we have discussed how the cutoffs were optimised, we now explore how trees were created from various subsets of the PDB.

Trees made from non-redundant subsets of PDB

Three non-redundant subsets of PDB chains were obtained from PISCES the webserver (Wang and Dunbrack 2003). Each of these structures was deconvoluted into cliques consisting of the nearest 19 C^α and C^β atoms for each atom in the structure, forming 20 membered cliques. Two trees were created for each of these subsets corresponding to the tight and permissive cutoffs, forming a total of six trees. These trees were limited to 12 levels (the largest indexed motifs are of size 12),

implying that the best-fitting 12 atoms from the 20 in a clique were selected to form the motifs.

To evaluate the level of structural homogeneity within nodes of the tree, we selected the top hundred nodes (based on population) from levels 3-12 and superimposed 1000 members based on the equivalences listed in the tree. The RMSF of each atomic position was evaluated by considering member cliques as alternate positions (Similar to the previous section). We observe that the RMSF values consistently lie in the range 0-0.3 Å with very small error bars (Figure 2.10 A-F). Even outliers are within 1 Å (the size of a hydrogen atom). These observations combined imply very high structural homogeneity within the structural motifs encoded by the nodes in the trees. Interestingly, noticeable differences can be observed between the tight and permissive (weak) cutoffs (Figure 2.10 A, C and E vs B, D and F).

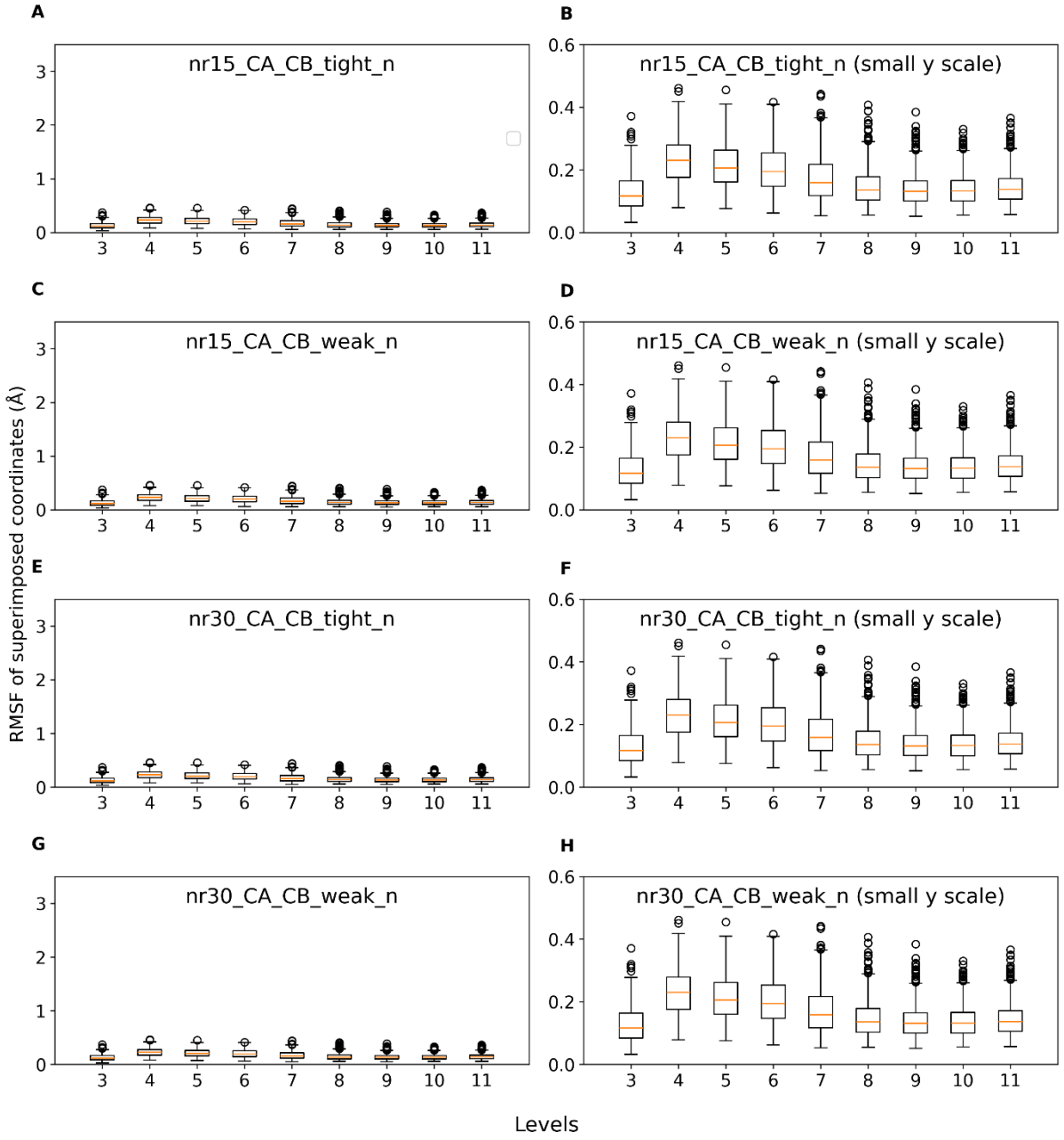


Figure 2.8: RMSF of superimposed coordinates vs Level. Panels show box plots of RMSF in the coordinates of the top 100 nodes of trees constructed from various databases (redundancy levels of databases indicated in panel titles). The orange line shows the mean, the edges of the boxes show the interquartile ranges, and the whiskers show 1.5 times the interquartile range. Axis of the graphs in panes A, C, E, and G are the same scale as that of Figure 2.8 to compare the difference in RMSF before and after optimisation of the parameters. Y axes of panes B, D, F, and H are adjusted to show the difference between the box plots.

In addition, we have also evaluated the number of nodes at different levels to check if the tree has enough levels to capture the structural information in the database efficiently. We see that the number of nodes plateaus at levels 10-11 for all the databases (Figure 2.11 A-F). The total number of nodes increases with database size, suggesting that all motifs are not captured within NR15 and NR30 subsets and that a

more extensive database encounters more unique motifs (Figure 2.11 each row represents a database).

There is a noticeable decrease in the number of hanging (temporary) nodes between levels 8 and 9 (Figure 2.11 A-F). This can be explained by the fact that SO_d cutoff increases (becomes more permissive) between levels 8 and 9 (Tables 2.2 and 2.3). Surprisingly, the dip in the number of hanging nodes is not reflected in the number of true nodes. This implies that there is scope for more permissive cutoffs without affecting the clustering efficiently. Thus, a finer optimisation of SO_d would result in more efficient trees.

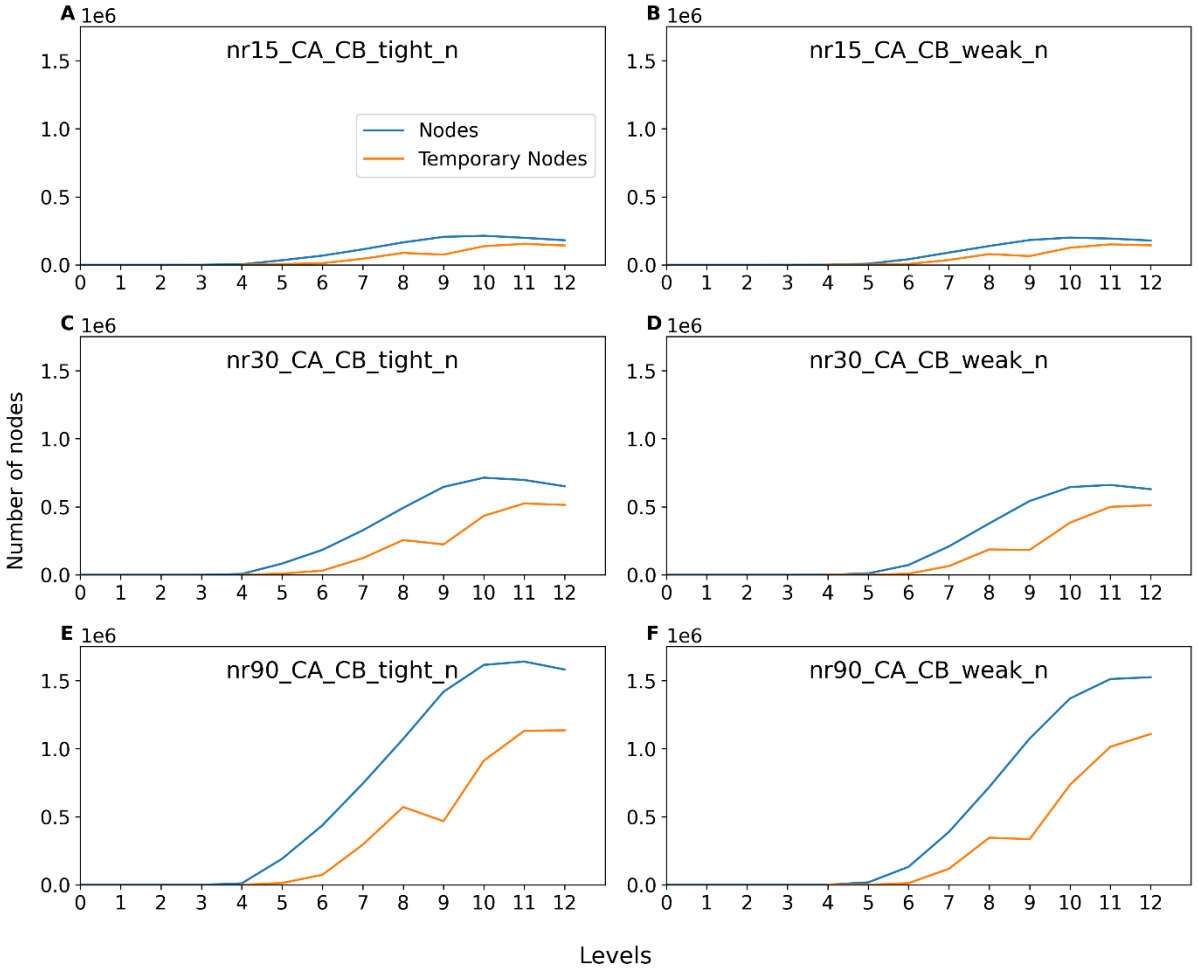


Figure 2.9: Number of nodes vs levels. Panels show the number of nodes on the y-axis and the number of nodes on the x-axis for trees constructed from various databases (indicated in the title). The blue trace shows the number of nodes, whereas the orange trace is the number of hanging nodes. The y-axis values are multiples of 1 million.

Visual representation of structural motifs encoded by nodes

In this section, we aim at visualising the nodes and their relation in the tree. We do this by outputting each member of a node in a clique as a PDB format file (Berman et al. 2000). The file contains each clique as a separate model, with the first model being the node representative. These PDB files can then be visualised in UCSF ChimeraX (Pettersen et al. 2021).

We have selected node 2 and its top 2 children and grandchildren from the tree constructed using the weak cutoffs from NR30 for representative purposes. One hundred cliques from each node were visualised in ChimeraX, and the representations are arranged in their respective hierarchical order in Figure 2.12. The colours of the atoms denote equivalent locations, and the atom type is denoted as text. The atoms have very close grouping in space to such an extent that nodes 2, 4 and 5 appear as a single point. The three-dimensional distributions formed by the members of a node can be observed in nodes 22 and 16375 (Figure 2.12). Progressing from a lower to a higher level entails the addition of extra atoms. This process is represented with atom colours in Figure 2.12. Notice that the position of this additional atom between two children of a node is distinct and represents different structural motifs.

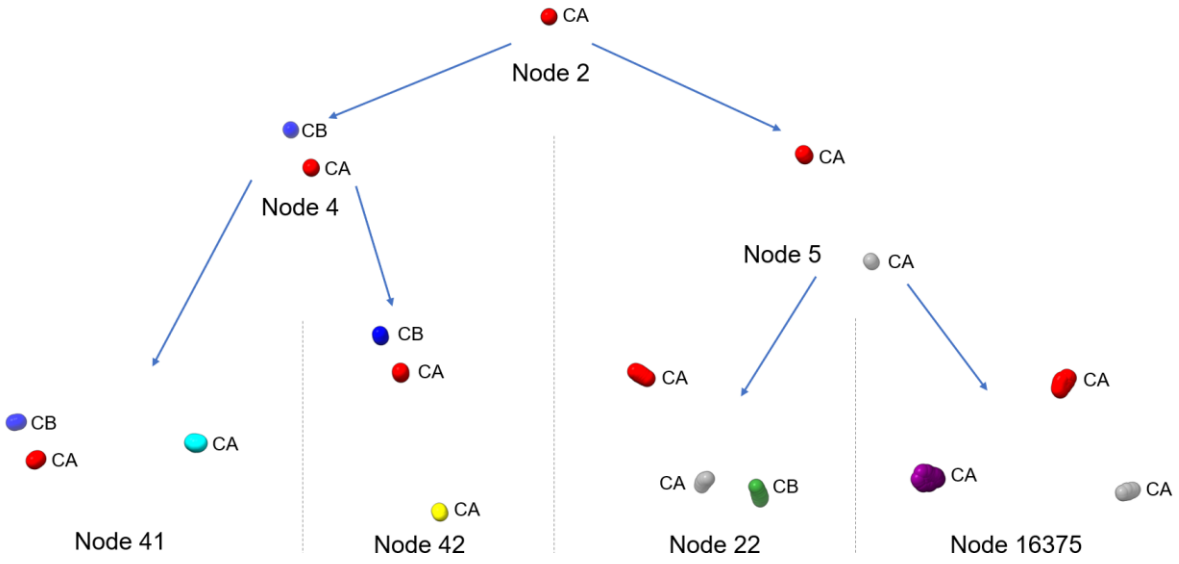


Figure 2.10: Visual representation of nodes and their hierarchy in the tree. Nodes from a tree, made from NR30 subset of PDB using permissive cut-offs Coloured spheres represent atoms with colours representing equivalent positions. There are 100 atoms plotted in each position. Grey dotted lines demarcated different nodes. Blue arrows show the relation between a parent and child node.

Now that we have established how trees are constructed let us look at some of the tree's applications.

Proteins Coordinate Basic Local Alignment Search Tool (ProCoBLAST)

Any comprehensive comparison of two proteins requires the comparison of their 3D structures. Since protein structures are experimentally solved with differing reference frames, such comparisons require superimposition of the protein structures. This process involves finding a rotation and translation that transforms the coordinates of one protein such that a subset (or all) of the atoms overlaps. Such rotation and translation can be quickly deduced analytically if the pairs of overlapping atoms (equivalences) are already known (Kabsch 1976; Kearsley 1989; Theobald 2005).

Structural comparison of sequentially related proteins is easy as the equivalences can be obtained using a sequence alignment. The latest method that uses such an approach is FoldSeek (van Kempen et al. 2024). This approach converts the sequence of amino acids into a sequence of 'structural alphabets' encoded by a neural network. Such a sequence database is then indexed, enabling fast searches for structurally and sequentially similar proteins. However, two structures having similar three-dimensional shapes do not necessarily have a similar sequence or even arrangement (or order) of secondary structures. Take the example of structures with PDB IDs 1tig and 1xxa (parts of which are shown in Figure 2.13) that are similar in three dimensions. However, notice that the connections between the secondary structure and the orientation of the middle β strand are different. Nevertheless, many of their C^{α} atoms superimpose due to their structural similarity. Sequence-dependent approaches cannot capture such similarities.

Sequence-independent (topology-independent) structure alignments require first finding the equivalences between two structures. There are many approaches that tackle this problem. A popular approach is TM-align, which uses the similarity of secondary structures or threading to align two structures. However, this approach implies that only the main chain of the structures can be superimposed. In many situations, other atoms, such as hydrogen bond donors and acceptors, must also be superimposed to study DNA-protein interactions (Nair and Madhusudhan 2022). In such cases, an algorithm called CLICK can select the atoms used in superimposition (Nguyen et al. 2017; M. N. Nguyen et al. 2011). CLICK first finds all pairs of three-membered 'cliques' (see Chapter 1 methods section) of atoms that superimpose within a Root Mean Square Deviation (RMSD) cutoff. Then, it iteratively adds atoms that are

in the vicinity of the three-membered clique to generate larger superimpositions. This is done till seven members are identified. This seven-membered superimposition is used to superimpose the two structures using which other overlapping atoms based on overlap are identified, generating the set of equivalent atoms. This approach has had various applications, including drug discovery, analysis of DNA-protein interaction and protein-protein interactions (Nair and Madhusudhan 2022; Nguyen et al. 2019; Sen et al. 2019; Sen and Madhusudhan 2022). However, searching a database with this approach is inefficient as it requires individual pairwise comparisons.

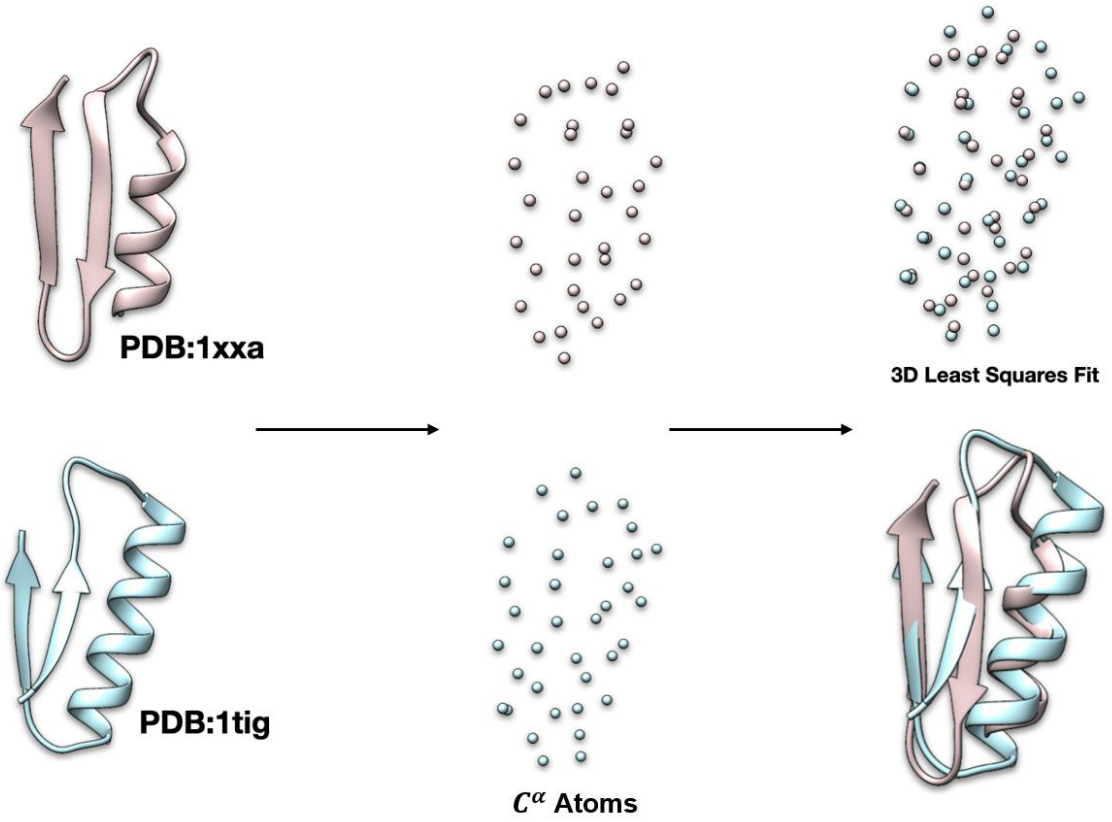


Figure 2.11: Topology independent structure alignment. Partial structures of 3-isolorpymlalate dehydrogenase and translation initiation factor 3 are shown in red and blue, respectively. The C^{α} atoms of each structure are shown in ball representation. The superimposition with the least square fit (minimum RMSD) is shown in the top right corner for C^{α} atoms in ball representation and in ribbon in the bottom right corner

In this section, we describe a method, called ProCoBLAST, that provides the functionality of CLICK in the context of an efficient database search. CLICK uses pairwise similarity between cliques of atoms between two structures for their superimposition (M. N. Nguyen et al. 2011). We used ProCoBLAST to search NR30 databases using trees generated using tight cutoffs (Chapter 1). Example searches based on C^{α} and C^{β} atoms were performed for structures of green fluorescent protein

GFP (PDB ID: 1ema) (Shimomura, Johnson, and Saiga 1962) and topoisomerase inhibitor CcdB (PDB ID: 3vub) (Miki, Yoshioka, and Horiuchi 1984), and the results were sorted based on query coverage and RMSD.

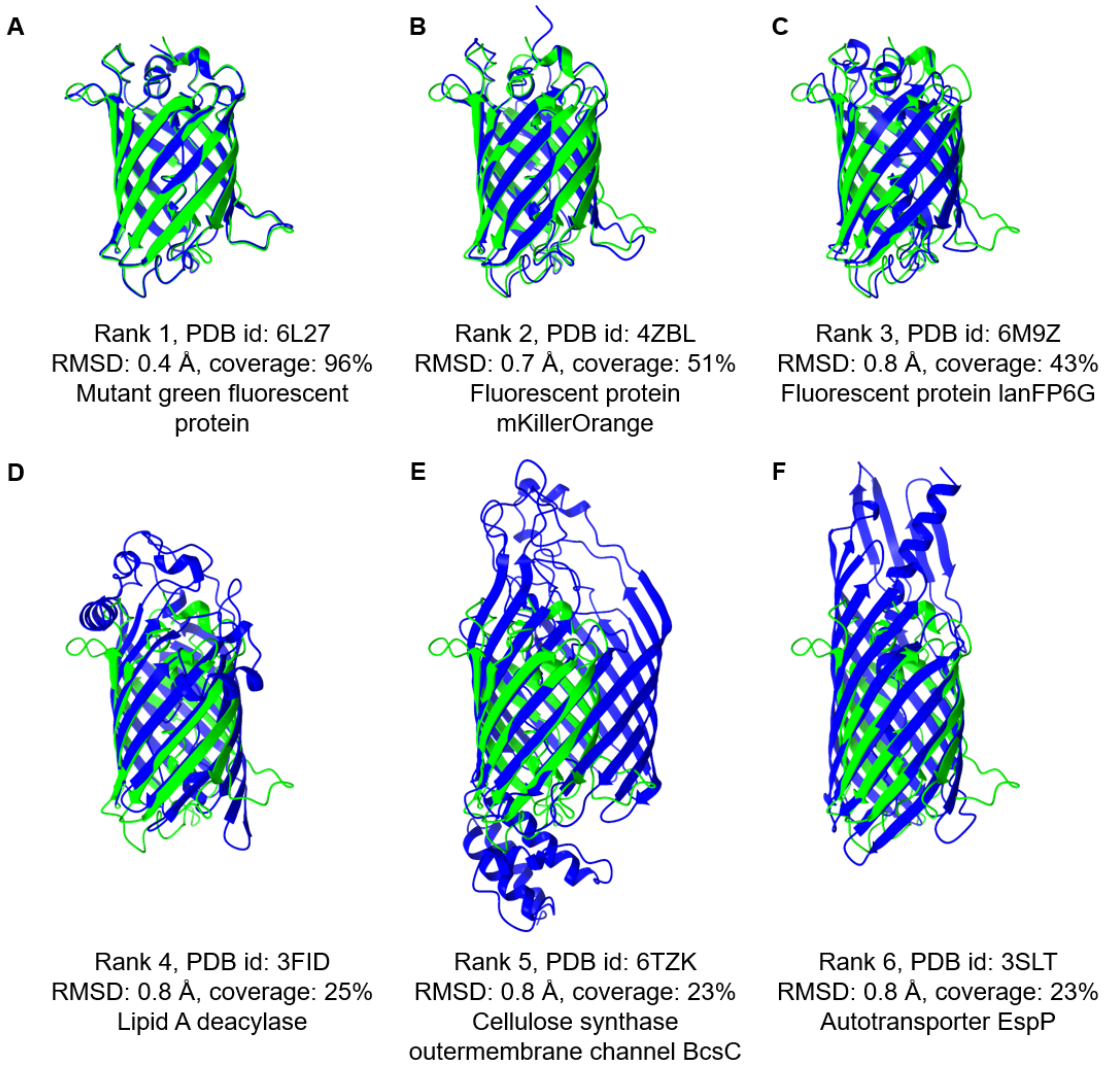


Figure 2.12: Top 6 hits of ProCoBLAST using Green Fluorescent Protein (PDB ID 1ema) as the query. Panels show the superimposition of GFP (shown in green) and matches from the NR30 database (shown in blue). Rank of the superimposition, PDB IDs, RMSD, query coverage and the name of the proteins are shown as captions.

The top six matches on searching the NR30 database with GFP are shown in Figure 2.14. The top three matches are all fluorescent proteins (Figure 2.14 A-C). The rest of the three proteins are all membrane-embedded β-barrels with various functions. These proteins are not evolutionarily related. These structures are superimposed in a topology-independent manner by ProCoBLAST with the partial matches of the β-strands. (Figure 2.14 D-F).

The superimpositions of CcdB show a more diverse set of matches. Again, the match with rank 1 is structurally very similar to Ccdb (Figure 2.15 A) (MazF mRNA interfering proteins). Ranks 2, 4, 5 and 6 superimposed to the SH3-like domain of CcdB with varying levels of coverage (Figure 2.15 B, D, E and F). Rank 3, however, is superimposed by the alpha helix and a significant portion of the nearby β-strands.

The searches used a conservative distance cutoff of 1.5 Å to compute overlaps. This means that the superimpositions shown in Figure 2.14 and Figure 2.15 are performed such that no two matching atoms (equivalences) are farther than 1.5 Å. This also gives us RMSDs (Figure 2.14 and Figure 2.15) that are below 1 Å. A more permissive cutoff will increase the query coverage while sacrificing the fit quality. These searches were conducted over the entire database of 12079 structures in approximately 10 minutes. A search of the NR90 database (taking a similar amount of time) was also conducted, and most of the top-ranking matches were structures of close relatives of the query.

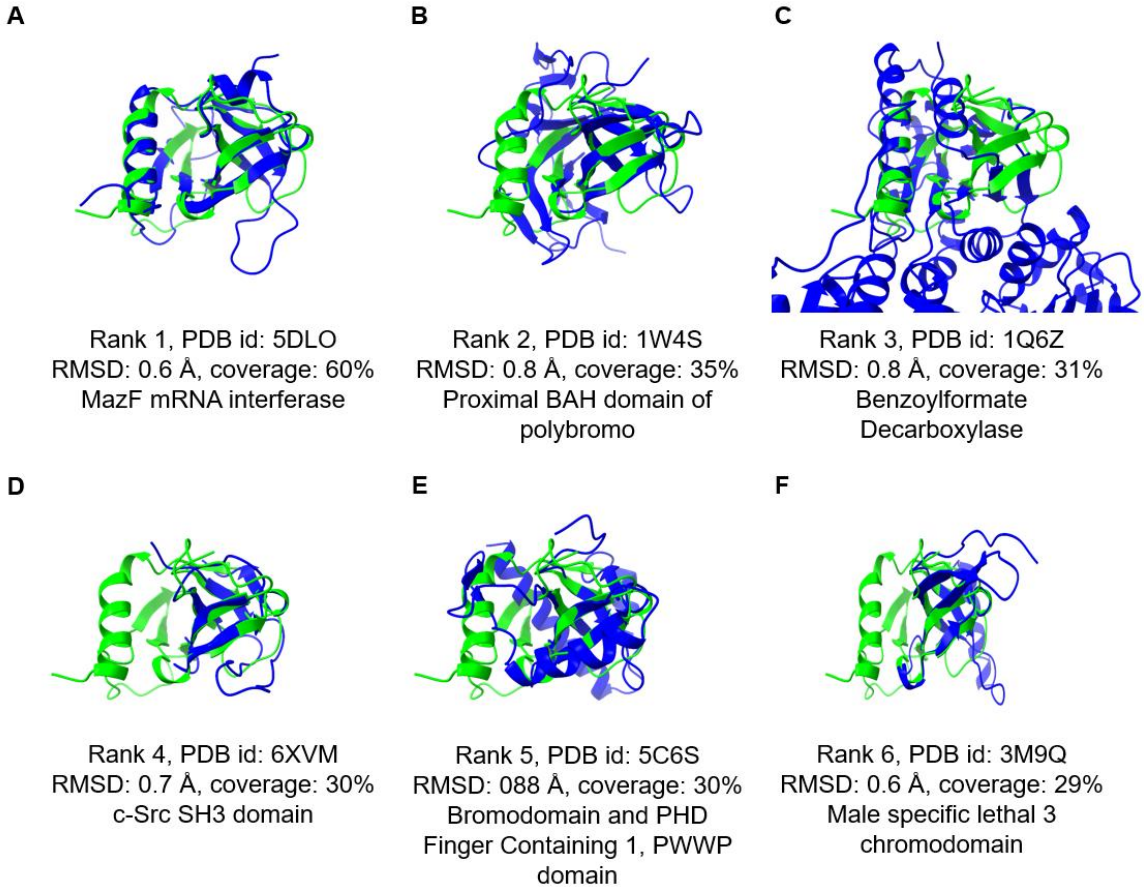


Figure 2.13: Top 6 hits of ProCoBLAST using Topoisomerase inhibitor CcdB (PDB ID 3vub) as the query. Panels show the superimpositions of CcdB (shown in green) and the matches from the NR30 database (shown in blue). Rank of the superimposition, PDB IDs, RMSD, query coverage and the name of the proteins are shown as captions.

These examples demonstrate the topology-independent superimposition capabilities of our approach. However, these results can also be obtained via TM-align, and a subset of the hits can also be obtained from FoldSeek. CLICK, however, identifies a similar but not identical set of matches (Table 2.4). This is likely due to differences in superimposition parameters. All top six matches for 1EMA (Figure 2.14) are identified by CLICK. In addition, CLICK also identified matches that are not in the top list in ProCoBLAST. This is due to the tighter cutoffs of ProCoBLAST, which is optimised to sacrifice global similarity for better RMSD for local fit. In the case of 3VUB (Figure 2.15), only 5DLO, 6XVM and 3M9Q were superimposed by CLICK. This can be explained by the smaller size of 3VUB, which benefits from the tighter cutoffs. T

The superimposition of NR30 with click took 4 hrs compared to 20 minutes using identical hardware. CLICK in general identified more matches at poorer RMSD values. ProCoBLAST on the other hand was set to output top 100 matches, most of which falls in the range of 0.5-1 Å.

Table 2.4: Statistics of CLICK vs ProCoBLAST

RMSD (Å)	1EMA				3VUB			
	CLICK		ProCoBLAST		CLICK		ProCoBLAST	
	Number of hits	Average number of superimposed atoms	Number of hits	Average number of superimposed atoms	Number of hits	Average number of superimposed atoms	Number of hits	Average number of superimposed atoms
0-0.5	0	-	1	414	0	-	0	-
0.5-1	5	92.2	98	80.2	4	10.8	100	48.78
1-1.5	8	15.25	1	73	5	14.5	0	-
1.5-2	13	107.07	0	-	16	40.5	0	-

Superimposition of small fragments of proteins

Three-dimensional superimposition of a binding pocket to an uncharacterised protein would help annotate functions. For the function of small molecule binding, this process would require a program that superimposes a small pocket made up of discontiguous fragments. We have created a database of such pockets involved in phosphoinositide binding identified in the PDB database (List of PDBs used are shown in Appendix Section 6). We then constructed trees from this set of pockets to create a database of phosphoinositide binding pockets. As a test case, this database was with the

phosphoinositide binding proteins with PDB ID 1h6h. The superimpositions of the output of such queries show substantial overlap between the cyclohexane ring of the phosphoinositide molecules from different structures (Figure 2.16). The protein binding pockets show various conformations but nevertheless superimpose at conserved residue positions, which are presumably important for phosphoinositide binding (Figure 2.16).

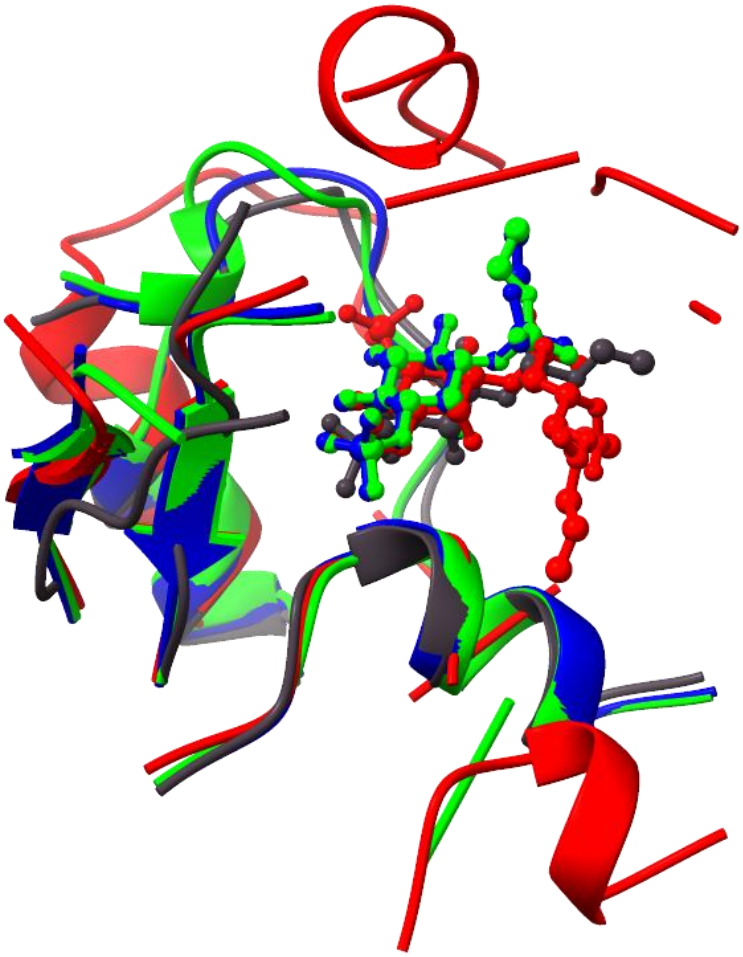


Figure 2.14: Superimposition of phosphoinositide binding pockets. The ribbon representation shows the proteins part. Ball and stick representation shows the phosphoinositide. Green shows the pocket binding pocket in PDB ID 1ocu. Blue shows the pocket binding pocket in PDB ID 7blq. Red shows the pocket binding pocket in PDB ID 6koj. Grey shows the pocket binding pocket in PDB ID 2rak.

Note that searching any of the fragments shown in in Figure 2.16Figure 2.14 using both TM-align and FoldSeek outputs no results. This is because both TM-align and FoldSeek deduce features (secondary structures and alphabets) based on the input. These features are then used to search the database and they fail when these features cannot be deduced, as in our example of very short discontinuous fragments that form

the pockets. Our method, however, does not have this limitation, as all searches are performed directly using only atomic coordinates.

Discussion

In this chapter, we have described a tree data structure that can be used to index and cluster contiguous and noncontiguous structural motifs purely based on three-dimensional coordinates. We have described how to create structural motifs called cliques. We then explained in depth how to index and search the tree. We also described the process of optimising various cutoffs for the tree. We have created trees from various subsets of PDB at two different permissiveness levels of the cutoffs. We have also seen example representations of nodes of these trees showing the structural motifs that cluster in 3 dimensional space. Searching a tree with a query clique maps a query clique to a lineage of nodes. Since each node in the tree represents a group of structurally similar cliques from the database, this search inherently maps the query clique to all similar cliques from the database. Such mapping can have various applications.

We explored one such application to make a database search variant of CLICK where the tree can accelerate the pairwise mapping of the clique between a query structure and all the structures in a database. We have seen some examples of this application, including a case where our method identifies structural matches even when both TM-align and FoldSeek fail.

Another application of the tree is the identification of structural templates/scaffolds for protein engineering; we describe such an approach in detail in chapter 3.

Our method is unique in its ability to identify repeating non-contiguous three-dimensional motifs in large datasets. In addition, these motifs are conveniently arranged in a hierarchical manner to enable efficient searching. The location of such motifs across the dataset is also recorded in the process, forming an index of spatial local neighborhoods. We are actively developing and studying this method and its various applications.

Reference

Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. 1990. “Basic Local Alignment Search Tool.” *Journal of Molecular Biology*

215(3):403–10. doi: 10.1016/S0022-2836(05)80360-2.

Anfinsen, Christian B. 1973. “Principles That Govern the Folding of Protein Chains.” *Science* 181(4096).

Berman, H. M., John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. “The Protein Data Bank.” *Nucleic Acids Research* 28(1):235–42. doi: 10.1093/nar/28.1.235.

Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. “Array Programming with NumPy.” *Nature* 585(7825):357–62. doi: 10.1038/s41586-020-2649-2.

Kabsch, W. 1976. “A Solution for the Best Rotation to Relate Two Sets of Vectors.” *Acta Crystallographica Section A* 32(5):922–23. doi: 10.1107/S0567739476001873.

Kearsley, S. K. 1989. “On the Orthogonal Transformation Used for Structural Comparisons.” *Acta Crystallographica Section A* 45(2):208–10. doi: 10.1107/S0108767388010128.

van Kempen, Michel, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. 2024. “Fast and Accurate Protein Structure Search with Foldseek.” *Nature Biotechnology* 42(2):243–46. doi: 10.1038/s41587-023-01773-0.

De La Briandais, Rene. 1959. “File Searching Using Variable Length Keys.” Pp. 295–298 in *Papers Presented at the the March 3-5, 1959, Western Joint Computer Conference*, *IRE-AIEE-ACM ’59 (Western)*. New York, NY, USA: Association for Computing Machinery.

Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert. 2015. “Numba: A LLVM-Based Python JIT Compiler.” in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, *LLVM ’15*. New York, NY, USA: Association for Computing Machinery.

Miki, T., K. Yoshioka, and T. Horiuchi. 1984. “Control of Cell Division by Sex Factor F in Escherichia Coli. I. The 42.84-43.6 F Segment Couples Cell Division of the Host Bacteria with Replication of Plasmid DNA.” *Journal of Molecular Biology* 174(4):605–25. doi: 10.1016/0022-2836(84)90086-x.

Nair, Sanjana, and M. S. Madhusudhan. 2022. “JEDII: Juxtaposition Enabled DNA-Binding Interface Identifier.” *BioRxiv* 2022.05.19.492702. doi: 10.1101/2022.05.19.492702.

Needleman, Saul B., and Christian D. Wunsch. 1970. “A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins.” *Journal of Molecular Biology* 48(3):443–53. doi: https://doi.org/10.1016/0022-2836(70)90057-4.

Nguyen, M N, K. P. Tan, and M. S. Madhusudhan. 2011. “CLICK - Topology-Independent Comparison of Biomolecular 3D Structures.” *Nucleic Acids Research* 39(SUPPL. 2). doi: 10.1093/nar/gkr393.

Nguyen, M. N., K. P. Tan, and M. S. Madhusudhan. 2011. “CLICK - Topology-Independent Comparison of Biomolecular 3D Structures.” *Nucleic Acids Research* 39(SUPPL. 2):W24–28. doi: 10.1093/nar/gkr393.

Nguyen, Minh N., Neeladri Sen, Meiyin Lin, Thomas Leonard Joseph, Candida Vaz, Vivek Tanavde, Luke Way, Ted Hupp, Chandra S. Verma, and M. S. Madhusudhan. 2019. “Discovering Putative Protein Targets of Small Molecules: A Study of the P53 Activator Nutlin.” *Journal of Chemical Information and Modeling* 59(4):1529–46. doi: 10.1021/acs.jcim.8b00762.

Nguyen, Minh N., Adelene Y. L. Sim, Yue Wan, M. S. Madhusudhan, and Chandra Verma. 2017. “Topology Independent Comparison of RNA 3D Structures Using the CLICK Algorithm.” *Nucleic Acids Research* 45(1). doi: 10.1093/nar/gkw819.

Pettersen, Eric F., Thomas D. Goddard, Conrad C. Huang, Elaine C. Meng, Gregory S. Couch, Tristan I. Croll, John H. Morris, and Thomas E. Ferrin. 2021. “UCSF ChimeraX: Structure Visualization for Researchers, Educators, and Developers.” *Protein Science : A Publication of the Protein Society* 30(1):70–82. doi: 10.1002/pro.3943.

Ramachandran, G. N., C. Ramakrishnan, and V. Sasisekharan. 1963. “Stereochemistry of Polypeptide Chain Configurations.” *Journal of Molecular Biology* 7(1).

Ramakrishnan, C., and G. N. Ramachandran. 1965. “Stereochemical Criteria for Polypeptide and Protein Chain Conformations. II. Allowed Conformations for a Pair of Peptide Units.” *Biophysical Journal* 5(6):909–33. doi: 10.1016/S0006-3495(65)86759-5.

Sen, Neeladri, Tejashree Rajaram Kanitkar, Ankit Animesh Roy, Neelesh Soni, Kaustubh Amritkar, Shreyas Supekar, Sanjana Nair, Gulzar Singh, and M. S. Madhusudhan. 2019. “Predicting and Designing Therapeutics against the Nipah Virus.” *BioRxiv* 623603. doi: 10.1101/623603.

Sen, Neeladri, and Mallur S. Madhusudhan. 2022. “A Structural Database of Chain–Chain and Domain–Domain Interfaces of Proteins.” *Protein Science* 31(9):e4406. doi: https://doi.org/10.1002/pro.4406.

Shimomura, Osamu, Frank H. Johnson, and Yo Saiga. 1962. “Extraction, Purification and Properties of Aequorin, a Bioluminescent Protein from the Luminous Hydromedusan, Aequorea.” *Journal of Cellular and Comparative Physiology* 59(3):223–39. doi: https://doi.org/10.1002/jcp.1030590302.

Theobald, Douglas L. 2005. “Rapid Calculation of RMSDs Using a Quaternion-Based Characteristic Polynomial.” *Acta Crystallographica Section A* 61(4):478–80. doi: 10.1107/S0108767305015266.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric

Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T. Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, Yoshiki Vázquez-Baeza, and SciPy 1. .. Contributors. 2020. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” *Nature Methods* 17(3):261–72. doi: 10.1038/s41592-019-0686-2.

Wang, Guoli, and Roland L. Dunbrack. 2003. “PISCES: A Protein Sequence Culling Server.” *Bioinformatics* 19(12):1589–91. doi: 10.1093/bioinformatics/btg224.

Zemla, Adam. 2003. “LGA: A Method for Finding 3D Similarities in Protein Structures.” *Nucleic Acids Research* 31(13):3370–74. doi: 10.1093/nar/gkg571.

Zhang, Yang, and Jeffrey Skolnick. 2004. “Scoring Function for Automated Assessment of Protein Structure Template Quality.” *Proteins* 57(4):702–10. doi: 10.1002/prot.20264.

Zhang, Yang, and Jeffrey Skolnick. 2005. “TM-Align: A Protein Structure Alignment Algorithm Based on the TM-Score.” *Nucleic Acids Research* 33(7):2302–9. doi: 10.1093/nar/gki524.

Chapter 3: Engineering functional chemical neighborhoods into existing protein structures

Introduction

Proteins are polymers of twenty different amino acids that fold upon themselves to form a specific 3-dimensional structure (Anfinsen 1973). The structure of a protein determines its function (Mannige 2014). This is because the folding process brings various types of amino acids in the polypeptide chain in proximity to form active sites. Designing new proteins is essential to creating synthetic biology tools for multiple applications. This process of protein design aims to find the best sequence for a given structure, inverse-folding.

Protein design

Modern computational design methods generally follow two steps. The first step involves obtaining a peptide backbone for given inputs, and the second step involves finding the best set of side chains to fit the backbone. The process of obtaining a peptide backbone follows two predominant approaches. The first method is assembling fragments of the backbone found in the PDB structure. This process is based on the fact that nature uses different combinations of the same structural motifs to create various structures. These fragments are identified from RCSB PDB and stored in a database. For example, Baker and coworkers, in their design software Rosetta, uses fragment from this database to construct protein backbones that fit the input topology (Bowers, Strauss, and Baker 2000; Kuhlman et al. 2003). In SEWING, Kuhlman and coworkers used continuous and discontinuous substructures assembled based on N and C terminal similarity (Jacobs et al. 2016). Fleishman and co-workers used fragments from aligned structures of antibodies to design backbone templates for new antibodies (Lapidoth et al. 2015).

The second step of finding the best sequence for the backbone template usually involves searching a rotamer space using an appropriate search algorithm and an energy function. Rotamers are low energy conformations of amino acid sidechains observed in the PDB (Dunbrack 2002; Ponder and Richards 1987). The lowest energy combination of such rotamers can be found using stochastic algorithms like simulated annealing using Monte Carlo sampling, e.g. Rosetta by Baker and coworkers (Bowers et al. 2000; Kuhlman et al. 2003; Kuhlman and Baker 2000; Kuhlman and Bradley 2019). Or deterministic algorithms like dead-end elimination, e.g. DEEPer by Donald and coworkers (Desmet et al. 1992; Hallen, Keedy, and Donald 2013). Stochastic

algorithms are much faster than deterministic algorithms, but their results may not give the global minimum. An energy function is used to evaluate the stability of each sequence at each step of the design process. They are usually a combination of knowledge-based and molecular mechanics energy terms. They tend to vary significantly from method to method and between designs (Boas and Harbury 2007). One method that eliminates the need for rotamers is dTERMen by Grigoryan and coworkers. They identified structural motifs called TERMS. They also calculated the amino acid propensity for each residue position for these motifs. Using these propensities and motifs, they redesigned the mCherry surface and found that the redesigned protein has native-like properties (Mackenzie, Zhou, and Grigoryan 2016; Zhou, Panaitiu, and Grigoryan 2020). Application of attention and diffusion neural networks for protein structures have resulted in programs like ProteinMPNN and RFdiffusion, which have enhanced the processes of protein design significantly by improving both accuracy and accessibility (Dauparas et al. 2022; Watson et al. 2023).

Conventional protein design methods focus on generating a sequence for a given structure. Even though there has been considerable progress in the stability and complexity of designed proteins, de-novo design of proteins with a function reliably is still difficult. This is because proteins undergo conformational changes to perform their functions. But, designed proteins are usually hyper-stable (Dou et al. 2018; Kuhlman et al. 2003; Polizzi and DeGrado 2020; Xu et al. 2020) and cannot do this. The design of functional proteins usually involves engineering an existing de-novo-designed scaffold to introduce functional sites (Yeh et al. 2023). In this chapter, we aim to create such a method that transfers the functional site of one protein into another scaffold. We use this process to optimise a sequence after introducing a GFP chromophore forming neighbourhood into a non-fluorescent structure. Although this is not the first attempt at designing a fluorescent protein (Dou et al., 2018), it is the first attempt where the chromophore self-cyclizes within the protein.

Green fluorescent protein

Shimomura and co-workers first isolated green fluorescent protein (GFP) and aequorin from Aequorea victoria in 1962. (Shimomura, Johnson, and Saiga 1962). In *Aequorea victoria*, GFP seems responsible for converting the blue chemiluminescence from the Ca^{2+} sensor aequorin into green emission. The wild-type GFP is a 238-residue-long

single chain that is stable and resistant to proteolysis. Its two absorption maxima are 395 nm and 475 nm, with absorption at 395nm, giving an emission peak of 508 nm with a quantum yield of 0.72-0.85 (Heim, Prasher, and Tsien 1994; Ormö et al. 1996). Today, GFP and its various mutants are commonly used as a biomarker to tag proteins *in vivo*.

The first structure of GFP was solved by Ormö and co-workers in 1996 (PDB ID:1EMA). It consists of an 11-stranded beta-sheet (each strand approximately 9 to 13 residues long) surrounding a central alpha-helix (Figure 3.1). The p-hydroxybenzylideneimidazolidinone chromophore is formed from the amino acids S/T65-Y66-G67 in the central alpha helix. It is present roughly at the centre of the protein and is completely protected from the bulk solvent (Ormö et al. 1996) (Figure 3.1). The basis of the formation of the chromophore and its fluorescence is described in the following section.

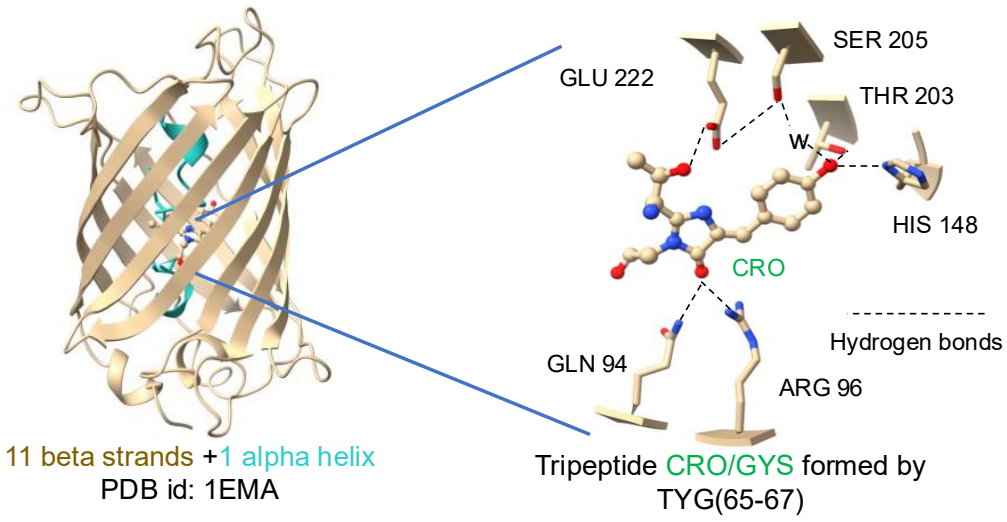


Figure 3.15: GFP and the neighbourhood of its chromophore. Panel on the left shows the structure of GFP (PDB ID 1EMA). The ball and stick structure behind the beta-sheet is the chromophore. Panel on the right shows a schematic showing the chromophore's first and second coordination spheres (Ormö et al.,1996).

The chromophore formation in GPF is an auto-catalytic main-chain cyclisation process followed by the oxidation of S/T65-Y66-G67 residues, which results in the formation of the chromophore p-hydroxybenzylideneimidazolidinone as part of the polypeptide chain. A key study that deduced the mechanism of GFP chromophore formation was published by Elizabeth Getzoff and co-workers in 2003 (Barondeau et al. 2003). Here,

the authors used structures of the pre-cyclised intermediate and the final chromophore to explain the mechanism of chromophore formation. They propose a conjugation trapping mechanism for the cyclisation and further oxidation of the chromophore facilitated by dehydration. They identified R96, E222 and T62 (Figure 3.1) as crucial residues for this reaction. They also suggest that the architectural distortion at the tripeptide contributes more to the formation as it is thought to avoid the energy cost of breaking the main-chain hydrogen bonds in the alpha helix, and it also brings the G67(N67) and S/T65(C65) together. Chemically synthesised chromophore in solution in non-fluorescent, as is denatured GFP (fluorescence recovers on refolding) (Niwa et al. 1996; Ward and Bokman 1982). This is because cis-trans isomerisation dissipates the excitation energy if the protein fold does not restrict the chromophore (Niwa et al. 1996; Romei and Boxer 2019). The residues at closer proximity to the chromophore are crucial in this and are R96, E222, T203, H148, S205 and Q94 (Romei and Boxer 2019) (Figure 3.1).

In this study, we aim to graft the function of GFP into another protein by replicating the local functional environment of GFP. We identify structural motifs that match the conformation of the chromophore forming residues in structures of GFP with pre-cyclised intermediates of the chromophore. In principle, we can use the tree described in Chapter 2 for a fast database search for this application. However, we did not use the tree here as it was conceptualised and developed before the work described in this chapter. Future attempts and improvements of the study in this chapter would include the use of the tree.

Methods

Protein function grafting by replication of local structure and environment

Our approach is broadly divided into three steps (Figure 3.2). The first step is searching the PDB database (Berman et al. 2000) for a structure motif. In the case of GFP, this consists of the finding motifs that are structurally similar to the three contiguous residues that cyclise to form the chromophore (Figure 3.2). The second step involves analysing the residues around the identified structural motifs to check if either those residues or their mutants can form the required chemical environments (Figure 3.2). For GFP, the chemical environment refers to the catalytic residues that help in cyclisation and the interactions that lock the chromophore in its isomeric state.

This process can be expected to generate many hits. The third step involved using various methods to filter and rank order these hits (Figure 3.2).

Figure 3.16: Function grafting workflow. The three distinct steps of POGGERS are shown under their corresponding titles. The structures shown in green are GFP (PDB ID 1ema), and blue is the tripeptide mutant (PDB ID 1qyo) in green ball and stick representation is the chromophore of GFP (labelled as CRO). Residues shown in pink show alternate conformations of the rotamers.

Search of structural motifs

The chromophore in GFP cannot be used as a reference for a structural motif search because of its cyclised nature. Thus, we have used a TYG to GGG mutant of GFP (PDB ID 1qyo) where the tripeptide forming the main chain is crystallised in the uncyclised state as a reference structure. Please note that the GGG tripeptide also forms a cyclic structure but has a longer cyclisation time (Barondeau et al. 2003). The superimposition of the tripeptide TYG (from PDB ID 1ema) and GGG (from PDB ID 1qyo) is shown in Figure 3.2 (left panel).

We use two approaches to search for the contiguous chromophore forming tripeptide. The first approach computes the phi and psi torsion angles of the tripeptide obtained from the GGG motif. We then compare these torsion angles to the torsion angles of all the query structures from the PDB database using a root mean square of the difference between the corresponding torsion angles. We quantified this RMSD of all tripeptide motifs in PDB compared to that of GGG. The distribution of RMSD shows how close the structural motifs in the PDB are to that of the GGG tripeptide. There are two distinct peaks in this distribution: alpha helices and beta sheets (Figure 3.3 A). The number of queries that satisfy a given cutoff is shown in Figure 3.3 B. This curve is essentially an integral of the distribution shown in Figure 3.3 A. The curve shows a sharp increase in between 0-50º. The trend of the curve between 0-20º shows a logarithmic rise in the number of hits with an increase in RMSD cutoff (Figure 3.3 C). A cutoff of 10º was selected based on these results.

Using a cutoff of 20º and 15º resulted in the selection of alpha helical motifs, which is energetically unfavourable, as the chromophore formation process would need to break hydrogen bonds. The cutoff of 10º was found to reliably avoid selection of alpha helical motifs during the search of structural motifs.

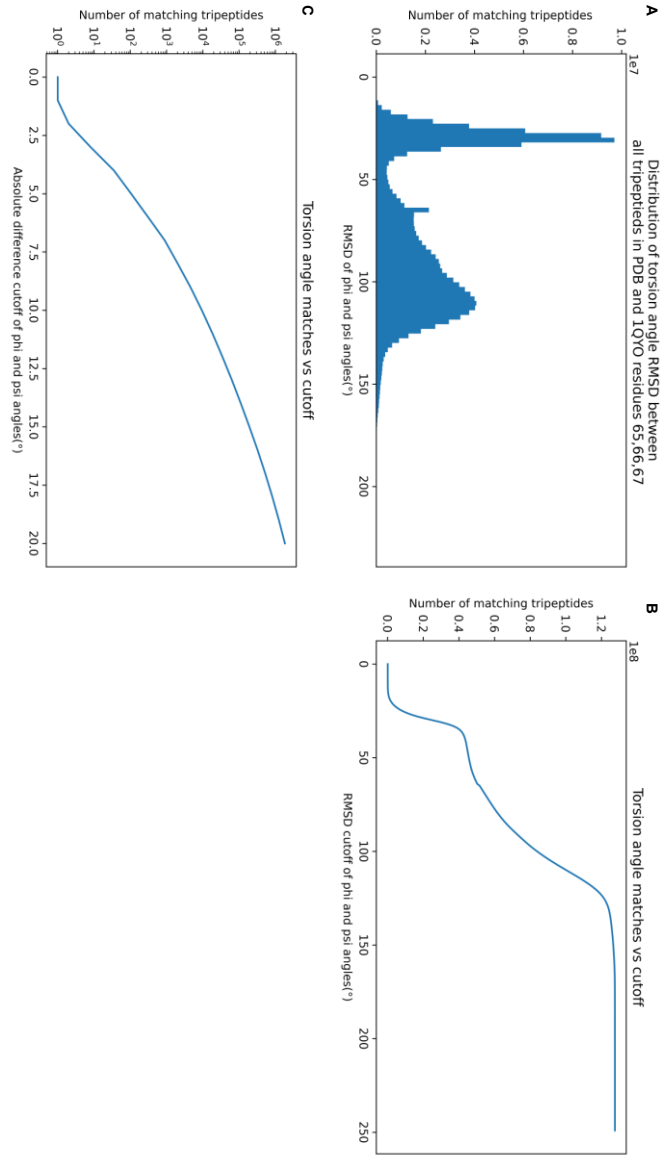


Figure 17.3: Optimisation of RMSD of phi and psi angles. Panel A shows the distribution of RMSD values for all PDB structures, number of matching tripeptides in y axis, RMSD (º) in x axis. Panel B shows the number of queries that satisfy different RMSD cutoff. Number of matching tripeptides in y axis and RMSD shown in the x-axis. Panel C is an enlarged part of panel B where the x-axis is 0-20º.

The second approach superimposed the main-chain atoms using Kearsley's algorithm (Kearsley 1989) to find the RMSD of the superimposition of the atoms. The structural motifs that superimpose with an RMSD lower than 0.2 Å were selected. Both processes provide a list of tripeptides that are structurally similar to the GGG tripeptide.

Identification of interacting residues

The query is first superimposed to the 1qyo based on the previously identified structural match. 1qyo, in turn, is already superimposed to 1ema based on sequence similarity. Thus, the chromophore from 1ema can now be superimposed on the query.

Residues within 6 Å Euclidean distance around the chromophore were selected as candidates for mutation. All rotamers (including mutants) with matching phi and psi angles and a probability greater than 0.001 were superimposed via the main-chain atoms for each of these residues. These rotamers were obtained from the Dunbrack rotamer library (Shapovalov and Dunbrack 2011). Hydrogen bonds, π-stacking, cation-π, and salt bridge interactions made by the chromophore with the rotamers and the original candidate residues were identified. Rotamers and template residues that satisfy the interactions and catalytic residues observed in GFP structures were identified. The rotamers that clash with the chromophore or any main chain atoms were removed from this list. This list was then ranked by scoring the entries according to the following preference

1) The original candidate's residue
    - Score = 2
2) A rotamer of the original residue’s type
    - Score = 1 + probability of occurrence of the rotamer from the rotamer table.
3) A mutant rotamer
    - Score = Probability of occurrence of the rotamer from the rotamer table.

A wild-type residue that can satisfy an interaction in its native conformation is most preferred, and it receives the maximum score of 2. This is because we do not have to make a mutation, and the residue does not have to move to make the required interaction. However, in most cases, the wildtype residue will have to change its side chain orientation to make our required interaction. This, however, carries the risk of disrupting existing interactions by the residue and is thus scored as 1 + the probability of occurrence of the rotamer according to the rotamer table (Shapovalov and Dunbrack 2011). This allows us to rank order these entries according to the rank of the rotamer in the rotamer table, but keep the scores below that of the previous case. But we still prefer this over mutating a residue, and thus a value of 1 is assigned, which ensures the score is in the range (1,2). Finally, if we cannot make an interaction with either the wildtype residues or any of their rotamers, we suggest a mutation that would provide the required interaction. This is scored the lowest as mutations carry the risk of destabilisation of the protein and thus the score is in the range (0,1). This scoring ensures that mutations are only selected if wildtype residues that can form the

interaction are not present. In addition, even within the set of wildtype residues, the ones that can form the required interactions in their native conformation are preferred.

Combinations of mutations were selected from this ranked list so that each interaction could be satisfied by a unique, non-clashing residue. These combinations were scored as a sum of rotamer scores, called combination scores. The combination scores were then used to rank the combinations of mutations. These combinations and the mutations to incorporate the TYG tripeptide at the structural match should add the function of fluorescence to the query protein.

Computation of torsion angles

Consider four points a,b,c and d. Let the torsion angle formed by these angles be θ.

$$\overrightarrow{v_1} = \vec{a} - \vec{b}$$

$$\overrightarrow{v_2} = \vec{b} - \vec{c}$$

$$\overrightarrow{v_3} = \vec{c} - \vec{d}$$

$$\overrightarrow{n_1} = \overrightarrow{v_1} \times \overrightarrow{v_2}$$

$$\overrightarrow{n_2} = \overrightarrow{v_2} \times \overrightarrow{v_3}$$

$$\theta_{magnitude} = \cos^{-1}\left(\frac{\overrightarrow{n_1} \cdot \overrightarrow{n_2}}{|\overrightarrow{n_1}||\overrightarrow{n_2}|}\right)$$

$$\overrightarrow{n_c} = \overrightarrow{n_1} \times \overrightarrow{n_2}$$

$$s = \overrightarrow{v_2} \cdot \overrightarrow{n_c}$$

$$\theta = \begin{cases} \theta_{magnitude}, & s \leq 0 \\ -\theta_{magnitude}, & s > 0 \end{cases}$$

Identification of interactions and clashes

We identify hydrogen bonds, π-stacking, cation-π, and salt bridges. The following methods were used to identify these interactions.

- Hydrogen bonds:
    - Donors and acceptor atoms in between 2.5 Å and 3.5 Å
    - Donor acceptor antecedent angle in range 90º-180º
- π-stacking

    - A π system is calculated, consisting of the centroid of the ring's atoms and the plane in which any three of the ring atoms occupy. The atoms used for different residues (including chromophore) are listed in Appendix section 4.
    - Parallel stacking
        - Distance between centroids of π systems in the range 3.5 Å and 4.5 Å
        - Angle between perpendiculars of the π planes in range 0º-30º
    - Perpendicular stacking
        - Distance between centroids of π systems in the range 4.5 Å and 5.5 Å
        - Angle between perpendiculars of the π planes in range 60º-90º
- Cation-π
    - π systems calculated using the same process as π-stacking
    - Angle between the perpendicular of the π plane and the vector between the centroids of the π system and the cation < 30º
    - Distance between centroids of π system and the cation atom < 5 Å.
- Salt bridges
    - Maximum distance = 3 Å between a donor and an acceptor atom, AND not a hydrogen bond

The list of donor and acceptor atoms, the list of atoms that form π systems and the list of charged atoms for each residue are listed in the Appendix sections 3, 4 and 5, respectively.

A clash is detected if two atoms are less than 2.5 Å apart.

Generating output structures

The corresponding rotamers were placed at the identified positions of mutations. MODELLER (Šali and Blundell 1993) was used to generate energy-minimised models. The chromophore was misformed after minimisation and was replaced after this step.

Molecular dynamics simulations

Molecular dynamics simulations were performed using GROMACS 2021 (Abraham et al. 2015; Pronk et al. 2013). The input files were generated using CharmmGUI

webserver using charmm36m forcefield (Brooks et al. 2009; Huang et al. 2017; Jo et al. 2008). The simulations were performed for 100 ns.

GROMACS command trjconv was used to correct for periodic boundary conditions and then superimpose the molecules over each other. The same program was used to convert the compressed trajectories to an ensemble of PDB format. Hydrogen bonds were then quantified for each frame in the trajectory.

Preparing DH10B competent cells for heat shock (Chemically competent cells)

Prerequisites:

1. Glycerol stocks of DH10B
2. LB media (50 ml + 3 ml)
3. 0.1M $CaCl^2$ (chilled) (60 ml)
4. 0.1M $CaCl_2$+15%glycerol (300-500 µl)
5. Sterile microcentrifuge tubes
6. 250 ml conical flask
7. Falcon tubes to grow 3 ml cultures
8. Shaker incubator at 37C
9. Centrifuge for 3000G spin at 4 ºC
10. -80 ºC storage
11. Laminar flow hood

Steps:

1. Pipette 10 µl of glycerol stock to 3 ml LB media and grow overnight.
2. Inoculate 50 ml of LB media (in a 250 ml flask, for aeration) with 50 µl(1%) of the previously grown 3 ml culture.
3. Incubate on a shaker for 1.5-2hrs at 37 ºC till cells are visible against light while swirling the flask.
4. Transfer into culture into 50 ml falcon.

All subsequent steps are to be performed on ice.

5. Centrifuge at 3000G for 10 mins at 4 ºC and discard media.
6. Resuspend the pellet in 20 ml chilled 0.1M $CaCl_2$.
7. Incubate on ice for 15 mins.

8. Centrifuge at 3000G for 10 mins at 4 ºC and discard supernatant.
9. Wash pellet with 20 ml of 0.1M $CaCl_2$.
10. Centrifuge at 3000G for 10 mins at 4 ºC and discard supernatant.
11. Wash pellet with 20 ml of 0.1M $CaCl_2$.
12. Centrifuge at 3000G for 10 mins at 4 ºC and discard supernatant.
13. Resuspend the pellet in 300-500 µl of 0.1M $CaCl_2$+15% glycerol.
14. Aliquot 20-50 µl to pre-chilled microcentrifuge tubes, depending on the pellet size.
15. Freeze and store at -80 ºC.

Misc info:

50 ml LB culture gives ~15 vials (30 µl each) of competent cells.

Preparing DH10B competent cells for Electroporation (Electro competent cells)

Prerequisites

1. 6 ml LB media.
2. 500 ml LB media.
3. 200 ml of autoclaved MilliQ water (deionised water).
4. 1.5 ml tubes.
5. 50 ml flacons for spin and wash steps.
6. 10% autoclaved glycerol.
7. Laminar flow hood

Steps

1. Grow 6 ml of primary culture overnight.
2. Transfer this to 500 ml LB media.
3. Incubate on a shaker for 1.5-2 hrs at 37 ºC till cells are visible against light while swirling the flask.

All subsequent steps are to be performed on ice.

4. Place the flask in ice for 15 mins
5. Pellet with 6000G for 15 mins at 4 ºC.
6. Wash with 100ml MilliQ.
7. Pellet with 6000G for 15 mins at 4 ºC.

8. Wash with 50ml MilliQ.
9. Pellet with 6000G for 15 mins at 4 ºC.
10. Wash with 2ml MilliQ.
11. Pellet with 6000G for 15 mins at 4 ºC.
12. Resuspend pellet in 10% glycerol final vol = 1.6 ml
13. Aliquote 50 µl, flash freeze and store at -80ºC.

Misc info:

The pellet should be white after washing. Yellow implies impurities from LB media that can lead to arcing in the electroporator.

Transformation of DH10B by heat shock

Prerequisites

1. Ice
2. DH10B chemical competent cells
3. Plasmid
4. Heating block/ water bath at 42 ºC
5. Incubator at 37 ºC
6. LA (or appropriate) selection plates with antibiotics corresponding to the plasmid.
7. 1 ml LB media per transformation.
8. Glass L rod, 100 % ethanol and lighter for platting and sterilisation (Tips or glass beads can also be used).

Steps

1. Thaw chemically-competent cells on ice.
2. Add a few nanograms of plasmid to competent cells.
3. Tap 10 times and mix.
4. Incubate on ice for 15 mins without any disruption.
5. Keep at 42 ºC for 1min 30 sec and then immediately back to ice (heat shock).
6. Add 1 ml LB media to the competent cells and incubate at 37 ºC for 45 mins.
7. Pellet the cells by spinning at 4000G for 5 mins.
8. Discard the media by tipping the tubes.

9. Resuspend the cells in the remaining media.
10. Plate the cells in appropriate selection plates corresponding to the plasmid.
11. Grow the plates for 15 hours.

Transformation of DH10B by electroporation

Prerequisites

1. Ice
2. DH10B electrocompetent cells
3. Plasmid
4. Electroporation machine and matching cuvettes.
5. Incubator at 37ºC
6. LA (or appropriate) selection plates with antibiotics corresponding to the plasmid.
7. LB media
8. MilliQ water
9. Glass L rod, 100 % ethanol and lighter for platting and sterilisation (Tips or glass beads can also be used).

Steps

1. Thaw electrocompetent cells on ice.
2. Cool the cuvette by placing it on ice.
3. Dilute the competent cells inside the cuvette using Milli-Q water till the solution becomes transparent.
4. Add a few nanograms of plasmid to the cuvette.
5. Tap 10 times and mix.
6. Electroporate with the electroporator with 2kV voltages with other conditions based on cuvette type.
7. Add 1:1 LB media to the competent cells and incubate at 37 ºC for 45 mins.
8. Plate 100 µl of the culture in appropriate selection plates corresponding to the plasmid.
9. Grow the plates for 15 hours.

Optimization of induction of protein expression from pCA24N

pCA24N is an expression vector with the expression of the protein being regulated by a lac-T5 fusion promoter. This promoter is not leaky and is inducible by Isopropyl β-D-1-thiogalactopyranoside (IPTG), which targets the lac operon on the plasmid. A very high concentration of IPTG (1 mM) is commonly used for protein expression in media for protein purification. However, this concentration is unsuitable for our application since we aim to check the colonies' phenotype in LB agar plates. The minimal concentration of IPTG that can induce protein expression is ideal for preventing additional stress on growing cells. We used three concentrations of IPTG, 0.1mM, 0.01mM and 0.001mM, to find a concentration that induces FolA-GFP fusion proteins from pCA24n. We found that 0.1 mM IPTG could reliably induce the expression (Figure 3.4). The expression was strong enough that the green fluorescence was noticeable even by the eye.

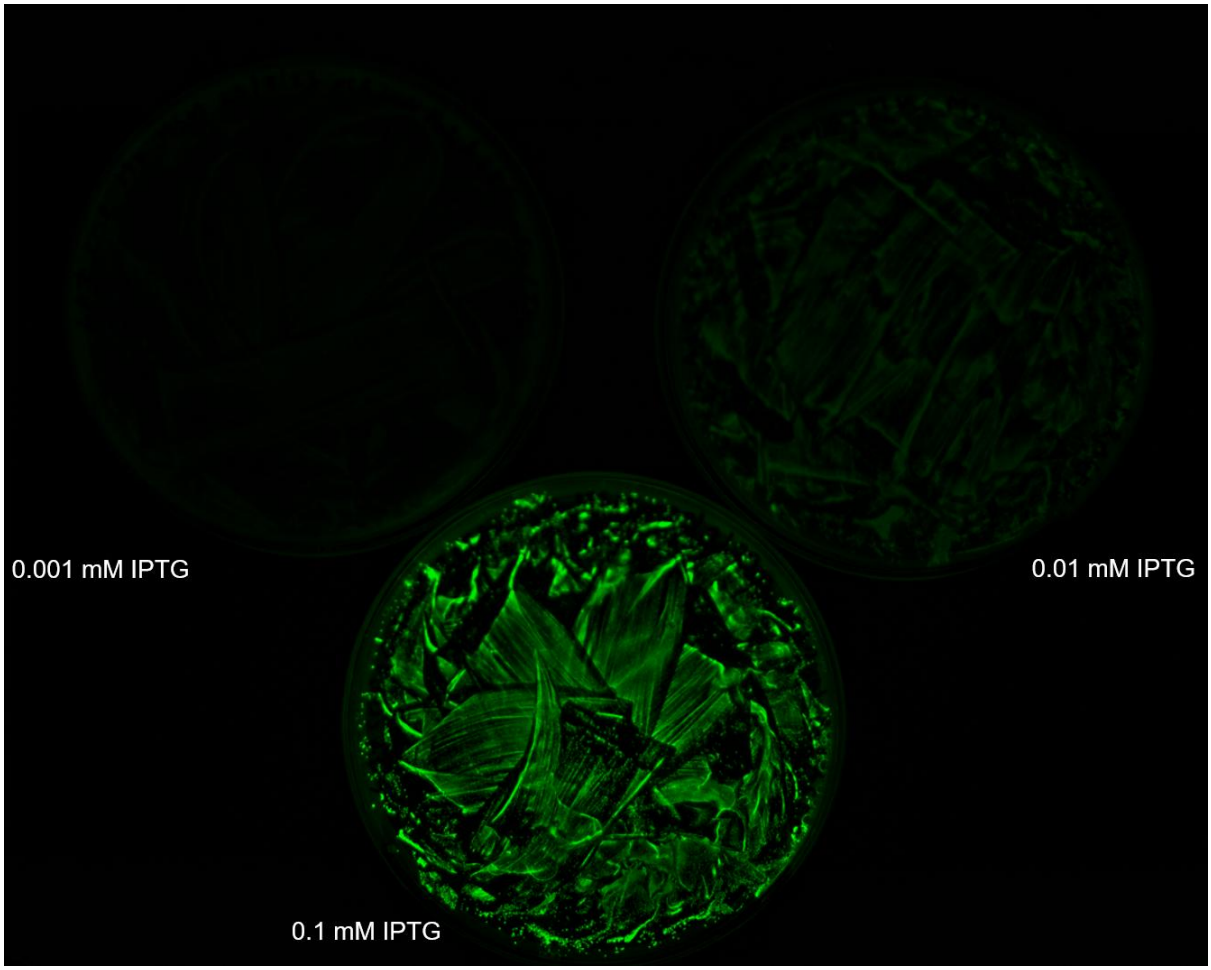


Figure 3.18: Optimisation of IPTG concentrations. The figure shows three plates with three different concentrations of IPTG, labelled in white next to the plates.

Results

Analysis of interactions of chromophores in the PDB

We could identify 418 structures of GFP and its mutants and variants in the PDB. Among them, only 319 contained the GFP chromophore (identified by residue codes CRO and GYS). Among these, only 312 structures contained the coordinates of the chromophores. A subset of these structures was selected with a resolution lower than 1.5 Å. This set contained 66 entries. A comprehensive list of such structures is shown in Appendix sections 1 and 2. The residue numbers in these structures did not match and had to be aligned and renumbered using the salign function of MODELLER (Madhusudhan et al. 2009; Webb and Sali 2016). The mapping of critical residue in PDB ID 1ema to the new numbering system is shown in Table 3.1.

Table 3.1: Mapping of numbers from 1ema to the renumbered residues

Number in 1ema	Unified number for all structures
ARG 96	ARG 101
GLN 94	GLN 99
HIS 148	HIS 154
THR 203	THR 222
GLU 222	THR 241
VAL 61	VAL 68

We analysed the average frequency of interactions made by the interaction moieties in the chromophores in these 66 structures (Table 3.2). The average frequencies of cation π and π stacking interaction are very low compared to hydrogen bonds. Salt bridge interactions were not found in the 66 structures. Based on this analysis, we have concluded that only hydrogen bonding is essential for the GFP chromophore. A shorter list with residue numbers corresponding to 1ema and with a ball and stick representation of the chromophore is shown in Figure 3.5. Both the hydrogen bonding nitrogen atoms, N1 and N2 have a frequency of hydrogen bonds significantly lower than one. This implies that interactions made by N1 and N2 are not consistent across multiple structures and are not crucial for fluorescence. Moreover, the oxygen atoms OH, OG, O2 and O3 require 2, 2, 2 and 1 hydrogen bonds, respectively, totalling seven interactions.

Table 3.2: Hydrogen bonds between chromophore and residues in its neighbourhood. The table shows the average frequency of hydrogen bonds for each hydrogen bonding atom in the chromophore. This is further detained in sub-tables showing the residue numbers contributing to this value, its fractional contribution, residue types found at those positions, and the atom with which the residues make the hydrogen bonds. Similar tables are also shown for salt bridges and cation π interactions

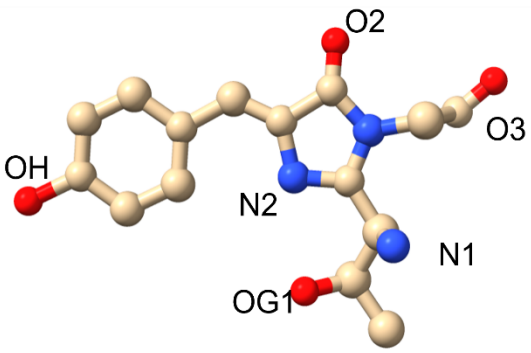
HYDROGEN BONDS				
Hydrogen bonds for N1 atom				
Frequency Per Structure = 0.770				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atom
A68	44	0.721	VAL,SER	O
B68	2	0.033	VAL	O
A152	1	0.016	VAL	O
Hydrogen bonds for OG1 atom				
Frequency Per Structure = 2.246				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atom
A241	47	0.770	GLN,GLU	OE2,OE1,NE2
A68	44	0.721	THR,VAL,SER	O
WATER	34	0.557	DOD,HOH	O
A71	4	0.066	PHE	O,N
A241	47	0.770	GLN,GLU	OE2,OE1,NE2
A68	44	0.721	THR,VAL,SER	O
WATER	34	0.557	DOD,HOH	O
A71	4	0.066	PHE	O,N
A48	2	0.033	HIS,GLN	NE2
B68	2	0.033	VAL	O
B241	2	0.033	GLU	OE1
A152	1	0.016	VAL	O
A238	1	0.016	GLN	NE2
Hydrogen bonds for OH atom				
Frequency Per Structure = 3.984				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atom
WATER	128	2.098	DOD,HOH	O
A222	56	0.918	THR	OG1
A154	38	0.623	HIS	ND1,NE2
A153	11	0.180	SER	OG

B222	4	0.066	THR	OG1
A150	2	0.033	TYR	OH
B154	2	0.033	HIS	ND1
A152	2	0.033	SER	OG
Hydrogen bonds for O2 atom				
Frequency Per Structure = 2.000				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atom
A101	57	0.934	ARG	NH1,NH2
A99	50	0.820	GLN	NE2
WATER	5	0.082	HOH	O
A74	4	0.066	ARG,GLN,LYS	NZ,NH1,NE2
B99	2	0.033	GLN	NE2
B101	2	0.033	ARG	NH2
A196	1	0.016	GLN	NE2
A198	1	0.016	ARG	NH2
Hydrogen bonds for O3 atom				
Frequency Per Structure = 2.000				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atom
WATER	91	1.492	DOD,HOH	O
A126	11	0.180	ASN	ND2
A99	2	0.033	TRP	NE1
A115	1	0.016	GLN	NE2
B126	1	0.016	ASN	ND2
Hydrogen bonds for N2 atom				
Frequency Per Structure = 2.000				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atom
WATER	28	0.459	DOD,HOH	O
PI STACKING INTERACTIONS				
Pi stacking interactions for atoms CB2, CE1, CE2, CD1, CD2, CZ				
Frequency Per Structure = 0.131				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atoms

A154	3	0.049	HIS	CG,NE2,CE1,CD2,ND1
A222	2	0.033	TYR	CG,CE1,CE2,CD1,CD2,CZ
A243	2	0.033	HIS	CG,NE2,CE1,CD2,ND1
A218	1	0.016	HIS	CG,NE2,CE1,CD2,ND1
Pi stacking interactions for atoms CB2, CE1, CE2, CD1, CD2, C				
Frequency Per Structure = 0.016				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atoms
A 218	1	0.016	HIS	CG, NE2, CE1, CD2, ND1
CATION PI				
Pi stacking interactions for atoms CB2, CE1, CE2, CD1, CD2, CZ				
Frequency Per Structure = 0.016				
Residue id	Frequency	Frequency Per Structure	Residue Name	Atoms
A243	1	0.016	HIS	NE2

OH		
Res No.	$\bar{\nu}$	**R**
Total	3.984	-
WATER	2.098	-
203	0.918	THR
148	0.623	HIS
147	0.180	SER

N2		
Res No.	$\bar{\nu}$	**R**
Total	0.459	-
WATER	0.459	-



OG1		
Res No.	$\bar{\nu}$	**R**
Total	2.246	-
222	0.770	GLN, GLU
61	0.721	THR, VAL, SER
WATER	0.557	-

N1		
Res No.	$\bar{\nu}$	**R**
Total	0.770	-
61	0.754	VAL, SER

O2		
Res No.	$\bar{\nu}$	**R**
Total	2.000	-
96	0.934	ARG
94	0.820	GLN

O3		
Res No.	$\bar{\nu}$	**R**
Total	1.738	-
WATER	1.492	-
121	0.180	ASN

Figure 3.19: Hydrogen bonding interactions made by hydrogen bond donors and acceptors of GFP chromophore (shown in ball and stick representation.). $\bar{\nu}$ is the average frequency of hydrogen bonds. R is the residues that make the hydrogen bonds.

Analysis of catalytic residues for chromophore formation

Elizebeth Getzoff and coworkers solved structures of GFP (S65T variant with enhanced fluorescence) in various stages of cyclisation to deduce the mechanism of chromophore formation. The structures of these mutants are described in Table 3.3. The key catalytic residue is arginine 96. The mutants of R96 take 3 months to cyclise at 37°C. Another critical residue is glutamate 222. Note that both these residues form consistent hydrogen bonds in Table 3.2 and Figure 3.5. We incorporate the catalytic residues in our program by restricting the necessary hydrogen bonds identified in the previous section to be from either of these residues.

The chromophore formation also requires a favourable main chain conformation for the TYG tripeptide. This initial conformation of the main chain can be observed as uncyclised TYG motif in the Getzoff mutant 1QXT and 1QY3, and uncyclised mutated motif GGG in 1QYO (see Table 3.3). The structures 1QXT and 1QY3, however, have an environment mutation of R96A (R96 is essential for efficient cyclisation) that forms an empty space occupied by Y66, a conformation that is otherwise not available in the wildtype. In these structures, the main chain conformation of Y66 is different from that of its corresponding atoms in the cyclised chromophore. We found that the atoms (all main chain) of the GGG motif in 1QYO match the corresponding positions of the chromophore. In addition, the conformation of the key residues (listed above) in chromophore formation and stabilisation is conserved in 1QYO. Thus, we selected the backbone motif of 1QYO as our template for identifying engineering candidate structures.

Table 3.3: State of GFP chromophore in various chromophore formation mutants (Barondeau et al. 2003)

PDB	Mutation	Chromophore state	Comments
1QXT	R96A	Precyclised	TYG tripeptide in precyclised state. It has a different sidechain orientation from the AlphaFold structure of the wild type.
1QY3	R96A	Precyclised	Variant of the same mutant as 1QXT. Have similar sidechain positions to 1QXT
1QYF	R96A	Cyclised	A cyclised version of 1QXT and 1QY3. It's almost to WT, except for the missing R96.

			Chromophore maturation was achieved by incubation for 3 months at 37°C.
1QYO	T65G, Y66G	Precyclised	Anaerobic precyclised structure with GGG in place of TYG. The catalytic residues are not changed. RMSD of the backbone is 0.5 Å to that of 1QXT
1QYQ	T65G, Y66G	Cyclized	Cyclised GGG that forms a five-membered ring in place of the chromophore.

Search of RCSB PDB

We used our method to search 200199 structures from RCSB PDB. We found 12573 proteins that can be fluorescent upon mutation. Of these, 12513 were non-NMR structures. From this, 26 proteins were selected manually based on the size of the proteins, the number of interactions satisfied by native residues, changes in the number of atoms before and after mutation and repetition of related structures in the dataset.

We picked the following proteins from this list based on the following properties:

1. A combination of the number of native interactions, chain length and difference in number of atoms. We picked proteins with less than 200 amino acids, which only require two mutations in addition to the tripeptide, and the total change in the number of atoms is less than 5. We found the following structures.
    - 5 structures of cellular retinol-binding proteins
        - 1CBR, 1CBS, 2FR3, 2G79, 5OGB
    - 4 structures of carbohydrate-binding proteins
        - 2Y6G, 2Y6H, 2Y6J, 2Y6L
    - 4 structures from other proteins with only one hit each
        - 3K2Y, 3N2E, 4C8T, 7CG9
2. By protein size. We picked proteins with less than 90 amino acids. We found the following structures.
    - 3 structures from the SH3 domain
        - 1ZX6, 5NXJ, 6CQ7
    - 3 structures of MDM2 bound to inhibitors.
        - 4OGV, 6GGN, 7NUS

- 4 structures from other proteins with only one hit each.
  - 5J8Y, 6NZU, 6PSD, 6VYN

3. Less than one mutation (excluding tripeptide) and less than 300 amino acids in length
   - 1LM7, 3N2E and 7CG9

We further reduced the number of candidates by checking if the introduced mutations reduced the number of interactions compared to the wild-type query protein. The hits that satisfy this criterion are shown in Figure 3.6.

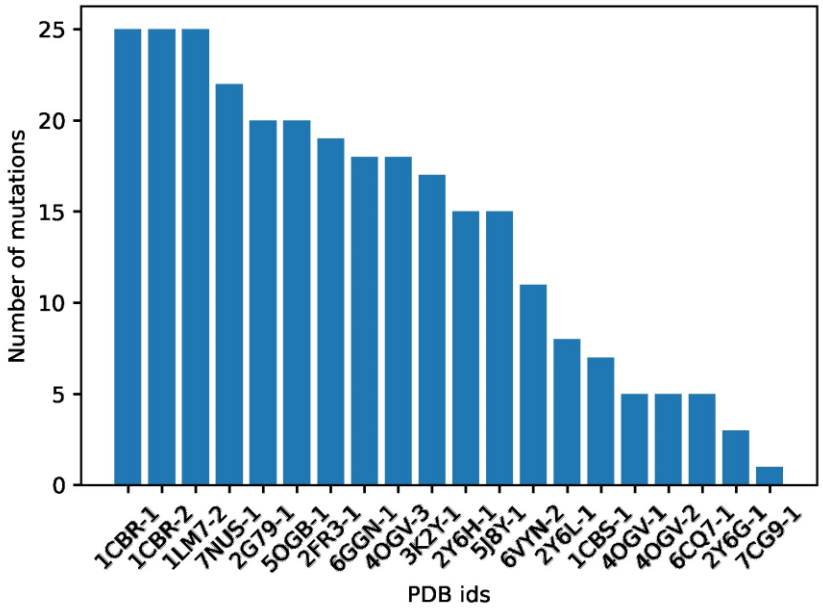


Figure 3.20: Number of combinations of mutations in different hits that do not reduce the number of interactions. -axis shows the PDB IDs of the hits and an identifier that specifies the structural match. Y-axis shows the number of combinations of mutations corresponding to the structure match that do not reduce the number of interactions in the proteins.

These structures were manually inspected along with their wildtypes to create candidates of different failure levels. These candidates were also categorised based on their sequence similarity.

1. Low risk
   - Carbohydrate-binding proteins
     - 2Y6L, 2Y6G, 2Y6H (apo)
     - Around five native interactions
   - 1LM7, Intermediate filament binding fragment of desmoplakin
     - 7 native interactions

        - But no hit on homologs
        - Possibility of disulphide bond (no disulphide in structure)
    - Retinol binding proteins:
        - 1CBR, 2G79, 5OGB, 2FR3, 5OGB, 1CBS
        - Multiple structural matches
        - Around five native interactions
        - Buried chromophore
        - Positive gain in number of interactions (8 in some cases)
        - Apo structures cannot form
2. Medium risk
    - MDM 2
        - 6GGN (lower resolution at 2.0 A), 4OGV
        - 3 structures from two groups have structure match at the same position.
        - 0-2 native contacts
    - SH3 domains
        - 6CQ7
        - Multiple structural matches at the same site from different proteins
        - But only one PDB does not lose interaction
        - 2-3 native contacts
3. High risk
    - 3K2Y
        - 5 native interactions
        - SH3 domain
        - Different structure match from other SH3 hits
        - CB clash with chromophore.

Molecular dynamic simulations of mutant models

The models of mutants of the selected hits were further analysed using molecular dynamic (MD) simulations. The trajectories of MD simulations were analysed to check if the chromophore forms the predicted hydrogen bonds. We analysed the number of interactions the chromophore forms at each simulation frame. These trajectories of this number are then plotted vs simulation time for analysis.

It is necessary to have a control to compare the trajectories as different parameters in molecular dynamics simulations can have differing outcomes, particularly with special residues like the chromophore. For this purpose, we use trajectories obtained from GFP (PDB ID 1ema) (Figure 3.7). The trajectories of three replicates conclusively show that the interactions made by OH and O2 atoms are paramount. The interactions by these atoms are seldom broken (red and green traces in Figure 3.7 A, B and C). Interactions by O3 were surprisingly rarely observed. We then conducted an identical analysis for one representative for each family we identified. The model of a mutation combination was used for this purpose. These are 2y6h for carbohydrate-binding protein, 1lm7 the intermediate filament binding fragment of desmoplakin, 6ggm for MDM2, 6cq7 for SH3 domain and 3k2y an uncharacterised protein. Retinol-binding proteins were excluded from our analysis as they could not be parametrised using the same approach.

Our analysis shows that only 1lm7 and 3k2y show consistent hydrogen bonds by OH and O2 the chromophore (Figure 3.9 and Figure 3.12). In contrast, 2y6h, 6ggm and 6cq7 show a minimal number of hydrogen bonds, which are unfortunately inconsistent (Figure 3.8, Figure 3.10 and Figure 3.11). For this reason, we have selected 1lm7 and 3k2y for our experimental validation. In addition to these two, we have added retinol-binding proteins to this list due to its many recurring hits.

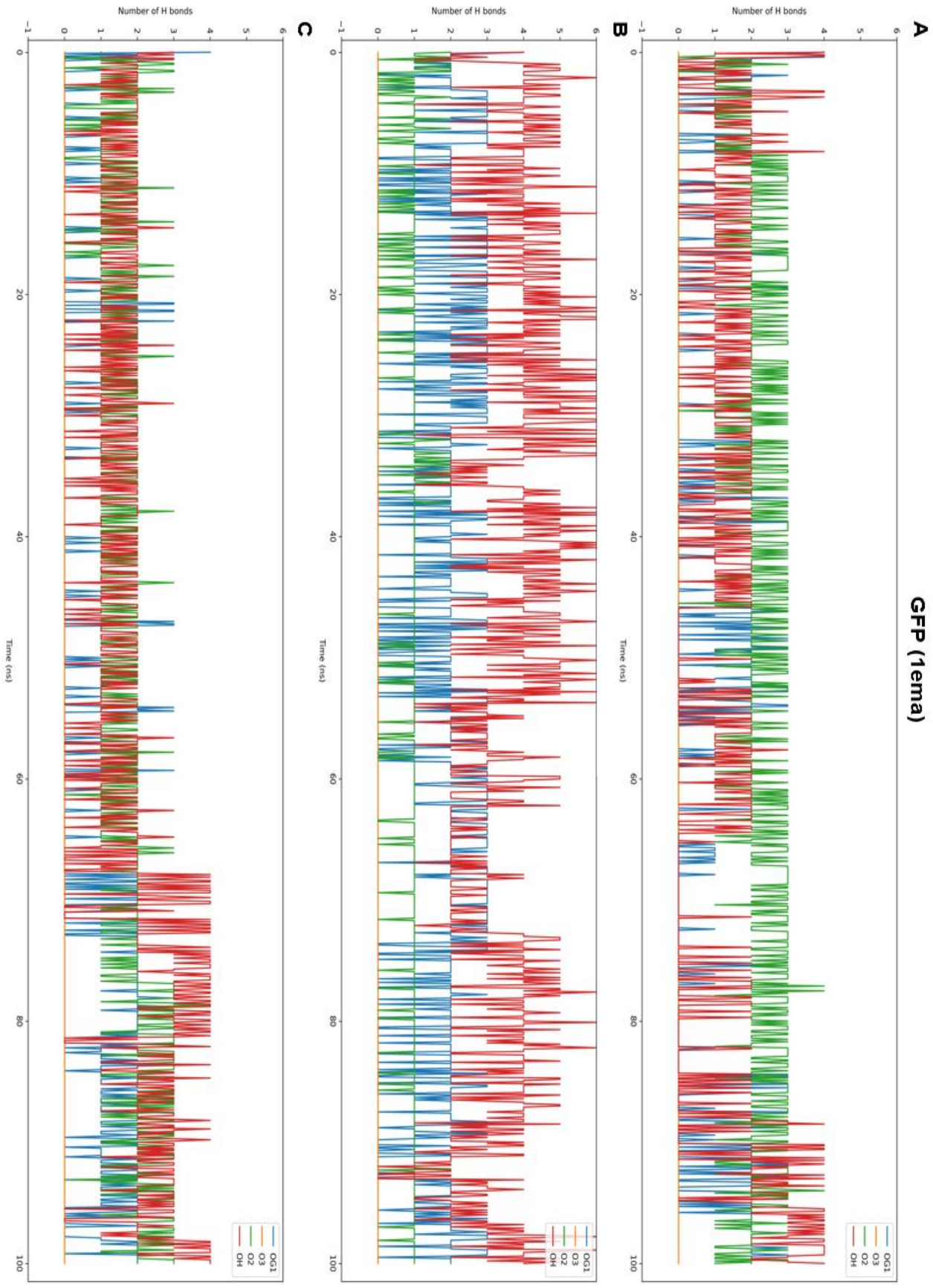


Figure 3.21: Trajectories of the number of interactions by each hydrogen bonding oxygen bonding atoms in the chromophore for GFP (PDB ID 1EMA). Panels A, B and C show three different replicates. Blue, orange, green and red show the trajectories for OG1, O3, O2 and OH respectively.

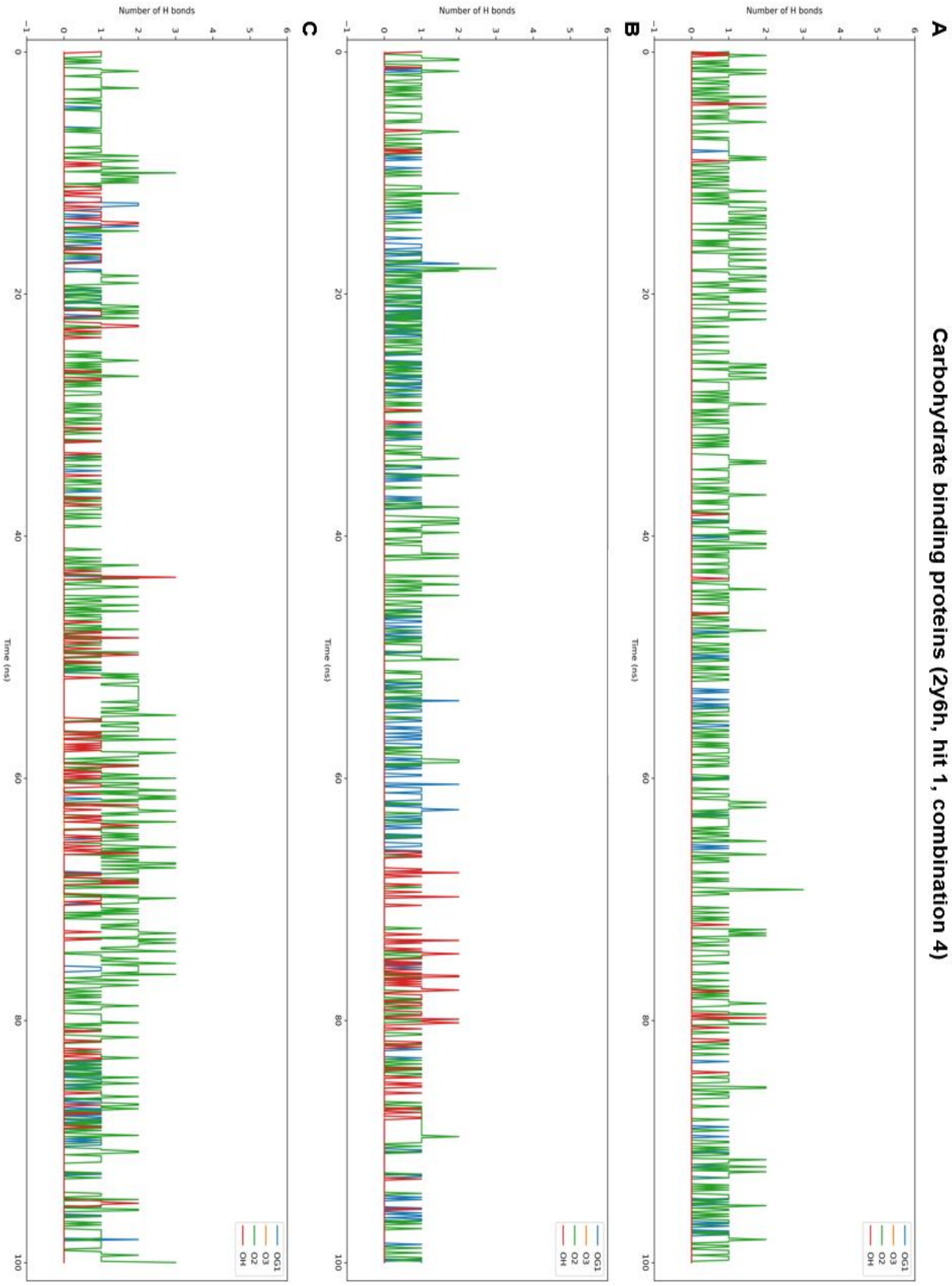


Figure 3.22: Trajectories of number of interactions by each hydrogen bonding oxygen bonding atoms in the chromophore for carbohydrate binding proteins with PDB ID 2y6h, mutated based on its mutation combination 4. Panels A, B and C show three different replicates. Blue, orange, green and red show the trajectories for OG1, O3, O2 and OH respectively.

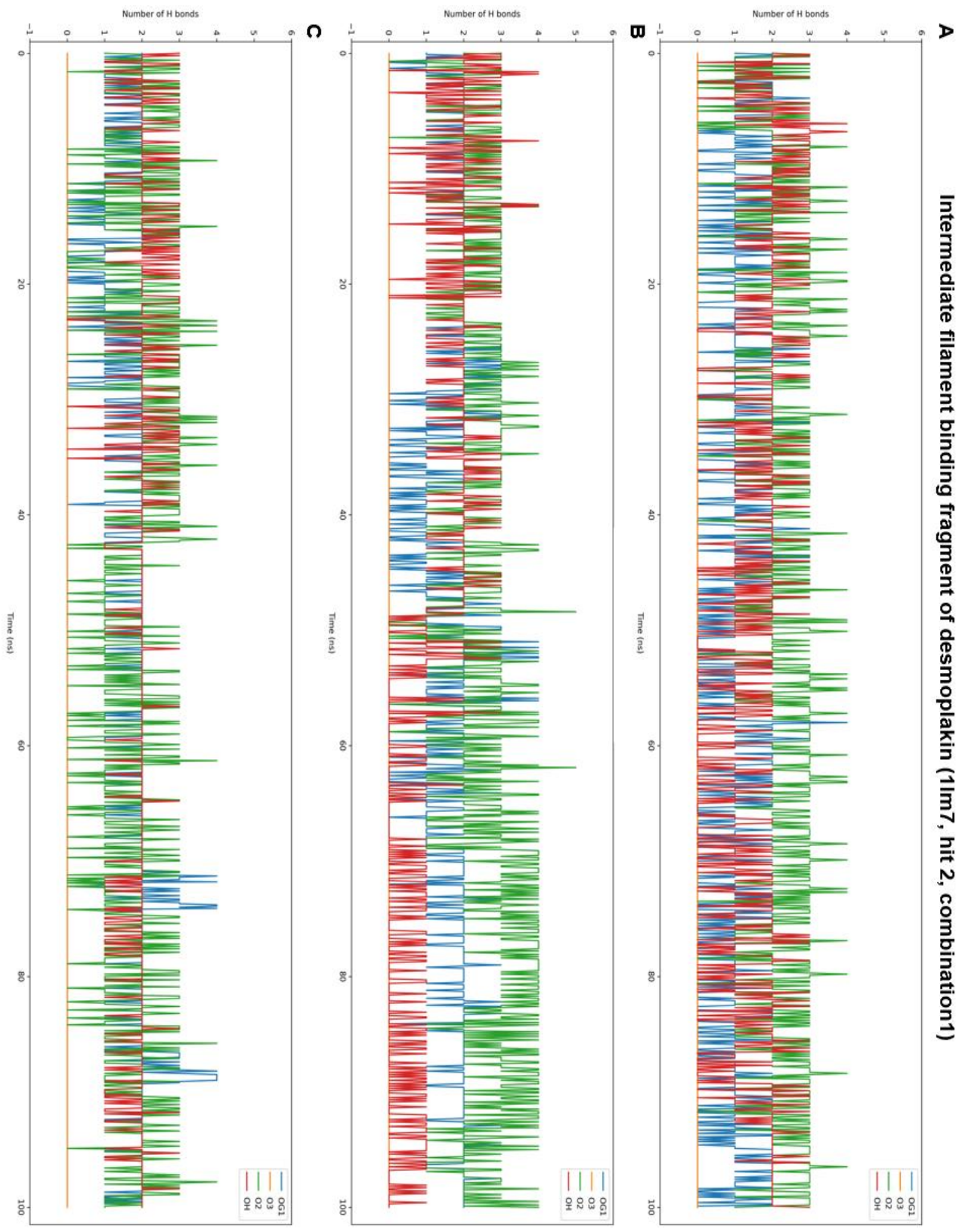


Figure 3.23: Trajectories of the number of interactions by each hydrogen bonding oxygen bonding atoms in the chromophore for intermediate filament binding fragment of desmoplakin with PDB ID 1lm7, mutated based on its structural match 2 and mutation combination 1. Panels A, B and C show three

different replicates. Blue, orange, green and red show the trajectories for OG1, O3, O2 and OH respectively.

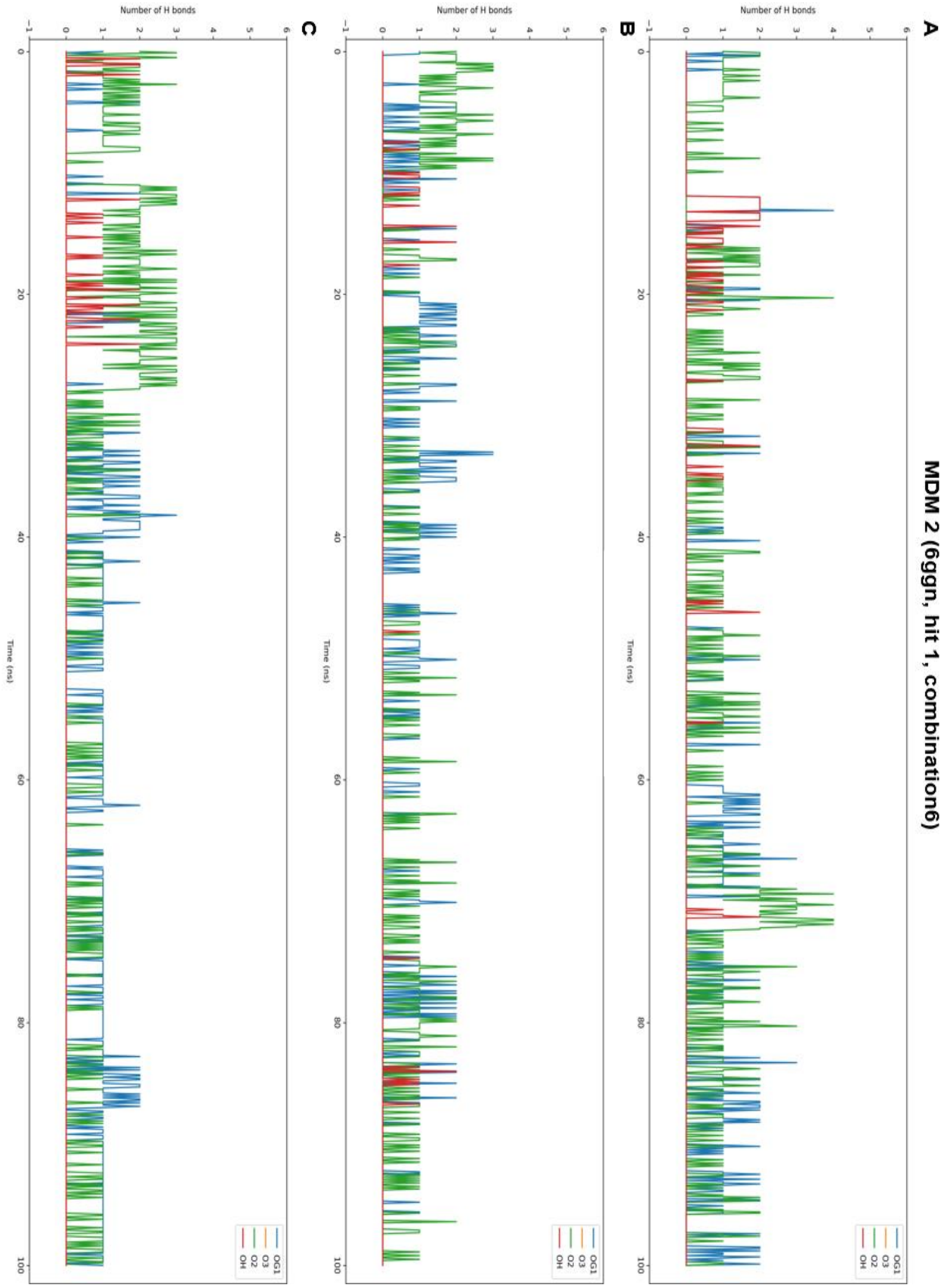


Figure 3.24: Trajectories of the number of interactions by each hydrogen bonding oxygen bonding atoms in the chromophore for MDM2 protein with PDB ID 6ggn, mutated based on its structural match 1 and mutation combination 6. Panels A, B and C show three different replicates. Blue, orange, green and red show the trajectories for OG1, O3, O2 and OH respectively.

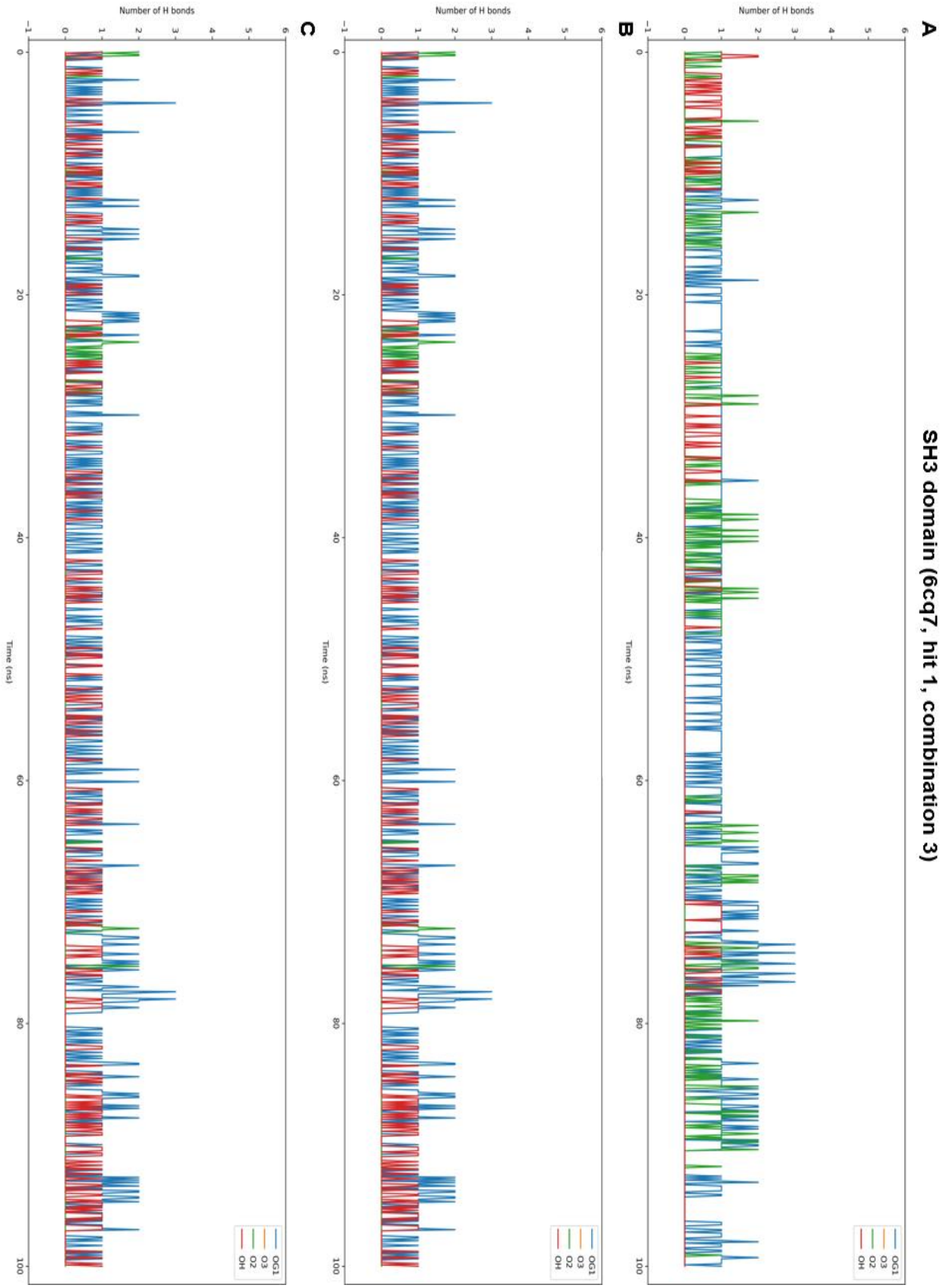


Figure 3.25: Trajectories of the number of interactions by each hydrogen bonding oxygen bonding atoms in the chromophore for SH3 domain protein with PDB ID 6cq7, mutated based on its structural

match 1 and mutation combination 3. Panels A, B and C show three different replicates. Blue, orange, green and red show the trajectories for OG1, O3, O2 and OH respectively

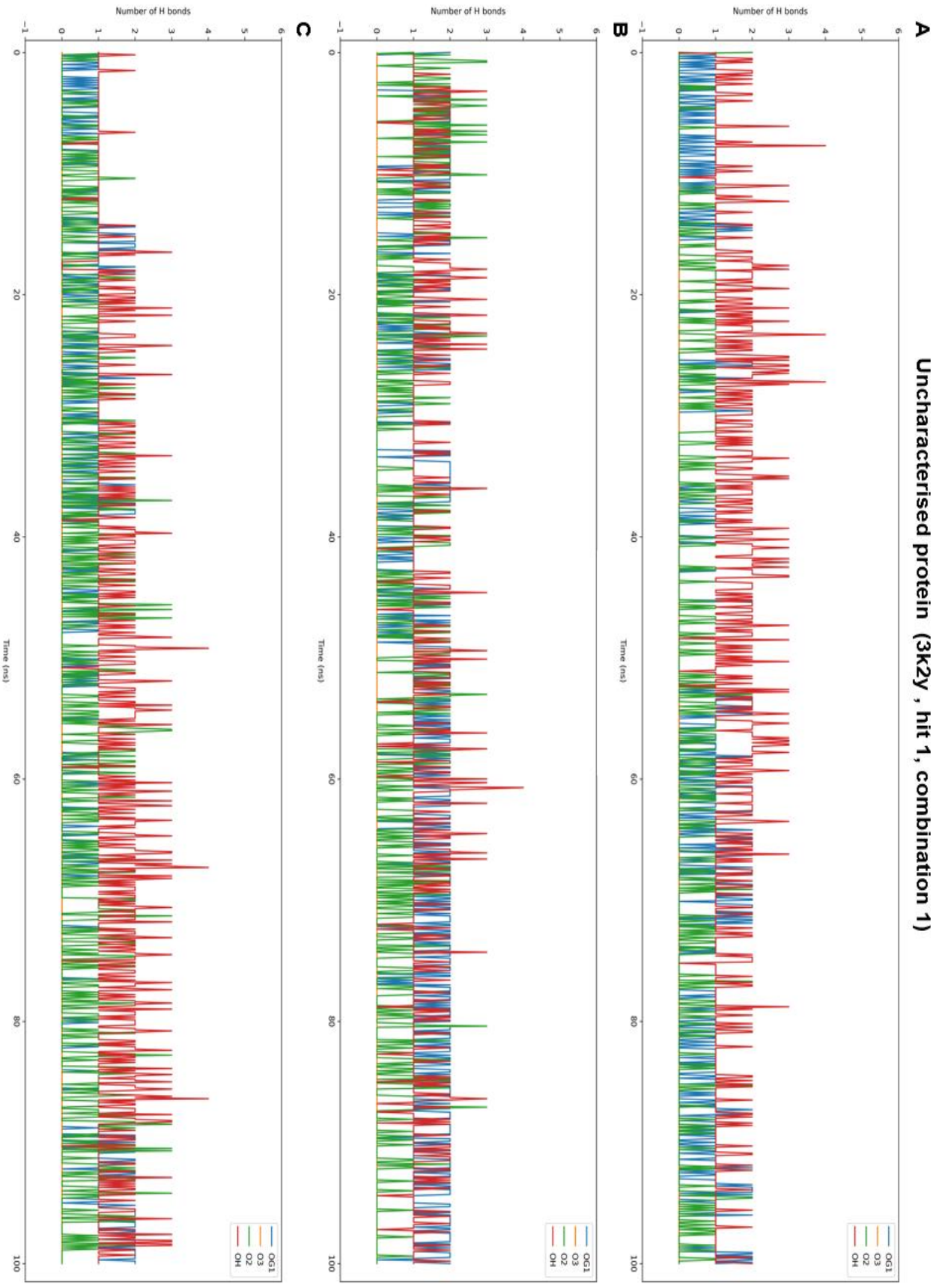


Figure 3.26: Trajectories of the number of interactions by each hydrogen bonding oxygen bonding atoms in the chromophore for the uncharacterised protein with PDB ID 3k2y, mutated based on its structural match 1 and mutation combination 1. Panels A, B and C show three different replicates. Blue, orange, green and red show the trajectories for OG1, O3, O2 and OH respectively.

Generating sequences for mutational libraries from combinations of mutations

Our program generates 25 combinations of mutations (as described in the section 'Identification of interacting residues') for each hit. A practical number of 25 was chosen due to storage limitations. Each combination has 8 output files. Thus 12573 identified structures generate in excess (some structures have multiple hits) of 2,514,600 files (12573*25*8). In addition, lower ranking combinations would require each larger number of lower ranking mutations, increasing the risk of destabilisation (see section 'Identification of interacting residues'). The combinations for 1lm7, 1cbs and 3k2y are shown in Tables 3.4, 3.5 and 3.6, respectively.

Table 3.4: List of fluorescence-inducing mutations for 1lm7.

Unique Key	Combination id	Number of interactions by native residue	Mutations
1LM7-2-1	1	7	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-2	2	6	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-3	3	6	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-4	4	6	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; PRO(A2311)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-5	5	6	GLU(A2293)GLU, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-6	6	6	GLU(A2293)ASN, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-7	7	5	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-8	8	5	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; PRO(A2311)ARG, THR(A2337)THR; GLY(A2338)GLY

1LM7-2-9	9	5	GLU(A2293)ASP, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-10	10	6	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)SER; GLY(A2338)GLY
1LM7-2-11	11	6	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)TRP
1LM7-2-12	12	5	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; PRO(A2311)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-13	13	6	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-14	14	5	GLU(A2293)GLU, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-15	15	5	GLU(A2293)GLU, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; PRO(A2311)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-16	16	5	GLU(A2293)ASN, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-17	17	5	GLU(A2293)ASN, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; PRO(A2311)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-18	18	5	GLU(A2293)ASN, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-19	19	5	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)SER; GLY(A2338)GLY
1LM7-2-20	20	5	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)TRP
1LM7-2-21	21	4	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; PRO(A2311)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-22	22	5	GLU(A2293)ASP, LEU(A2329)LEU; ILE(A2301)ILE, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; GLY(A2338)GLY

1LM7-2-23	23	4	GLU(A2293)ASP, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; ARG(A2214)ARG, THR(A2337)THR; TYR(A2339)GLN
1LM7-2-24	24	4	GLU(A2293)ASP, LEU(A2329)LEU; LEU(A2310)LEU, ALA(A2315)ASP; PRO(A2311)ARG, THR(A2337)THR; GLY(A2338)GLY
1LM7-2-25	25	6	GLU(A2293)GLU, LEU(A2329)LEU; ILE(A2301)ILE, LEU(A2310)LEU; ARG(A2214)ARG, THR(A2337)CYS; GLY(A2338)GLY

Table 3.5: List of fluorescence-inducing mutations for 1cbs.

Unique Key	**Combination id**	**Number of interactions by native residue**	**Mutations**
1CBS-1-1	1	5	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-2	2	4	ALA(A 32)ALA, THR(A 56)GLU; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-3	3	5	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-4	4	4	ALA(A 32)ALA, THR(A 56)THR; VAL(A 76)VAL, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-5	5	4	ALA(A 32)ALA, THR(A 56)GLU; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-6	6	3	ALA(A 32)ALA, THR(A 56)GLU; VAL(A 76)VAL, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-7	7	5	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, VAL(A 76)VAL; ILE(A 9)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-8	8	5	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, VAL(A 76)VAL; SER(A 12)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-9	9	5	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)TYR
1CBS-1-10	10	4	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG

1CBS-1-11	11	4	ALA(A 32)ALA, THR(A 56)THR; VAL(A 76)VAL, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-12	12	4	ALA(A 32)ALA, VAL(A 58)ASP; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-13	13	4	ALA(A 32)ALA, THR(A 56)GLU; ARG(A 59)ARG, VAL(A 76)VAL; ILE(A 9)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-14	14	4	ALA(A 32)ALA, THR(A 56)GLU; ARG(A 59)ARG, VAL(A 76)VAL; SER(A 12)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-15	15	4	ALA(A 32)ALA, THR(A 56)GLU; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)TYR
1CBS-1-16	16	3	ALA(A 32)ALA, THR(A 56)GLU; ARG(A 59)ARG, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-17	17	3	ALA(A 32)ALA, THR(A 56)GLU; VAL(A 76)VAL, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-18	18	4	ALA(A 32)ALA, THR(A 56)SER; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-19	19	5	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, VAL(A 76)VAL; SER(A 12)ARG, ARG(A 132)ARG; ILE(A 9)TYR
1CBS-1-20	20	4	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)ARG, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-21	21	4	ALA(A 32)ALA, THR(A 56)THR; ARG(A 59)GLU, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-22	22	4	ALA(A 32)ALA, THR(A 56)THR; VAL(A 76)VAL, GLY(A 78)TYR; ILE(A 9)ARG, ARG(A 132)ARG; SER(A 12)ARG
1CBS-1-23	23	4	ALA(A 32)ALA, THR(A 56)THR; VAL(A 76)VAL, GLY(A 78)TYR; SER(A 12)ARG, ARG(A 132)ARG; ILE(A 9)ARG
1CBS-1-24	24	4	ALA(A 32)ALA, THR(A 56)THR; VAL(A 76)VAL, GLY(A 78)TYR; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)TYR
1CBS-1-25	25	4	ALA(A 32)ALA, VAL(A 58)ASP; ARG(A 59)ARG, VAL(A 76)VAL; PHE(A 15)ARG, ARG(A 132)ARG; ILE(A 9)ARG

Table 3.6: List of fluorescence-inducing mutations for 3k2y.

Unique Key	Combination id	Number of interactions by native residue	Mutations
3K2Y-1-1	1	5	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-2	2	4	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-3	3	5	ILE(C 22)ILE, ALA(C 58)TRP; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-4	4	5	ILE(C 22)ILE, THR(C 54)GLU; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-5	5	4	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)ARG; ARG(C 30)ARG
3K2Y-1-6	6	4	ILE(C 22)ILE, ALA(C 58)TRP; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-7	7	4	ILE(C 22)ILE, THR(C 54)GLU; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-8	8	4	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)ASN; ARG(C 30)ARG
3K2Y-1-9	9	4	ILE(C 22)ILE, ALA(C 58)HIS; LEU(C 60)LEU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-10	10	4	ILE(C 22)ILE, ALA(C 58)TRP; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)ARG; ARG(C 30)ARG
3K2Y-1-11	11	4	ILE(C 22)ILE, LEU(C 60)GLU; GLU(C 16)GLU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-12	12	4	ILE(C 22)ILE, THR(C 54)GLU; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)ARG; ARG(C 30)ARG
3K2Y-1-13	13	5	ILE(C 22)ILE, ALA(C 58)ASN; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG

3K2Y-1-14	14	4	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)LYS; ARG(C 30)ARG
3K2Y-1-15	15	4	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-16	16	3	ILE(C 22)ILE, ALA(C 58)HIS; LEU(C 60)LEU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-17	17	4	ILE(C 22)ILE, ALA(C 58)TRP; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)ASN; ARG(C 30)ARG
3K2Y-1-18	18	4	ILE(C 22)ILE, ALA(C 58)TRP; LEU(C 60)LEU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-19	19	3	ILE(C 22)ILE, LEU(C 60)GLU; GLU(C 16)GLU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-20	20	4	ILE(C 22)ILE, THR(C 54)GLU; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)ASN; ARG(C 30)ARG
3K2Y-1-21	21	4	ILE(C 22)ILE, THR(C 54)GLU; LEU(C 60)LEU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-22	22	4	ILE(C 22)ILE, ALA(C 58)ASN; GLU(C 16)GLU, LEU(C 60)LEU; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-23	23	3	ILE(C 22)ILE, ALA(C 58)HIS; GLU(C 16)GLU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)CYS; ARG(C 30)ARG
3K2Y-1-24	24	4	ILE(C 22)ILE, ALA(C 58)HIS; LEU(C 60)LEU, GLY(C 61)ASP; LEU(C 88)ARG, GLN(C 90)GLN; ARG(C 30)ARG
3K2Y-1-25	25	3	ILE(C 22)ILE, ALA(C 58)HIS; LEU(C 60)LEU, GLY(C 61)HIS; LEU(C 88)ARG, GLN(C 90)ARG; ARG(C 30)ARG

If our program has perfect accuracy, generating any of the 75 mutants in Tables 3.4, 3.5 and 3.6 should result in a fluorescent protein. However, since that is unlikely, we aim to create a mutational library that samples all the mutations and their combinations predicted in the tables. For this purpose, we generate a new DNA sequence that combines all the amino acid mutations using degeneracy code. This is because the degeneracy codes (Table 3.7) can be used to synthesise DNA containing combinations

specified by the degeneracy. These predicted DNA sequences for 1cbs and 3k2y are shown in Table 3.8.

Notice that in top-ranking hits for 1lm7, all seven interactions can be satisfied by the native residues, and only the chromophore-forming tripeptide needs to be introduced. Thus, 1lm7 rank one was cloned manually.

Table 3.7: IUPAC codes for degenerate nucleotides

IUPAC code	**Bases**
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	A or T or G or C

Table 3.8: Sequences containing positions with degeneracy.

PDB ID	**Predicted degenerate DNA sequence with mutations encoded by degenerate sequence**
1cbs	TATGCCCAACTTCTCTGGCAACTGGAAA**H**DCATCCGA**M**GCGAAAAC**Y** **K**CGAGGAATTGCTCAAAGTGCTGGGGGTGAATGTGATGCTGAGGAAG ATTGCTGTGGCTACCTACGGTAAGCCAGCAGTGGAGATCAAACAGGA GGGAGACACTTTCTACATCAAAACCTCC**DM**AACCG**W**T**RR**AACCACAG AGATTAACTTCAAGGTTGGGGAGGAGTTTGAGGAGCAGACTGTGGAT **KR**CAGGCCCTGTAAGAGCCTGGTGAAATGGGAGAGTGAGAATAAAAT GGTCTGTGAGCAGAAGCTCCTGAAGGGAGAGGGCCCCAAGACCTCG TGGACCAGAGAACTGACCAACGATGGGGAACTGATCCTGACCATGAC GGCGGATGACGTTGTGTGCACCAGGGTTTCCGTCCGAAAGTAA
3k2y	ATGACAGGCGATGCAGCCGTCGCATTGGATACAGTCACGGTCGTTGG CGAACGGTATGTCGATGATATTGTGGCAACGTACGGTACGCTCAGAG TGGGTATGGCGGTGTTGCTCCAGCGAGAAAGTGGCAATCAATACGAT GATAATGCCATCTCAGTGTGG**RM**ATTGCAACAT**NVS**AAG**SW**G**SR**CTA CATTGCCCGGTATCAGAACCAGCCGTACGCGACACTGATGGATCAGG

	GGCAACGATTGTATGGCATCGTGACGGTACGTGAT**HRM**CAAAAACAG CATCTTGAATTGATGCTATGGCGG

Generating fragments for DNA synthesis

We need to fragment the sequences from Table 3.8 into smaller sections as the accurate synthesis of oligonucleotides is limited by length (120 bases as specified by Sigma Aldrich). We then use Golden Gate assembly (Engler, Kandzia, and Marillonnet 2008) to assemble these to generate full-length sequences. Golden gate assembly relies on unique complementary four-base pair overhangs that form after digestion of type IIs endonucleases. Although any four-base pair can be used for this purpose, all '4mers' do not have the same ligation efficiency due to mild sequence preferences of DNA ligases and non-Watson Crick base pairing. The efficiency and fidelity of all '4mers' have previously been determined, and based on this data, a program called SplitSet was developed to identify ideal break sites (Potapov, Ong, Kucera, et al. 2018; Potapov, Ong, Langhorst, et al. 2018; Pryor et al. 2020, 2022). SplitSet has various shortcomings that make it unsuitable for our work, and we address them with our method, Overhang Optimiser for Golden Gate Assembly (OOGGA) (Chapter 4). We used OOGGA to fragment the DNA sequences from Table 3.8 to generate the sequences shown in Table 3.9 and Table 3.10 for 1cbs and 3k2y, respectively. Note that these sequences also contain the adapter sequences for golden gate assembly. These fragments were purchased as oligonucleotides from Sigma Aldrich.

Table 3.9: Oligonucleotides predicted by OOGGA for golden gate assembly for 1cbs.

Oligos Names	**Oligo length**	**Oligo sequence 5' - 3'**
F1F	84	GGCTACGGTCTCCTATGCCCAACTTCTCTGGCAACTGGAAAHDCATCCGAM GCGAAAACYKCGAGGAATTGGGAGACCGTAGCC
F1R	84	GGCTACGGTCTCCCAATTCCTCGMRGTTTTCGCKTCGGATGHDTTTCCAGTT GCCAGAGAAGTTGGGCATAGGAGACCGTAGCC
F2F	95	GGCTACGGTCTCCATTGCTCAAAGTGCTGGGGGTGAATGTGATGCTGAGGA AGATTGCTGTGGCTACCTACGGTAAGCCAGCGGAGACCGTAGCC
F2R	95	GGCTACGGTCTCCGCTGGCTTACCGTAGGTAGCCACAGCAATCTTCCTCAGC ATCACATTCACCCCCAGCACTTTGAGCAATGGAGACCGTAGCC

F3F	73	GGCTACGGTCTCCCAGCAGTGGAGATCAAACAGGAGGGAGACACTTTCTAC ATCAAAACCGGAGACCGTAGCC
F3R	73	GGCTACGGTCTCCGGTTTTGATGTAGAAAGTGTCTCCCTCCTGTTTGATCTCC ACTGCTGGGAGACCGTAGCC
F4F	92	GGCTACGGTCTCCAACCTCCDMAACCGWTRRAACCACAGAGATTAACTTCA AGGTTGGGGAGGAGTTTGAGGAGCAGACGGAGACCGTAGCC
F4R	92	GGCTACGGTCTCCGTCTGCTCCTCAAACTCCTCCCCAACCTTGAAGTTAATCT CTGTGGTTYYAWCGGTTKHGGAGGTTGGAGACCGTAGCC
F5F	96	GGCTACGGTCTCCAGACTGTGGATKRCAGGCCCTGTAAGAGCCTGGTGAAA TGGGAGAGTGAGAATAAAATGGTCTGTGAGCAGGAGACCGTAGCC
F5R	96	GGCTACGGTCTCCTGCTCACAGACCATTTTATTCTCACTCTCCCATTTCACCAG GCTCTTACAGGGCCTGYMATCCACAGTCTGGAGACCGTAGCC

Table 3.10: Oligonucleotides predicted by OOGGA for golden gate assembly for 3k2y.

Oligos Names	**Oligo length**	**Oligo sequence 5' - 3'**
F1F	104	GGCTACGGTCTCCTATGACAGGCGATGCAGCCGTCGCATTGGATACAGTCA CGGTCGTTGGCGAACGGTATGTCGATGATATTGTGGCAACGGAGACCGTAG CC
F1R	104	GGCTACGGTCTCCGTTGCCACAATATCATCGACATACCGTTCGCCAACGACC GTGACTGTATCCAATGCGACGGCTGCATCGCCTGTCATAGGAGACCGTAGC C
F2F	94	GGCTACGGTCTCCCAACGTACGGTACGCTCAGAGTGGGTATGGCGGTGTTG CTCCAGCGAGAAAGTGGCAATCAATACGATGGAGACCGTAGCC
F2R	94	GGCTACGGTCTCCATCGTATTGATTGCCACTTTCTCGCTGGAGCAACACCGC CATACCCACTCTGAGCGTACCGTACGTTGGGAGACCGTAGCC
F3F	112	GGCTACGGTCTCCCGATGATAATGCCATCTCAGTGTGGRMATTGCAACATN VSAAGSWGSRCTACATTGCCCGGTATCAGAACCAGCCGTACGCGACACGGA GACCGTAGCC

F3R	112	GGCTACGGTCTCCGTGTCGCGTACGGCTGGTTCTGATACCGGGCAATGTAG YSCWSCTTSBNATGTTGCAATKYCCACACTGAGATGGCATTATCATCGGGAG ACCGTAGCC
F4F	119	GGCTACGGTCTCCACACTGATGGATCAGGGGCAACGATTGTATGGCATCGT GACGGTACGTGATHRMCAAAAACAGCATCTTGAATTGATGCTATGGCGGTA GCATGGAGACCGTAGCC
F4R	119	GGCTACGGTCTCCATGCTACCGCCATAGCATCAATTCAAGATGCTGTTTTTGK YDATCACGTACCGTCACGATGCCATACAATCGTTGCCCCTGATCCATCAGTGT GGAGACCGTAGCC

Library preparation by Golden Gate assembly

Golden gate assembly was performed using the BsaI golden gate assembly kit from New England Biolabs (NEB) using the protocol specified by the manufacturer. The fragments were introduced into the empty linearised vector pCA24n. 10 µl of the reaction was transformed using the heat shock method into LA Plates with chloramphenicol and 0.1 mM IPTG to select the transformants and to induce the expression, respectively. The plates were grown for 15 hours, and the colonies were imaged in an Invitrogen iBright 1500 with automatic exposure.

Our negative control is the empty vector, and our positive control is pCA24nFolA-GFP, which produces the fusion proteins containing GFP when induced with IPTG. The comparison between the FolA-GFP and negative control is shown in Figure 3.13, panel A. Notice that the negative control is almost invisible. The green colour is from the low intensity of auto-fluorescence from the plate and colonies. All colonies from both the mutant libraries show fluorescence levels close to the negative control auto-fluorescent (Figure 3.13 B). The colonies in the libraries and empty vector in Figure 3.13 panel B appear to show similar fluorescence in comparison to panel A. This is because the iBright 1500 used to image the plates automatically adjusts its sensitivity to the highest level, which in the case of panel B was autofluorescence. Thus, the panels in Figure 3.13 must be compared based on the corresponding negative controls. This experiment was repeated two additional times and was also separately repeated using electrocompetent cells, yielding similar results. The cells from the library were also analysed by flow cytometry but yielded negative results.

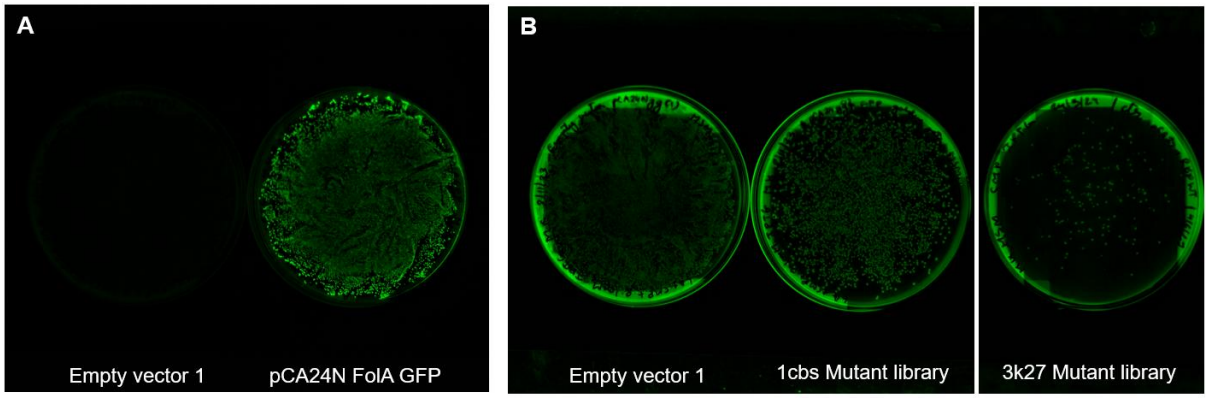


Figure 3.27: Expression of mutant libraries. Panel A show a positive control for fluorescence with FolA GFP construct. Panel B shows the two mutant libraries in comparison to the negative control.

Cloning top ranking mutants

We have also generated clones for mutants of the top 3 ranks for 1cbs and 3k2y. The DNA sequences were purchased as gene fragments from GeneScript via the local intermediary CellBioasis. These sequences are shown in Table 3.11. The gene fragments were inserted into pCA24N vector. The transformed colonies were isolated and streaked.

None of the clones showed a higher level of fluorescence that the negative control (Figure 3.14).

Table 3.11: Sequence for gene order for top 3 ranking combinations from Tables 3.5 and 3.6.

Name	length	sequence
RBP_rank1	447	GGCTACGGTCTCCTATGCCCAACTTCTCTGGCAACTGGAAAATCATCCG AAAAGAAAACTTCGAGGAATTGCTCAAAGTGCTGGGGGTGAATGTGA TGCTGAGGAAGATTGCTGTGGCTACCTATGGCAAGCCAGCAGTGGAG ATCAAACAGGAGGGAGACACTTTCTACATCAAAACCTCCACCACCGTG CGCACCACAGAGATTAACTTCAAGGTTGGGGAGGAGTTTGAGGAGCA GACTGTGGATGGGAGGCCCTGTAAGAGCCTGGTGAAATGGGAGAGTG AGAATAAAATGGTCTGTGAGCAGAAGCTCCTGAAGGGAGAGGGCCCC AAGACCTCGTGGACCAGAGAACTGACCAACGATGGGGAACTGATCCT GACCATGACGGCGGATGACGTTGTGTGCACCAGGGTTTCCGTCCGAAA GTAGCATGGAGACCGTAGCC

RBP_rank2	447	GGCTACGGTCTCCTATGCCCAACTTCTCTGGCAACTGGAAAATCATCCG AAAAGAAAACTTCGAGGAATTGCTCAAAGTGCTGGGGGTGAATGTGA TGCTGAGGAAGATTGCTGTGGCTACCTATGGCAAGCCAGCAGTGGAG ATCAAACAGGAGGGAGACACTTTCTACATCAAAACCTCCGAAACCGTG CGCACCACAGAGATTAACTTCAAGGTTGGGGAGGAGTTTGAGGAGCA GACTGTGGATGGGAGGCCCTGTAAGAGCCTGGTGAAATGGGAGAGTG AGAATAAAATGGTCTGTGAGCAGAAGCTCCTGAAGGGAGAGGGCCCC AAGACCTCGTGGACCAGAGAACTGACCAACGATGGGGAACTGATCCT GACCATGACGGCGGATGACGTTGTGTGCACCAGGGTTTCCGTCCGAAA GTAGCATGGAGACCGTAGCC
RBP_rank3	447	GGCTACGGTCTCCTATGCCCAACTTCTCTGGCAACTGGAAAATCATCCG AAAAGAAAACTTCGAGGAATTGCTCAAAGTGCTGGGGGTGAATGTGA TGCTGAGGAAGATTGCTGTGGCTACCTATGGCCGTCCAGCAGTGGAGA TCAAACAGGAGGGAGACACTTTCTACATCAAAACCTCCACCACCGTGC GCACCACAGAGATTAACTTCAAGGTTGGGGAGGAGTTTGAGGAGCAG ACTGTGGATGGGAGGCCCTGTAAGAGCCTGGTGAAATGGGAGAGTGA GAATAAAATGGTCTGTGAGCAGAAGCTCCTGAAGGGAGAGGGCCCCA AGACCTCGTGGACCAGAGAACTGACCAACGATGGGGAACTGATCCTG ACCATGACGGCGGATGACGTTGTGTGCACCAGGGTTTCCGTCCGAAAG TAGCATGGAGACCGTAGCC
3K2Y_rank1	339	GGCTACGGTCTCCTATGACAGGCGATGCAGCCGTCGCATTGGATACAG TCACGGTCGTTGGCGAACGGTATGTCGATGATATTGTGGCAACCTATG GCACGCTCAGAGTGGGTATGGCGGTGTTGCTCCAGCGAGAAAGTGGC AATCAATACGATGATAATGCCATCTCAGTGTGGACGTTGCAACATCATA AGTTAGGCTACATTGCCCGGTATCAGAACCAGCCGTACGCGACACTGA TGGATCAGGGGCAACGATTGTATGGCATCGTGACGGTACGTGATCAG CAAAAACAGCATCTTGAATTGATGCTATGGCGGTAGCATGGAGACCGT AGCC
3K2Y_rank2	339	GGCTACGGTCTCCTATGACAGGCGATGCAGCCGTCGCATTGGATACAG TCACGGTCGTTGGCGAACGGTATGTCGATGATATTGTGGCAACCTATG GCACGCTCAGAGTGGGTATGGCGGTGTTGCTCCAGCGAGAAAGTGGC

		AATCAATACGATGATAATGCCATCTCAGTGTGGACGTTGCAACATCATA AGTTAGGCTACATTGCCCGGTATCAGAACCAGCCGTACGCGACACTGA TGGATCAGGGGCAACGATTGTATGGCATCGTGACGGTACGTGATTGCC AAAAACAGCATCTTGAATTGATGCTATGGCGGTAGCATGGAGACCGTA GCC
3K2Y_rank3	339	GGCTACGGTCTCCTATGACAGGCGATGCAGCCGTCGCATTGGATACAG TCACGGTCGTTGGCGAACGGTATGTCGATGATATTGTGGCAACCTATG GCACGCTCAGAGTGGGTATGGCGGTGTTGCTCCAGCGAGAAAGTGGC AATCAATACGATGATAATGCCATCTCAGTGTGGACGTTGCAACATTGG AAGTTAGGCTACATTGCCCGGTATCAGAACCAGCCGTACGCGACACTG ATGGATCAGGGGCAACGATTGTATGGCATCGTGACGGTACGTGATCA GCAAAAACAGCATCTTGAATTGATGCTATGGCGGTAGCATGGAGACCG TAGCC

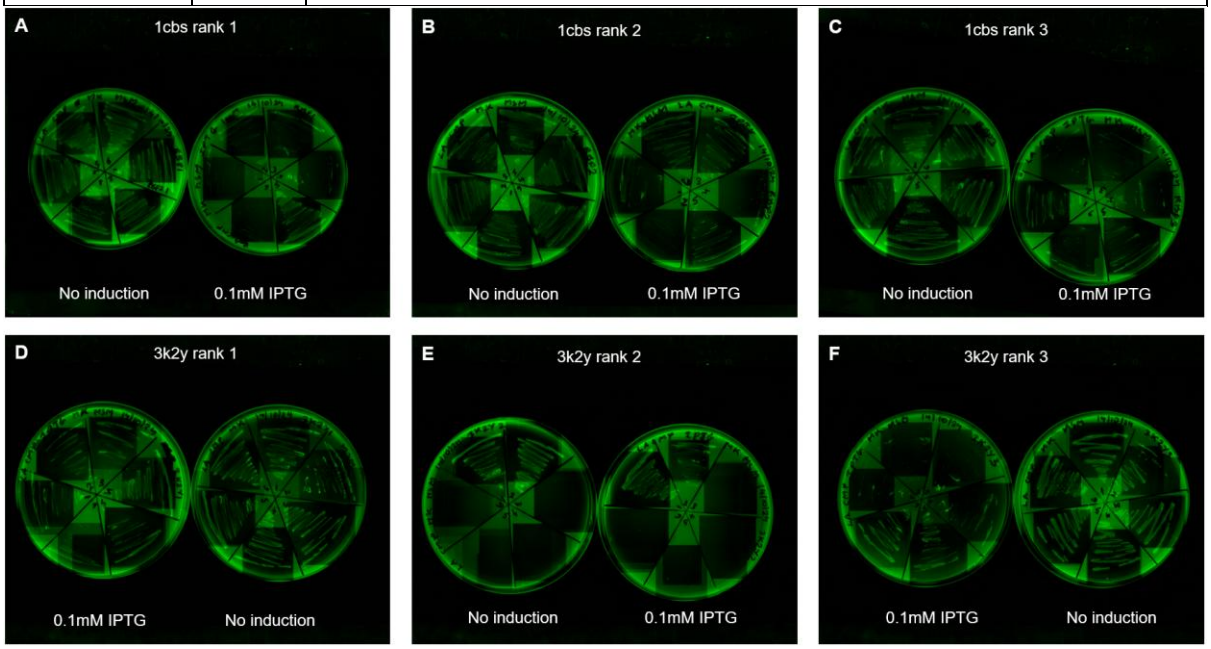


Figure 3.28: Panels show green fluorescence signal in streaks of bacteria expressing the protein encoded by the mutant sequences mentioned in the rows of Table 3.11. Each Panel also contains a negative control where there is no induction of the promoter.

Discussion

Proteins are the tools that biological systems evolved to execute various functions. Our aim in this project is to repurpose these tools for various industrial, medical, and scientific applications. To this end, we have described a method to transfer the function of one protein into another. We aim to achieve this solely by transferring the chemical

and structural features responsible for the function. We have predicted mutations in 12513 proteins that can make those proteins fluorescent. We have hand-selected 26 proteins from this set and reduced it to 3 proteins, through MD simulations and clustering based on similarity. We experimentally tested two of the three proteins and found them to be nonfluorescent, suggesting our method's failure. We are in the process of testing 1lm7, the third protein.

Reference

Abraham, Mark James, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindah. 2015. “Gromacs: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers.” *SoftwareX* 1–2:19–25. doi: 10.1016/j.softx.2015.06.001.

Anfinsen, Christian B. 1973. “Principles That Govern the Folding of Protein Chains.” *Science* 181(4096).

Barondeau, David P., Christopher D. Putnam, Carey J. Kassmann, John A. Tainer, and Elizabeth D. Getzoff. 2003. “Mechanism and Energetics of Green Fluorescent Protein Chromophore Synthesis Revealed by Trapped Intermediate Structures.” *Proceedings of the National Academy of Sciences* 100(21):12111–16. doi: 10.1073/pnas.2133463100.

Berman, H. M., John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. “The Protein Data Bank.” *Nucleic Acids Research* 28(1):235–42. doi: 10.1093/nar/28.1.235.

Boas, F. Edward, and Pehr B. Harbury. 2007. “Potential Energy Functions for Protein Design.” *Current Opinion in Structural Biology* 17(2):199–204. doi: https://doi.org/10.1016/j.sbi.2007.03.006.

Bowers, Peter M., Charlie E. M. Strauss, and David Baker. 2000. “De Novo Protein Structure Determination Using Sparse NMR Data.” *Journal of Biomolecular NMR* 18(4):311–18. doi: 10.1023/A:1026744431105.

Brooks, B. R., C. L. Brooks III, A. D. Mackerell Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. 2009. “CHARMM: The Biomolecular Simulation Program.” *Journal of Computational Chemistry* 30(10):1545–1614. doi: https://doi.org/10.1002/jcc.21287.

Dauparas, J., I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. 2022. “Robust Deep Learning-Based Protein Sequence Design Using ProteinMPNN.” *Science (New York, N.Y.)* 378(6615):49–56. doi: 10.1126/SCIENCE.ADD2187.

Desmet, Johan, Marc De Maeyer, Bart Hazes, and Ignace Lasters. 1992. “The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning.” *Nature* 356(6369):539–42. doi: 10.1038/356539a0.

Dou, Jiayi, Anastassia A. Vorobieva, William Sheffler, Lindsey A. Doyle, Hahnbeom Park, Matthew J. Bick, Binchen Mao, Glenna W. Foight, Min Yen Lee, Lauren A. Gagnon, Lauren Carter, Banumathi Sankaran, Sergey Ovchinnikov, Enrique Marcos, Po-Ssu Huang, Joshua C. Vaughan, Barry L. Stoddard, and David Baker. 2018. “De Novo Design of a Fluorescence-Activating β-Barrel.” *Nature* 561(7724):485–91. doi: 10.1038/s41586-018-0509-0.

Dunbrack, Roland L. 2002. “Rotamer Libraries in the 21st Century.” *Current Opinion in Structural Biology* 12(4):431–40. doi: https://doi.org/10.1016/S0959-440X(02)00344-5.

Engler, Carola, Romy Kandzia, and Sylvestre Marillonnet. 2008. “A One Pot, One Step, Precision Cloning Method with High Throughput Capability.” *PLOS ONE* 3(11):e3647.

Hallen, Mark A., Daniel A. Keedy, and Bruce R. Donald. 2013. “Dead-End Elimination with Perturbations (DEEPer): A Provable Protein Design Algorithm with Continuous Sidechain and Backbone Flexibility.” *Proteins* 81(1):18–39. doi: 10.1002/prot.24150.

Heim, R., D. C. Prasher, and R. Y. Tsien. 1994. “Wavelength Mutations and Posttranslational Autoxidation of Green Fluorescent Protein.” *Proceedings of the National Academy of Sciences* 91(26):12501 LP – 12504. doi: 10.1073/pnas.91.26.12501.

Huang, Jing, Sarah Rauscher, Grzegorz Nawrocki, Ting Ran, Michael Feig, Bert L. de Groot, Helmut Grubmüller, and Alexander D. Jr MacKerell. 2017. “CHARMM36m: An Improved Force Field for Folded and Intrinsically Disordered Proteins.” *Nature Methods* 14(1):71–73. doi: 10.1038/nmeth.4067.

Jacobs, T. M., B. Williams, T. Williams, X. Xu, A. Eletsky, J. F. Federizon, T. Szyperski, and B. Kuhlman. 2016. “Design of Structurally Distinct Proteins Using Strategies Inspired by Evolution.” *Science* 352(6286):687–90. doi: 10.1126/science.aad8036.

Jo, Sunhwan, Taehoon Kim, Vidyashankara G. Iyer, and Wonpil Im. 2008. “CHARMM-GUI: A Web-Based Graphical User Interface for CHARMM.” *Journal of Computational Chemistry* 29(11):1859–65. doi: https://doi.org/10.1002/jcc.20945.

Kearsley, S. K. 1989. “On the Orthogonal Transformation Used for Structural Comparisons.” *Acta Crystallographica Section A* 45(2):208–10. doi: 10.1107/S0108767388010128.

Kuhlman, Brian, and David Baker. 2000. “Native Protein Sequences Are Close to Optimal for Their Structures.” *Proceedings of the National Academy of Sciences* 97(19):10383 LP – 10388. doi: 10.1073/pnas.97.19.10383.

Kuhlman, Brian, and Philip Bradley. 2019. “Advances in Protein Structure Prediction and Design.” *Nature Reviews Molecular Cell Biology* 20(11):681–97. doi: 10.1038/s41580-019-0163-x.

Kuhlman, Brian, Gautam Dantas, Gregory C. Ireton, Gabriele Varani, Barry L. Stoddard, and David Baker. 2003. “Design of a Novel Globular Protein Fold with Atomic-Level Accuracy.” *Science* 302(5649):1364 LP – 1368. doi: 10.1126/science.1089427.

Lapidoth, Gideon D., Dror Baran, Gabriele M. Pszolla, Christoffer Norn, Assaf Alon, Michael D. Tyka, and Sarel J. Fleishman. 2015. “AbDesign: An Algorithm for Combinatorial Backbone Design Guided by Natural Conformations and Sequences.” *Proteins: Structure, Function and Bioinformatics* 83(8):1385–1406. doi: 10.1002/prot.24779.

Mackenzie, Craig O., Jianfu Zhou, and Gevorg Grigoryan. 2016. “Tertiary Alphabet for the Observable Protein Structural Universe.” *Proceedings of the National Academy of Sciences* 113(47):E7438–47. doi: 10.1073/pnas.1607178113.

Madhusudhan, M. S., B. M. Webb, M. A. Marti-Renom, N. Eswar, and A. Sali. 2009. “Alignment of Multiple Protein Structures Based on Sequence and Structure Features.” *Protein Engineering Design and Selection* 22(9):569–74. doi: 10.1093/protein/gzp040.

Mannige, Ranjan V. 2014. “Dynamic New World: Refining Our View of Protein Structure, Function and Evolution.” *Proteomes* 2(1):128–53. doi: 10.3390/proteomes2010128.

Niwa, Haruki, Satoshi Inouye, Takashi Hirano, Tatsuki Matsuno, Satoshi Kojima, Masayuki Kubota, Mamoru Ohashi, and Frederick I. Tsuji. 1996. “Chemical Nature of the Light Emitter of the Aequorea Green Fluorescent Protein.” *Proceedings of the National Academy of Sciences* 93(24):13617 LP – 13622. doi: 10.1073/pnas.93.24.13617.

Ormö, Mats, Andrew B. Cubitt, Karen Kallio, Larry A. Gross, Roger Y. Tsien, and S. James Remington. 1996. “Crystal Structure of the Aequorea Victoria Green Fluorescent Protein.” *Science* 273(5280):1392. doi: 10.1126/science.273.5280.1392.

Polizzi, Nicholas F., and William F. DeGrado. 2020. “A Defined Structural Unit Enables de Novo Design of Small-Molecule–Binding Proteins.” *Science* 369(6508):1227 LP – 1233. doi: 10.1126/science.abb8330.

Ponder, Jay W., and Frederic M. Richards. 1987. “Tertiary Templates for Proteins: Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes.” *Journal of Molecular Biology* 193(4):775–91. doi: https://doi.org/10.1016/0022-2836(87)90358-5.

Potapov, Vladimir, Jennifer L. Ong, Rebecca B. Kucera, Bradley W. Langhorst, Katharina Bilotti, John M. Pryor, Eric J. Cantor, Barry Canton, Thomas F. Knight, Thomas C. Jr. Evans, and Gregory J. S. Lohman. 2018. “Comprehensive Profiling of Four Base Overhang Ligation Fidelity by T4 DNA Ligase and Application to DNA Assembly.” *ACS Synthetic Biology* 7(11):2665–74. doi: 10.1021/acssynbio.8b00333.

Potapov, Vladimir, Jennifer L. Ong, Bradley W. Langhorst, Katharina Bilotti, Dan Cahoon, Barry Canton, Thomas F. Knight, Thomas C. Evans Jr, and Gregory J. S. Lohman. 2018. “A Single-Molecule Sequencing Assay for the Comprehensive

Profiling of T4 DNA Ligase Fidelity and Bias during DNA End-Joining." *Nucleic Acids Research* 46(13):e79–e79. doi: 10.1093/nar/gky303.

Pronk, Sander, Szilárd Páll, Roland Schulz, Per Larsson, Pär Bjelkmar, Rossen Apostolov, Michael R. Shirts, Jeremy C. Smith, Peter M. Kasson, David Van Der Spoel, Berk Hess, and Erik Lindahl. 2013. "GROMACS 4.5: A High-Throughput and Highly Parallel Open Source Molecular Simulation Toolkit." *Bioinformatics*. doi: 10.1093/bioinformatics/btt055.

Pryor, John M., Vladimir Potapov, Katharina Bilotti, Nilisha Pokhrel, and Gregory J. S. Lohman. 2022. "Rapid 40 Kb Genome Construction from 52 Parts through Data-Optimized Assembly Design." *ACS Synthetic Biology* 11(6):2036–42. doi: 10.1021/acssynbio.1c00525.

Pryor, John M., Vladimir Potapov, Rebecca B. Kucera, Katharina Bilotti, Eric J. Cantor, and Gregory J. S. Lohman. 2020. "Enabling One-Pot Golden Gate Assemblies of Unprecedented Complexity Using Data-Optimized Assembly Design." *PLOS ONE* 15(9):e0238592.

Romei, Matthew G., and Steven G. Boxer. 2019. "Split Green Fluorescent Proteins: Scope, Limitations, and Outlook." *Annual Review of Biophysics* 48(1):19–44. doi: 10.1146/annurev-biophys-051013-022846.

Šali, Andrej, and Tom L. Blundell. 1993. "Comparative Protein Modelling by Satisfaction of Spatial Restraints." *Journal of Molecular Biology* 234(3):779–815. doi: 10.1006/jmbi.1993.1626.

Shapovalov, Maxim V., and Roland L. Dunbrack. 2011. "A Smoothed Backbone-Dependent Rotamer Library for Proteins Derived from Adaptive Kernel Density Estimates and Regressions." *Structure* 19(6):844–58. doi: 10.1016/j.str.2011.03.019.

Shimomura, Osamu, Frank H. Johnson, and Yo Saiga. 1962. "Extraction, Purification and Properties of Aequorin, a Bioluminescent Protein from the Luminous Hydromedusan, Aequorea." *Journal of Cellular and Comparative Physiology* 59(3):223–39. doi: https://doi.org/10.1002/jcp.1030590302.

Ward, William W., and Stephen H. Bokman. 1982. "Reversible Denaturation of Aequorea Green-Fluorescent Protein: Physical Separation and Characterization of the Renatured Protein." *Biochemistry* 21(19):4535–40. doi: 10.1021/bi00262a003.

Watson, Joseph L., David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. 2023. "De Novo Design of Protein Structure and Function with RFdiffusion." *Nature* 620(7976):1089–1100. doi: 10.1038/S41586-023-06415-8.

Webb, Benjamin, and Andrej Sali. 2016. "Comparative Protein Structure Modeling Using MODELLER." *Current Protocols in Bioinformatics* 2016:5.6.1-5.6.37. doi:

10.1002/cpbi.3.

Xu, Chunfu, Peilong Lu, Tamer M. Gamal El-Din, Xue Y. Pei, Matthew C. Johnson, Atsuko Uyeda, Matthew J. Bick, Qi Xu, Daohua Jiang, Hua Bai, Gabriella Reggiano, Yang Hsia, T. J. Brunette, Jiayi Dou, Dan Ma, Eric M. Lynch, Scott E. Boyken, Po-Ssu Huang, Lance Stewart, Frank DiMaio, Justin M. Kollman, Ben F. Luisi, Tomoaki Matsuura, William A. Catterall, and David Baker. 2020. “Computational Design of Transmembrane Pores.” *Nature* 585(7823):129–34. doi: 10.1038/s41586-020-2646-5.

Yeh, Andy Hsien-Wei, Christoffer Norn, Yakov Kipnis, Doug Tischer, Samuel J. Pellock, Declan Evans, Pengchen Ma, Gyu Rie Lee, Jason Z. Zhang, Ivan Anishchenko, Brian Coventry, Longxing Cao, Justas Dauparas, Samer Halabiya, Michelle DeWitt, Lauren Carter, K. N. Houk, and David Baker. 2023. “De Novo Design of Luciferases Using Deep Learning.” *Nature* 614(7949):774–80. doi: 10.1038/s41586-023-05696-3.

Zhou, Jianfu, Alexandra E. Panaitiu, and Gevorg Grigoryan. 2020. “A General-Purpose Protein Design Framework Based on Mining Sequence–Structure Relationships in Known Protein Structures.” *Proceedings of the National Academy of Sciences* 117(2):1059–68. doi: 10.1073/pnas.1908723117.

Chapter 4: Overhang Optimizer for Golden Gate Assembly (OOGGA)

Introduction

Golden gate cloning is a molecular cloning method used to assemble multiple fragments of DNA in a specific orientation (Engler, Kandzia, and Marillonnet 2008). This is achieved using type IIs restriction enzymes such as BsaI and BsmBI. These restriction enzymes recognize specific 6 nucleotide sequence motifs, and cut the DNA at 1/5 sites downstream, creating a 4 base pair overhang at non-specific sites. These sites can be chosen such that DNA fragments share complementary overhangs that lead to assemblies facilitated by Watson-Crick base pairing. The nicks in these assemblies are ligated by DNA ligases (for example T4 and T7) present in the reaction to create progressively larger segments of DNA. This method is capable of accurately assembling more than 50 fragments of DNA (Pryor et al. 2020, 2022).

The accuracy of this method depends on sequence complementarity conferred by Watson-Crick base pairing. But there are exceptions to this rule that lead to mismatches, contributing to errors in assembly. Moreover, DNA ligases have sequence preferences that determine the efficiency and accuracy of ligation (Bilotti et al. 2022). The ligation efficiency for all 256 X 256 pairwise combinations of four base pair overhangs has been determined by single molecule sequencing (Potapov, Jennifer L Ong, et al. 2018; Potapov, Jennifer L. Ong, et al. 2018). Selecting high-ranking overhangs from these data (in terms of efficiency and fidelity) is necessary for robust experimental designs. Manual selection of these overhangs can get cumbersome and error-prone.

The tool, NEBridge SplitSet® (SplitSet) uses data from Potatov et. al. to predict fragments of a DNA sequence with high-fidelity overhangs based on the number of required fragments. SplitSet Monte Carlo search approach which identifies a high-fidelity set of overhangs from a large number of possible combinations (10^{14} for 10 overhangs) (Pryor et al. 2020). However, Monte Carlo is not deterministic and does not guarantee the best set of overhangs and the results from Monte Carlo simulations also differ from one run to another. Moreover, the method, while optimizing for fidelity, ignores the efficiency of ligation which is critical for applications like mutational screening, where the resulting larger populations are helpful and sometimes necessary.

Here we report a program called Overhang Optimizer for Golden Gate Assembly (OOGGA). This program uses dynamic programming to find the fragments of a given DNA sequence with the best set of overhangs, given the range of fragment length (the required number of fragments can also be provided). The efficiency and fidelity matrices, derived from the single molecule sequencing data generated by Potapov et al, are optimized in the process. We compare 5 fragments predicted for DNA sequences by both OOGGA and SplitSet. Compared to SplitSet, OOGGA predicts fragments with higher fidelity overhangs. Moreover, OOGGA can also optimize the efficiency of the overhangs.

Methods

Obtaining scores of efficiencies and fidelity for overhangs

Metrics that quantify the efficiency and the fidelity of an overhang were calculated from the single molecule sequencing data generated by Potapov et al. The probability of an overhang undergoing ligation with its Watson-Crick counterpart (called match, whereas mismatch is anything other than Watson-Crick ligations) is considered as the metric for evaluating the fidelity (equation 1).

$$P_{fid}(overhang) = \frac{n_{match}^{overhang}}{n_{match}^{overhang} + n_{mismatch}^{overhang}} \qquad (1)$$

Here n is the number of reads obtained by Potapov et al. The value of the number of matching reads of an overhang divided by the value of the largest matching read among all the overhangs is considered as the metric of efficiency (equation 2). Note that this value ranges between 0-1 (equation 3).

$$P_{eff}(overhang) = \frac{n_{match}^{overhang}}{\max_{i \in all\ overhangs} n_{match}^{i})} \qquad (2)$$

$$P_{eff}(overhang) \in [0,1] \qquad (3)$$

Overhang Optimizer for Golden Gate Assembly (OOGGA)

OOGGA employs dynamic programming to optimize overhang positions and the number of fragments for the golden gate assembly of a DNA sequence. The input to the program is the DNA sequence of interest and the size range of the fragments. The

dynamic programming algorithm finds the combination of fragments that would maximise the probability of efficient and accurate assembly.

This is achieved by aligning overhang positions to the fragments while optimizing the net efficiency and fidelity of the assembly.

First, the number of fragments (K) is determined. If the number is not provided by the user, equation 4 is used to calculate the maximum possible number of fragments.

$$K = \frac{N}{len_{min}} \qquad (4)$$

Where N is the number of base pairs in the DNA sequence and K is rounded up to the higher integer. len_{min} is the minimum length of the fragment (similarly len_{max} is the maximum length of the fragment). After this, four matrices with K columns and N rows are created, indexed by i and j respectively. These are S, P_{eff}, P_{fid} and T which store the total score, the efficiency score, the fidelity score, and the trace respectively. The following recursive algorithm is used to populate these matrices.

Initiation

$$\boldsymbol{S}(0, j_{start}) = 1$$

$$\boldsymbol{P}_{eff}(0, j_{start}) = 1$$

$$\boldsymbol{P}_{fid}(0, j_{start}) = 1$$

$$\boldsymbol{T}(0, j_{start}) = 0$$

j_{start} is the position of the first fragment. This is by default set as 0. Assigning probabilities to multiple start sites can also optimise starting site fragments. Here *S*, P_{eff}, P_{fid} and *T* are matrices with K columns and N rows. *S* stores the total score (see 'Iteration' section to see how score is computed) corresponding to the fragmentation at a given DNA position. P_{eff} and P_{fid} store the efficiency and fidelity that is used to compute the score. *T* stores the previous fragmentation site based on which the current fragmentation site is computed.

Iteration

Pseudocode for computing *S*, P_{eff}, P_{fid} and *T*

$for\ i\ <=\ K$

$for\ j\ \in [len_{min} \times i, len_{max} \times i]$

$j_{max}^{i-1} = argmax\left(f\left(j^{i-1}\right), j^{i-1}\ \in [len_{min} \times (i-1), len_{max} \times (i-1)]\ AND\ len_{min} \leq\ j - j^{i-1}\ \leq len_{max}\right)$

Where,

$f\left(j^{i-1}\right) = \left(\boldsymbol{P}_{eff}\left(i-1, j^{i-1}\right) \times P_{eff}\left(j^{i}\right)\right)^{\boldsymbol{w_{eff}}} \times \left(\boldsymbol{P}_{fid}\left(i-1, j^{i-1}\right) \times P_{fid}\left(j^{i}\right)\right)^{\boldsymbol{w_{fid}}}$

Using j_{max}^{i-1} update the matrices

$\boldsymbol{S}(i,j) = f\left(j_{max}^{i-1}\right)$

$\boldsymbol{P}_{eff}(i,j) = \boldsymbol{P}_{eff}\left(i-1, j_{max}^{i-1}\right) \times P_{eff}\left(j^{i}\right)$

$\boldsymbol{P}_{fid}(i,j) = \boldsymbol{P}_{fid}\left(i-1, j_{max}^{i-1}\right) \times P_{fid}\left(j^{i}\right)$

$\boldsymbol{T}(i,j) = j_{max}^{i-1}$

Termination

When *i* > *K*

Here, w_{eff} and w_{fid} are weights for efficiency and fidelity respectively.

Traceback

The top n_trace of the score of any overhang that can be part of the final fragment is selected as the starting point of the traceback.

$$\left(i_{opt}, j_{opt}\right) = argmax(\boldsymbol{S}(i,j), j \in [N - len_{max}, N], i \in [i, K])$$

If the number of required fragments (*K*) is specified by the user, the following equation is used.

$$\left(i_{opt}, j_{opt}\right) = argmax(\boldsymbol{S}(K,j), j \in [N - len_{max}, N]])$$

Time complexity

The number of total calculations is the following.

$$n_{calc} = N \times K \times \left(1 + len_{frag} + 2 \times len_{frag} \ldots (K-1) \times len_{frag}\right)$$

$$n_{calc} = N \times K \times \left(1 + \frac{K}{2} \times len_{frag}\right)$$

Where len_{frag} is the maximum length of the fragment.

Thus, time complexity = $O\left(N \times K \times \left(1 + \frac{K}{2} \times len_{frag}\right)\right)$

Predicted efficiency and fidelity values

For a set of overhangs called *Overhangs*, the net efficiency is computed as

$$P_{eff}^{tot} = \prod_{i \in Overhangs} P_{eff}(i)$$

Predicted efficiency in percentage is computed by multiplying P_{eff}^{tot} with 100. This represents the percentage efficiency w. r. t. the best overhang identified by Potapov et al.

Similarly, the net fidelity (probability of assembly without mismatch) is computed as

$$P_{fid}^{tot} = \prod_{i \in Overhangs} P_{fid}(i)$$

Predicted fidelity in percentage is computed by multiplying P_{fid}^{tot} with 100. This represents the percentage of the population of the correct molecule from the set of all assembled molecules.

Results

Example

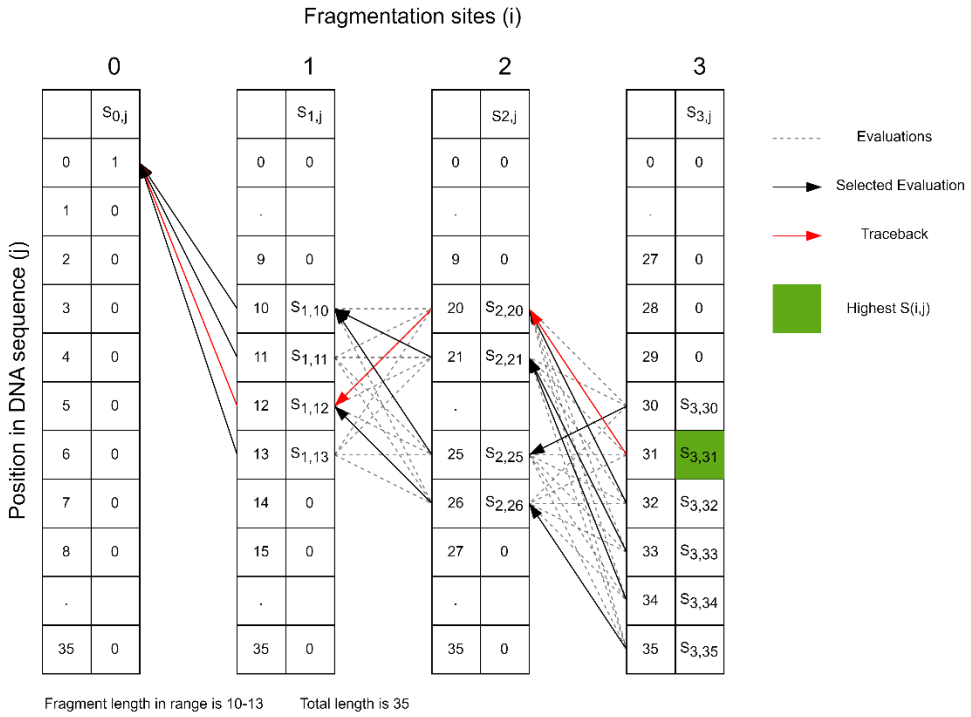


Figure 4.29: Illustration of OOGGA algorithm processing an example DNA sequence of length 35 for fragments of length range 10-13.

Consider a DNA sequence of length 35. The aim is to make K fragments between the range 10-13 and thus, in this case, $K = 4$. The fragments are indexed with i and DNA positions that form overhangs are indexed with j. The calculation starts at $i = 0$. Here $S(0,0)$ is set as 1 to denote the starting site. Note that multiple start sites can be set by adding probabilities to different j positions.

At $i = 1$, only j in the range 10-13 are evaluated ($f(j^{i-1})$ from section 2.2.2) (dotted grey lines in figure 4.1) as the rest of the positions do not qualify the fragment length criterion. The j^{i-1} which gives the highest value is picked as the trace (solid black and red arrows in Figure 4.1). For $i = 2$, the range of j evaluated is 20-26 (size range multiplied by i) as this range describes the shortest and largest fragments possible in relation to i-1. Similarly, the evaluation proceeds as per section 2.2.2 until $i > K$ (section 2.2.3).

The traceback starts by evaluating all i,j positions that fall within the allowed fragment range. Assume that $S(3,31)$ has the highest score in the example. The traceback follows the traces set during the evaluation stage starting from (3,31) until $i = 0$ (in this example (0,0) (solid red arrows in figure 4.1). The positions of overhangs identified are 10, 20 and 31 creating 4 fragments. Please note that the highest $S(i,j)$ may not have

the highest i. Thus, the algorithm can predict the optimal combination of overhangs and the number of fragments given only the DNA sequence and the size range of fragments.

Comparison of fidelity of overhangs predicted by OOGGA and SplitSet

We compared the fidelity of overhangs from the fragments predicted by OOGGA and SplitSet. To do this we used three gene sequences 1) retinol binding protein 2) protein lp_0118 from *Lactobacillus plantarum* and 3) desmoplakin. The DNA sequences of these genes can be found in the attached supplementary file sequences.fasta. SplitSet results were obtained from the web server after setting the required number of fragments as 5 (all other parameters were kept at their default set values). OOGGA results were obtained with w_{eff} set as 0 to optimize only the fidelity, and thus have a direct comparison with SplitSet. The number of fragments was set at 5 and the maximum and minimum fragment length was set as 20-150 for sequence 1, 20-100 for sequence 2 and 2000-3000 for sequence 3. The efficiency and fidelity values were calculated as shown in the methods section 'Predicted efficiency and fidelity values'.

OOGGA predicts fragments with overhangs with higher or matching fidelities compared to SplitSet (Table 4.1) in all scenarios. This demonstrates the advantage of dynamic programming optimization by OOGGA, which guarantees the best solution over the stochastic search of SplitSet. However, since neither methods optimize for the efficiency of ligation of the overhang, the efficiencies of the predicted overhangs predicted by both methods are poor (Table 4.2).

Table 4.1: The predicted fidelity of overhangs predicted by SplitSet compared to OOGGA for the 3 DNA sequences. Fidelity is represented as % of correct assemblies. $w_{eff} = 0$ *and* $w_{fid} = 1$ for OOGGA. SplitSet values were obtained from its web server. Bold values show higher/matching values.

	% correct assemblies (w_{eff} = 0)					
	Sequence 1		**Sequence 2**		**Sequence 3**	
	SplitSet	**OOGGA**	**SplitSet**	**OOGGA**	**SplitSet**	**OOGGA**
Overhang 1	96	**100**	98	**99**	91	**100**
Overhang 2	**99**	**99**	97	**99**	92	**99**
Overhang 3	98	**99**	90	**100**	98	**100**
Overhang 4	**99**	**99**	96	**99**	**99**	**99**
Net fidelity	92	**98**	82	**98**	82	**98**

Table 4.2: The predicted efficiency of overhangs predicted by SplitSet compared to OOGGA for three DNA sequences. Efficiency is represented in % w. r. t. the most efficient overhang. $w_{eff} = 0$ *and* $w_{fid} =$

1 for OOGGA. SplitSet values were obtained from its web server. Bold values show higher/matching values.

% Efficiency w.r.t. most efficient overhang (w_{eff} = 0)						
	Sequence 1		Sequence 2		Sequence 3	
	SplitSet	OOGGA	SplitSet	OOGGA	SplitSet	OOGGA
Overhang 1	**77**	70	31	**82**	**81**	70
Overhang 2	**89**	65	**81**	65	**81**	65
Overhang 3	**85**	66	57	**70**	**85**	30
Overhang 4	38	**62**	**77**	65	**82**	61
Net Efficiency	**22**	18	11	**24**	**46**	8

To check if OOGGA can predict overhangs that have both high efficiency and fidelity we set w_{eff} as 1.

OOGGA predicts overhangs with high fidelity and efficiency

We predicted overhangs with OOGGA for the three DNA sequences while setting w_{eff} as 1 and w_{fid} as 1 (equal preference to both efficiency and fidelity). With these weights, OOGGA Should optimize for overhangs with both high fidelity and efficiency. Other parameters used were the same as in section 3.2. SplitSet data used is the same as in section 3.2.

When w_{eff} was set as 1, OOGGA outperformed SplitSet in all efficiency-related scenarios (Table 4.3). However, this is expected as SplitSet does not optimize for efficiency. On evaluating the fidelity of the predicted overhangs, OOGGA predicts better overhangs for sequences 2 and 3 but falls behind SplitSet for sequence 1 (Table 4.4). However, it should be noted that the overhangs that OOGGA predicts are optimized for both efficiency and fidelity, compared to SplitSet which only optimizes for fidelity.

	% Efficiency w.r.t. most efficient overhang					
	Sequence 1		Sequence 2		Sequence 3	
	SplitSet	OOGGA	SplitSet	OOGGA	SplitSet	OOGGA
Overhang 1	77	**95**	31	**90**	81	**100**
Overhang 2	89	**100**	81	**99**	81	**95**
Overhang 3	85	**90**	57	**100**	85	**93**
Overhang 4	38	**99**	77	**94**	82	**92**
Net efficiency	22	**84**	11	**85**	46	**82**

Table 4.3: The predicted efficiency of overhangs predicted by SplitSet compared to OOGGA for the 3 DNA sequences. Efficiency is represented in % w. r. t. the most efficient overhang. $w_{eff} = w_{fid} = 1$ for OOGGA. SplitSet values were obtained from its web server. Bold values show higher/matching values.

	% correct assemblies					
	Sequence 1		Sequence 2		Sequence 3	
	SplitSet	OOGGA	SplitSet	OOGGA	SplitSet	OOGGA
Overhang 1	**96**	95	98	**99**	91	**95**
Overhang 2	**99**	95	**97**	93	92	**95**
Overhang 3	**98**	96	90	**95**	98	**99**
Overhang 4	**99**	93	**96**	95	**99**	98
Net fidelity	**92**	80	82	**83**	82	**88**

Table 4.4: The predicted fidelity of overhangs predicted by SplitSet compared to OOGGA for the 3 DNA sequences. Fidelity is represented as % of correct assemblies. $w_{eff} = w_{fid} = 1$ for OOGGA. Splitset values were obtained from its web server. Bold values show higher/matching values.

Thus, OOGGA predicts high-efficiency overhangs with fidelities comparable to SplitSet.

Discussion

We have created a dynamic programming algorithm that predicts fragments with optimal overhangs for Golden Gate assembly (OOGGA). We compared it against NEBridge SplitSet® which is the state-of-the-art method of predicting fragments for Golden gate assembly.

OOGGA outperforms or matches SplitSet in all scenarios on a direct comparison where both programs only optimize for fidelity. This is expected because OOGGA is a dynamic programming optimization algorithm while SplitSet is a stochastic search method, and the first guarantees the optimal solution. Moreover, OOGGA can predict overhangs with comparable accuracies to SplitSet even while optimizing for efficiency of ligation of overhangs (SplitSet does not optimize efficiency).

The difference in fidelity between OOGGA and SplitSet for each fragment is small enough that sequence upstream of the cleavage site can affect the outcome. However, on an average, the errors caused by the effect of the upstream site is equal on both OOGGA fragments and SplitSet fragments as they both used the same experimental data (Potapov, Jennifer L Ong, et al. 2018; Potapov, Jennifer L. Ong, et al. 2018) to predict fidelity. Since OOGGA is a dynamic programing predictor, it mathematically

assures the best outcome (global minima) based on the provided data, which is not true for SplitSet. Thus, predictions from OOGGA will theoretically always (even if the difference is marginal) be better than SplitSet.

One might ask whether there is a need for such rigorous optimization. The following two paragraphs answer this question with answers distinct from academic curiosity.

Overhangs with high ligation efficiency would presumably reduce incubation times by an insignificant amount for day-to-day cloning. However, it is an important metric while designing overhangs for library preparations where higher efficiency implies a larger sample of the population. Thus, the overhangs generated by SplitSet are unsuitable for applications in large library preparations and OOGGA must be used for these applications as it predicts high efficiency and high-fidelity overhangs.

It is common to use fragments generated for golden gate assembly as part of modular systems. While optimizing for fidelity SplitSet only considers the mismatches between the predicted overhangs from the same sequence. Thus, these overhangs can have high error rates when combined with another system of fragments with a different set of overhangs. OOGGA avoids this by computing its fidelity score as the probability of an overhang of binding to only its Watson Crick match, mismatches with all overhangs are considered in doing this. Thus, the fidelity of the overhangs predicted by OOGGA is more universal compared to those predicted by SplitSet. Thus, overhangs of the fragments predicted by OOGGA are better in both the accuracy and efficiency of assembly.

These advantages of OOGGA can be experimentally verified by assembling two sets of fragments of a plasmid predicted by OOGGA and SplitSet separately, followed by counting the number of colonies after transformation (metric of efficiency) and selection by the corresponding antibiotic. Sequencing the plasmids isolated from the colonies can provide the rate of erroneous assembly (metric of fidelity). We are currently pursuing such an experiment.

OOGGA also supports the processing of DNA sequences with non-ATGC bases (eg. degeneracy codes), though the overhangs containing these bases are not considered for output. There can also be the exclusion of specific positions near these bases so that the overhangs are not predicted near them.

Note that the time complexity (section 2.3) and memory requirements of OOGGA, being a dynamic programming approach will be inferior compared to that of the SplitSet Monte Carlo method. However, designing fragments for golden gate assembly is not a time-sensitive event. The fidelity and efficiency of the fragments take priority over the increase in compute time and memory requirements.

Our results show that OOGGA is a better method compared to SplitSet to generate fragments for golden gate assembly. Our method predicts more accurate overhangs compared to SplitSet while also optimizing for efficiency and thus should be used for predicting overhangs for golden gate assembly instead of SplitSet.

Reference

Bilotti, Katharina, Vladimir Potapov, John M. Pryor, Alexander T. Duckworth, James L. Keck, and Gregory J. S. Lohman. 2022. “Mismatch Discrimination and Sequence Bias during End-Joining by DNA Ligases.” *Nucleic Acids Research* 50(8):4647–58. doi: 10.1093/nar/gkac241.

Engler, Carola, Romy Kandzia, and Sylvestre Marillonnet. 2008. “A One Pot, One Step, Precision Cloning Method with High Throughput Capability.” *PLOS ONE* 3(11):e3647.

Potapov, Vladimir, Jennifer L. Ong, Rebecca B. Kucera, Bradley W. Langhorst, Katharina Bilotti, John M. Pryor, Eric J. Cantor, Barry Canton, Thomas F. Knight, Thomas C. Evans, and Gregory J. S. Lohman. 2018. “Comprehensive Profiling of Four Base Overhang Ligation Fidelity by T4 DNA Ligase and Application to DNA Assembly.” *ACS Synthetic Biology* 7(11):2665–74. doi: 10.1021/acssynbio.8b00333.

Potapov, Vladimir, Jennifer L Ong, Bradley W. Langhorst, Katharina Bilotti, Dan Cahoon, Barry Canton, Thomas F. Knight, Thomas C. Evans Jr, and Gregory J. S. Lohman. 2018. “A Single-Molecule Sequencing Assay for the Comprehensive Profiling of T4 DNA Ligase Fidelity and Bias during DNA End-Joining.” *Nucleic Acids Research* 46(13):e79–e79. doi: 10.1093/nar/gky303.

Pryor, John M., Vladimir Potapov, Katharina Bilotti, Nilisha Pokhrel, and Gregory J. S. Lohman. 2022. “Rapid 40 Kb Genome Construction from 52 Parts through Data-Optimized Assembly Design.” *ACS Synthetic Biology* 11(6):2036–42. doi: 10.1021/acssynbio.1c00525.

Pryor, John M., Vladimir Potapov, Rebecca B. Kucera, Katharina Bilotti, Eric J. Cantor, and Gregory J. S. Lohman. 2020. “Enabling One-Pot Golden Gate Assemblies of Unprecedented Complexity Using Data-Optimized Assembly Design.” *PLOS ONE* 15(9):e0238592.

Chapter 5: High-affinity biomolecular interactions are modulated by low-affinity binders

Publication

Mukundan, S., Deshpande, G. & Madhusudhan, M.S. High-affinity biomolecular interactions are modulated by low-affinity binders. npj Syst Biol Appl 10, 85 (2024). https://doi.org/10.1038/s41540-024-00410-z

Abstract

The strength of molecular interactions is characterized by their dissociation constants (K_D). Only high-affinity interactions ($K_D \leq 10^{-8}$ M) are extensively investigated and support binary on/off switches. However, such analyses have discounted the presence of low-affinity binders ($K_D > 10^{-5}$ M) in the cellular environment. We assess the potential influence of low-affinity binders on high-affinity interactions. By employing Gillespie stochastic simulations and continuous methods, we demonstrate that the presence of low-affinity binders can alter the kinetics and the steady state of high-affinity interactions. We refer to this effect as 'herd regulation' and have evaluated its possible impact in two different contexts including sex determination in Drosophila melanogaster and in signalling systems that employ molecular thresholds. We have also suggested experiments to validate herd regulation in vitro. We speculate that low-affinity binders are prevalent in biological contexts where the outcomes depend on molecular thresholds impacting homoeostatic regulation.

1 Introduction

Key to all biological processes are biomolecular interactions. Networks of these interactions are organised into pathways, that in many instances, intersect (Huh et al., 2003; Kanehisa et al., 2022). These pathways usually consist of tight binding interactions. In a cellular context, these interactions do not occur in isolation but in the presence of a plethora of other molecules and pathways. This, in turn, ensures that biomolecules can promiscuously interact with multiple partners with varying degrees of affinity. It stands to reason that some of these interactions and their interaction strengths have evolved to be integral parts of signalling pathways.

In this study, we have analysed the influence of specific but low affinity binders on tight binding high affinity interactions. The strengths of these interactions are reported in terms of their dissociation constants, K_D. K_D is the equilibrium constant of the reverse reaction given by the ratio of the reverse and forward rate constants. Typically, members of a pathway would interact with their partners, with K_D values in the range of 10^{-6} M to 10^{-14} M (Hu, 2017; Perkins et al., 2010), with lower values corresponding to stronger binding. However, the cellular environment consists of a variety of interactors with variable affinities. In addition to its specific high affinity binding partner(s), a biomolecule could encounter a cohort of other binders with K_D values >

10^{-6} M present in the vicinity. Many of these lower affinity interactions could be energetically favourable (negative ΔG) and thus are competent to sequester the reactants from the high affinity reaction.

Studies of biological processes usually focus almost exclusively on the important tight binding interactions. This is in part because these canonical interactions are considered to be central to cellular processes. Moreover, experimental techniques mostly detect strong interactors (Galaway and Wright, 2020; Jiang and Barclay, 2009; Li et al., 2014). Here, we have examined how these high affinity (tight binding) interactions could be altered and/or modulated in the presence of several specific low affinity binders. For this study, low affinity interactors were conservatively considered to have K_D values of 2 to 5 orders of magnitude less favourable than their stronger counterparts. We also vary the concentration of the low affinity binders to be in the range of 1X to 10X to that of the tight binding partners. This is a conservative estimate as the cellular milieu may contain a higher concentration of the low affinity binding partners, mechanisms for local enrichment of binding partners notwithstanding.

Using these values of binding strengths and relative concentrations, we investigated the combined effect of such weak binding partners on the kinetics and equilibrium of a tight binding interaction. We computed these effects using Gillespie stochastic simulations (Gillespie, 1977, 2013) and also replicated them with a system of differential equations. For instance, consider the following interaction.

$$A + B \rightleftharpoons AB$$

Here, the reactants *A* and *B* interact tightly to form the product *AB*. Let A_i and B_i represent the cohort of specific low affinity binders of *B* and *A* respectively. We have studied how the kinetics and steady-state changes with varying levels and binding affinities of A_i and B_i in the cases where – i) there are low affinity binders for either *A* or *B*, ii) both reactants have low affinity binders, and iii) there are low affinity binders to *A* and *B* and the A_i's and B_i's can interact with each other. We have coined the term herd regulation to represent the model of low-affinity binders interfering with high affinity interactions. Intuitively, this would be possible when the low-affinity interactions are in sizeable quantities (herds).

In a cellular context, such interactions should be prevalent as the low affinity binders are likely widespread. Their presence could modulate molecular interactions and

hence biological activities. Here, we have modelled the process of sex determination in *Drosophila melanogaster* and signalling thresholds of biomolecular processes, two processes whose outcomes are dependent on thresholds.

Sex-lethal (*Sxl*), a master-switch gene, which determines the female identity, is turned on by the doubling of the X chromosome signal in females (Cline and Meyer, 1996). Analysis of mutations in X-chromosome-linked activators has confirmed that activation of early embryonic promoter of *Sxl* (*Sxl-Pe*) is critical in females and thus, underlies the binary fate choice. By contrast, strong repressors capable of affecting sex determination i.e. sex-specific identity and/or viability, have not been identified (Mahadeveraju et al., 2020; Salz and Erickson, 2010). Loss of function mutations in such repressors would result in anomalous activation of *Sxl-Pe* in males, which in turn, can induce male-to-female transformation or lethality. Here, we hypothesise that such repressors are a set of low affinity binders. A single mutation, typically isolated during genetic screens, however, would not elicit strong phenotypic consequences (e.g. sex-specific lethality) used as a conventional and stringent criteria for selection. To examine the possible impact of such repressors, we have modelled this system considering *A* as the switch gene, *B* as the X chromosome signal and A_i and B_i as repressors. We demonstrate that our simulation strategy yields outputs that are consistent with previously reported experimental findings.

Successful activation of biological signalling pathways often requires signal strength to attain or surpass a defined threshold value. Such systems involve the conversion of a linear change in the signal, to a binary on/off state. Morphogen gradients leading to cellular patterning in a variety of developmental contexts constitute one such example (Wartlick et al., 2009). Although such cases are prevalent in biological systems, precise molecular mechanisms underlying these decisions are not always fully understood. Previous modelling attempts at elucidating the mechanisms have employed models that depend on cellular compartments and/or promiscuous binders (Iyer et al., 2022). In this study, we provide a simple yet broadly applicable mechanism that explains such thresholds using low affinity binders.

2 Results

2.1 The effect of low affinity competitors on strong interactions.

2.1.1 Only one of the strong interactors has competition.

To start with, we examined the effect of low affinity interactors on one of the components of a tight binding complex, i.e., the effect of B_i on the formation of the complex *AB* (see equations E1 and E2). The outcome of these reactions is intuitive but serves the crucial purpose of creating a control case for subsequent sections. *AB* (dissociation constant K_D = 10^{-8} M) is 3 orders of magnitude stronger than AB_i (dissociation constant K_{D_i} = 10^{-5} M). We have monitored the effect of two parameters, initial levels of the low affinity interactors, B_{i0}, and their binding affinities, K_{D_i}. We analysed the kinetics and steady states of these reactions using the Gillespie stochastic simulation algorithm (Gillespie, 2013).

First, we studied the effect of B_{i0} levels. The simulation was set up with 500 copies of *A* and *B* and with K_D = 10^{-8} M. Kinetics of reactions with different levels of B_{i0}, ranging from 0 to 10000 were analyzed at K_{D_i} = 10^{-5} M. At the beginning of the reaction the entities *A*, *B* and B_i are present as free monomers. Almost instantaneously, all the monomers of *A* are bound to either *B* or B_i, proportional to their relative abundance. As expected, the larger the value of B_{i0}, the longer the system takes to equilibrate (figure 1A). This is because the forward rate constants of the formation of *AB* and AB_i are assumed to be equivalent. Consequently, the levels of AB_i may be transiently higher than that of *AB* (see figure A6 panel A for the case where the reverse rate constant is equal). AB_i levels decrease while *AB* levels increase since AB_i are more likely to dissociate than *AB.* (Figure 1A). As a result, the average time taken (from 50 replicates) for the formation of *AB* (90% of the maximum possible level, rise time) increases with rising levels of B_{i0} (figure 1B).

We also examined the effect of variation in the dissociation constants of low-affinity interactors, on the formation of *AB.* This was done by varying K_{D_i} from 10^{-4} M to 10^{-6} M while maintaining a B_{i0} level of 5000. Again, as expected, decreasing K_{D_i} (increasing the binding strength of the low affinity binder) decreases the rise time of *AB* (figure 1C). When K_{D_i} is greater than 10^{-4} M, the formation of AB complex breaches the 90% level almost instantaneously. At the other extreme, when K_{D_i} is

smaller than 10^{-6} M (getting closer to K_D), intuitively, the rise time of the AB complex tends to infinity (figure 1D).

To calculate the amount of *AB* found at equilibrium, we have analysed the steady state levels of the AB complex at 625 individual combinations of 25 B_{i0} levels ranging from 0 to 9600 and 25 K_{D_i} values ranging from 10^{-3} M to 10^{-8} M, each replicated 25 times. We observed that AB is the major product as long as K_{D_i} is greater than 10^{-7} (figure 1E). However, B_{i0} levels start to affect the amount of *AB* formed beyond 5000 molecules. Interestingly, variance in the outcome of the reaction also increases with increasing B_{i0} and increasing K_{D_i} (figure 1F). This is quite counterintuitive as one would expect the variance from Gillespie stochastic simulations to decrease with an increase in the size of the system. The two regions in figure 1F with low variance are either dominated by *AB* or AB_i and the 'twilight zone' where both complexes are found in comparable amounts has high variations.

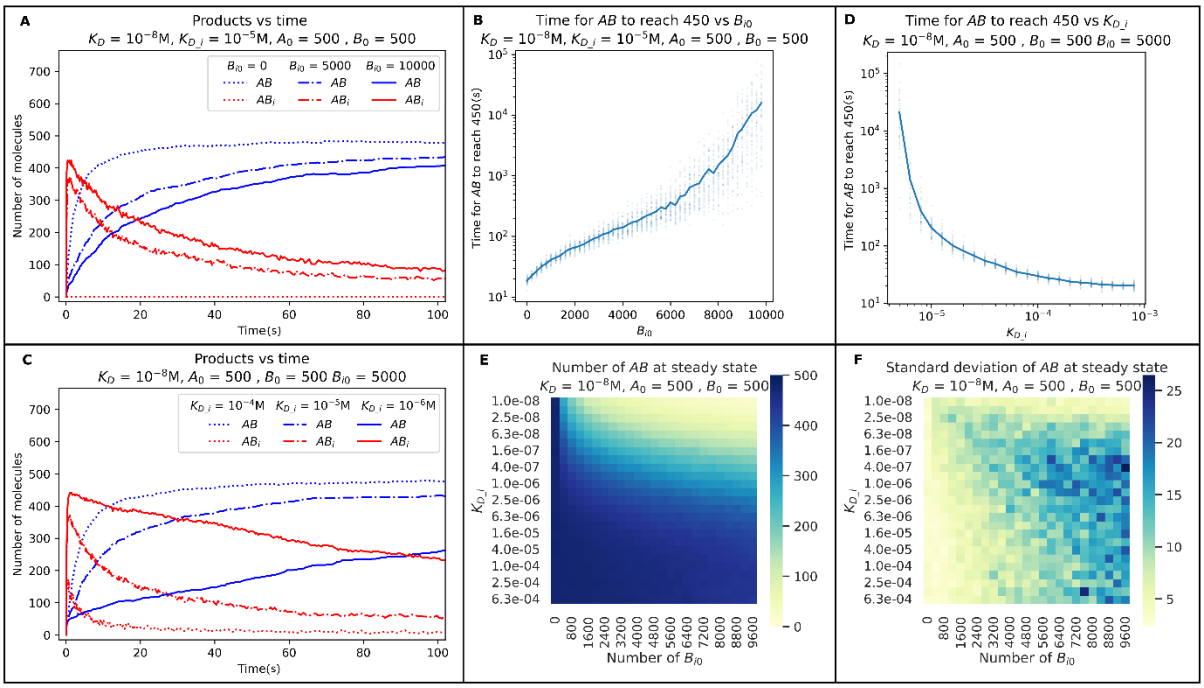


Figure 1: The dependence of complex formation on the starting levels and dissociation constants of the low affinity binder B_i. The kinetics of *AB* (blue lines) and AB_i (red lines) formation at different B_{i0} and K_{D_i} (panels A and C respectively). The effect of B_{i0} and K_{D_i} on the number of *AB* at steady state (average of 25 replicates per box in panel E) with the standard deviations in panel C. The time taken for *AB* to reach 450 molecules (90% of the maximum possible value of 500) at different B_{i0} and K_{D_i} levels (panels B and D respectively). Panels A and C show single trajectories taken from triplicate simulations. The lines in panels B and D are averages over 50 replicates with individual data points shown as scatter plots. The parameters used are shown above the figures.

Since low affinity interactors acting on one component of the high affinity interaction can significantly impact the rate of high affinity interaction, we sought to assess the effect of low affinity interactors on both the high affinity binders.

2.1.2 Both strong interactors face competition.

In addition to low affinity interactors of A (B_i), we now also consider the presence of interactors of B (A_i) in the reaction. A_i interacts with B and forms the complex A_iB (see equations E1, E2 and E3). The dissociation constants of AB_i and A_iB are both described by a single term, K_{D_i}, as we consider them to be equivalent. For this reason, all the results only describe/show AB_i, while the equivalent A_iB is not shown. As in the previous section, the binding affinity of the tight binder (K_D) is 3 orders of magnitude stronger than K_{D_i}. We analyzed the effect of varying the starting levels of B_i and A_i, as well as their binding affinities K_{D_i} on AB formation using Gillespie stochastic simulations.

We first examined the effect of starting levels of B_i and A_i on AB formation. Simulations were set up with 500 copies of A and B and with $K_D = 10^{-8}$ M. The kinetics of reactions with different levels of A_{i0} and B_{i0}, ranging from 0 to 10000 were analyzed at $K_{D\_i} = 10^{-5}$ M. We observed that the presence of low affinity binders increases the rise time of AB (figure 2A). The time taken for AB to equilibrate is an order of magnitude higher than what we noticed when only B_i was present in the system (figures 1A and 2A). Also, the rise time increases exponentially with increasing values of A_{i0} and B_{i0} (figure 2B). These results suggest a non-linear effect arising from the addition of one more low affinity binder.

We then investigated the effect of K_{D_i} on AB formation at values ranging from 10^{-4} M to 10^{-6} M. Decreasing K_{D_i} increases the time taken for AB to equilibrate (figure 2C). Surprisingly, at $K_{D\_i} = 10^{-6}$ M (two orders of magnitude weaker than AB), the reaction seems to prefer the formation of the low affinity complex (figure 2C). We also found that the rise time exponentially increases with decreasing dissociation constant (figure 2D).

To investigate this further, we analysed the steady state of these reactions using 625 different combinations of 25 A_{i0} and B_{i0} levels ranging from 0 to 9600 and 25 K_{D_i} values ranging from 10^{-3} M to 10^{-8} M. AB is not the major product if the number of A_{i0} and B_{i0} are both above 3000 and K_{D_i} is below 10^{-6} M (figure 2E). This is despite a K_D of 10^{-8}

M and with 500 copies of both *A* and *B*. The variability in outcome for reactions within this range is also low. Remarkably, the variability in *AB* levels at steady state is comparatively higher betwixt the areas that completely prefer *AB* or AB_i (figure 2F). Thus, the simultaneous presence of the weak binders, A_{i0} and B_{i0}, in the system, has a greater impact on the formation of *AB*, than when only one weak binder is present. This enhancement is sufficient to allow the weaker complex to become the major product. Note that adding A_i to the system does not increase the concentration of one binder as B_i and A_i binds specifically to *A* and *B,* albeit with low affinity. Henceforth, we refer to this effect of weak binders as *herd regulation*.

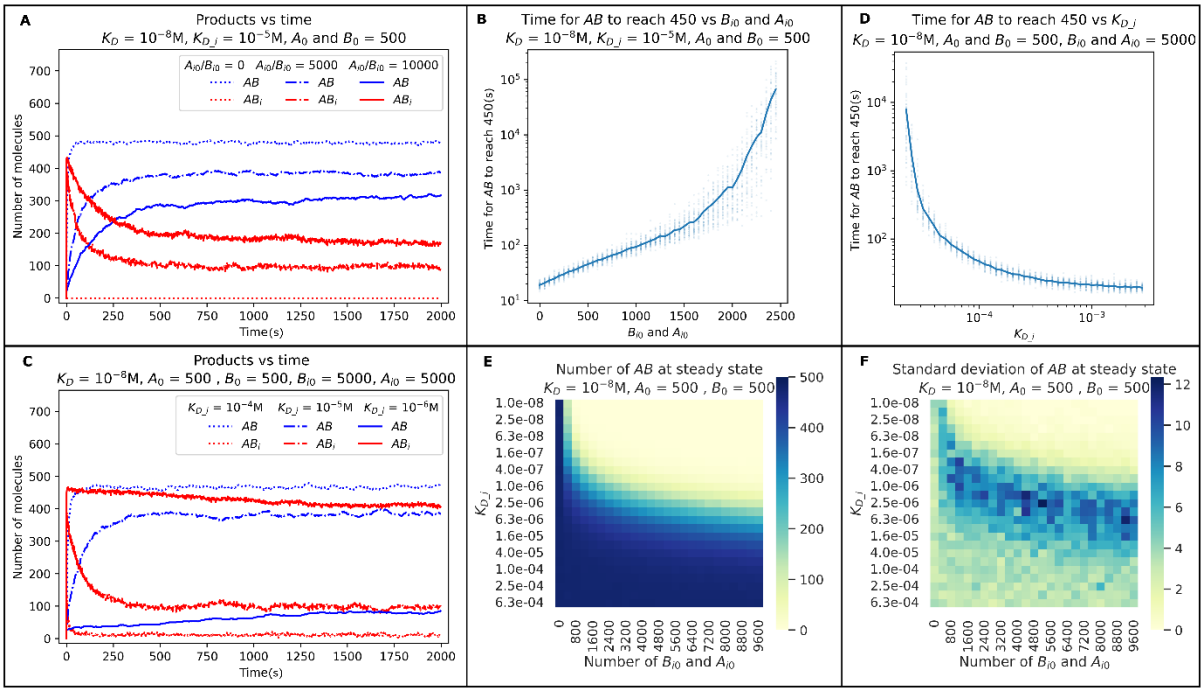


Figure 2: The dependence of complex formation on the starting number and dissociation constant of low affinity binders B_i and A_i. The kinetics of *AB* (blue lines) and AB_i (red lines) formation at different starting levels and dissociation constants of B_i and A_i (panels A and C respectively). The effect of different starting levels and dissociation constants of B_i and A_i on the *AB* levels at steady state (average of 25 replicates per box in panel E and the standard deviations in panel F). The time taken for *AB* to reach 450 (90% of the maximum possible value of 500, black dotted line in panels A and C) at different starting levels and dissociation constants of B_i and A_i (panels B and D respectively). Panels A and C represent single trajectories taken from triplicate simulations. The lines in panels B and D are averages over 50 replicates. The parameters used are shown above the figures.

The presence of monomeric A_i and B_i in the system is likely responsible for the sharp decline in *AB* formation. This is because monomeric *A* and *B* formed by the dissociation of AB_i or A_iB have a higher chance of interacting with B_i and A_i respectively

than with another tight binding monomer. To assess if this is true, we have analyzed the outcome when free A_i and B_i can interact with one another to produce A_iB_i, a system with minimal amount of monomeric A_i and B_i.

2.1.3 Interaction amongst the low affinity competitors decreases herd regulation.

We investigated the effect of low affinity interactors A_i and B_i when they can bind to each other to give rise to the reaction product A_iB_i (E4). The dissociation constant of A_iB_i is denoted by K_{D_cross}. We set the dissociation constant for AB_i and A_iB as 10^{-6} M, a value at which herd regulation favours the low affinity products (figure 2C). We simulated the effect of cross reaction between A_i and B_i on the kinetics of this system using Gillespie stochastic simulations.

We found that decreasing K_{D_cross} (increasing cross reaction strength) increases the rate of AB formation (figure 3A). This is intuitive as it decreases the net concentration of low affinity monomers present. But it is noteworthy that in this section we are expanding on a particular case from the previous section where the low affinity binders were dominant even with 2 orders of magnitude weaker binding affinity. Also, note that even with a strong cross-reaction (compared to K_{D_i}) the system prefers weak binding complexes. Also, the time taken for AB to reach 90% saturation decreases with decreasing K_{D_cross} (figure 3B). Thus, stronger A_iB_i interactions favour more active product formation by sequestering the monomeric A_i and B_i as A_iB_i, allowing the lower probability dissociation reactions to occur and leading to the formation of the tight binding complex.

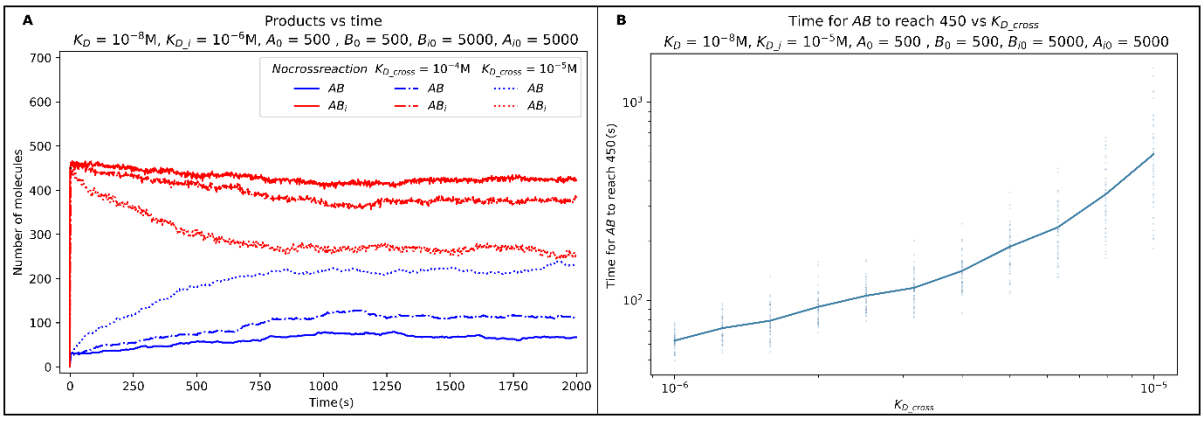


Figure 3: The effect of cross-reaction between low affinity inhibitors on herd regulation using Gillespie stochastic simulations. The kinetics of AB (blue) and AB_i (Red) complexes at cross-reaction dissociation constants (K_{D_cross}) 10^{-4}M and 10^{-5}M (panel **A**). Panel **B** shows the effect of K_{D_cross} on the time taken

for *AB* to reach 450 (90% of the maximum possible value of 500, black dotted line in panel **A**. One of 3 replicates are plotted in panel A. The plot in Panel **B** shows averages of 50 replicates (shown as scatter). Starting values of *A*, *B* = 500, K_D = 10^{-8}M and K_{D_i} = 10^{-6}M. Parameters shown above the figures.

Since we established that low affinity binders can affect the kinetics and equilibrium of tight binding interactions we now aim to formulate a general model that describes this effect (herd regulation).

2.1.4 General model for describing herd regulation with multiple interactors

In this section, we have formulated a general model that describes the kinetics and equilibrium of a herd regulated system based on sets of ordinary differential equations.

Consider the *N* bimolecular association reactions simultaneously happening in the same reaction volume.

$$a_n A_n + b_n B_n \rightleftharpoons c_n C_n$$

Here a, b and c are the stoichiometric coefficients of A, B and C respectively. And n denotes the individual reaction involved in the process. For every $n \in 1,2 \ldots N$

The net rate of formation of complexes for these reactions is defined as the sum of forward and reverse rates at any given time point. The following set of equations defined the rates of formation of each complex.

$$\frac{d[C_n]}{dt} = k_{forward_n}[A_n]^{a_n}[B_n]^{b_n} - k_{reverse_n}[C_n]^{c_n}$$

Here $[A_n]$, $[B_n]$ and $[C_n]$ represent the concentration of the respective entities at a given point in time and, $k_{forward_n}$ and $k_{reverse_n}$ represent the forward and reverse (on and off) rate constants for a reaction.

The rate of change of the complex is 0 at equilibrium and K_D is the ratio of the reverse and forward rate constants. Thus, the above equation simplifies to the following set of equations that describe the system at equilibrium.

$$[C_n]^{c_n} = \frac{[A_n]^{a_n}[B_n]^{b_n}}{K_{D_{C_n}}} \quad \text{(E5)}$$

The equilibrium states can be determined by solving the above set of equations.

The states of A_n and B_n at any given time can be defined in terms of their initial concentrations, the concentration of complexes that they are part of at a given time and their stoichiometry.

$$[A_n] = \left[A_{n_0}\right] - \sum_{i \in S_A} \frac{a_n}{c_i}(C_{i_0} - C_i)$$

$$[B_n] = \left[B_{n_0}\right] - \sum_{i \in S_B} \frac{b_n}{c_i}(C_{i_0} - C_i)$$

Here S_A is the set of complexes that consume A_n and S_B is the set of complexes that consume B_n. The concentration terms with subscript 0 are the initial concentrations of the respective entities.

This model can be used to generate equations that describe specific cases of herd regulation. For example, the following set of equations describes the system of reactions used in the study i.e. E1, E2, E3 and E4 (methods section 4.1).

$$\frac{d[AB]}{dt} = k_1([A])([B]) - k_{-1}[AB] \quad \text{(E6)}$$

$$\frac{d[AB_i]}{dt} = k_2([A])([B_i]) - k_{-2}[AB_i] \quad \text{(E7)}$$

$$\frac{d[A_iB]}{dt} = k_3([A_i])([B]) - k_{-3}[A_iB] \quad \text{(E8)}$$

$$\frac{d[A_iB_i]}{dt} = k_4([A_i])([B_i]) - k_{-4}[A_iB_i] \quad \text{(E9)}$$

The equilibrium states can be deduced by solving,

$$[AB] = \frac{[A][B]}{K_{D_{AB}}} \quad \text{(E10)}$$

$$[AB_i] = \frac{[A][B_i]}{K_{D_{AB_i}}} \quad \text{(E11)}$$

$$[A_iB] = \frac{[A_i][B]}{K_{D_{A_iB}}} \quad \text{(E12)}$$

$$[A_iB_i] = \frac{[A_i][B_i]}{K_{D_{A_iB_i}}} \quad \text{(E13)}$$

Since the reactants are shared by the reactions *A*, *B*, A_i and B_i are affected by multiple reactions and can be expressed as shown below.

$$[A] = [A_0] - ([AB] - [AB_0] + [AB_i] - [AB_{i0}])$$

$$[B] = [B_0] - ([A_iB] - [A_iB_0] + [AB] - [AB_0])$$

$$[A_i] = [A_{i0}] - ([A_iB] - [A_iB_0] + [A_iB_i] - [A_iB_{i0}])$$

$$[B_i] = [B_{i0}] - ([AB_i] - [AB_{i0}] + [A_iB_i] - [A_iB_{i0}])$$

Here the concentrations with a subscript of 0 imply the initial concentration

Using equations E6-E13 we computed the trajectories and equilibrium states for herd regulation with i) only B_i in section 9.4.1, ii) with B_i and A_i in section 9.4.2 and iii) where B_i and A_i cross-react in section 9.4.3. The results from this continuous model closely match the results obtained from Gillespie stochastic simulations.

Note that the equilibrium state only depends on the dissociation constant and is completely independent of the forward and reverse rate constants. Thus, our assumption of diffusion-limited forward rate constant being equal for all reactions only affects the results describing the kinetics of herd regulation. The concentrations of the complexes approach that of the equilibrium values regardless of the combinations of rate constants used. This is demonstrated in section 9.6 where we have studied the trajectories for the formation of *AB* and AB_i when we fix the reverse rate constant to 10^{-3} s^{-1}. These trajectories lack the initial transient peak for AB_i but still converge to the same values as corresponding trajectories with a fixed forward rate constant.

In the last four sections, we have explored the effect of low-affinity interactors on tight binders and formulated a general model that describes herd regulation. In the next two sections, we have examined the effect of herd regulation in deterministic biological contexts and have explored its potential impact on binary outcomes.

2.2 Herd regulation in biological processes.

Here we will illustrate how herd regulation could affect biological processes. In the first case, we will look at sex determination in *Drosophila melanogaster* where a two-fold increase in X chromosome-linked signal generates a binary response during the activation of the *Sxl* gene which determines the female identity. The second example is an illustration of a biological system of receptor and ligand that depend on threshold value determinants.

2.2.1 Example 1: Low affinity binders are responsible for the absence of *Sxl* activation in *Drosophila melanogaster* males

Sexual identity in *Drosophila melanogaster* depends on the female-specific activation of the binary switch gene Sex-lethal (Sxl) before cellularization at cycle 14 (Cline, 1984). Sxl activation, in turn, is regulated by the Sxl establishment promoter *Sxl-Pe*. This process is dependent on X-linked signal elements (XSEs) and autosomal repressors. As females have two X chromosomes, they are able to produce twice the amount of XSEs as compared to males and can specifically activate *Sxl-Pe*. By contrast, *Sxl-Pe* remains off in males who only have one X chromosome. (In insects, the Y chromosome does not contribute to male determination unlike in mammals.). The molecular identities of XSEs are well known (Cline, 1988; Cline and Meyer, 1996; Erickson and Cline, 1993, 1991; Kappes et al., 2011; Keyes et al., 1992; Salz and Erickson, 2010). However, the precise mechanism underlying how the doubling of XSE induces the activation of *Sxl-Pe* remains unknown (Deshpande et al., 1995). Of note, no direct repressor that binds to *Sxl-Pe* during early nuclear cycles has been identified, thus far. Moreover, mutations in deadpan (*dpn*) and groucho (*gro*), two known negative regulators of *Sxl-Pe*, display relatively modest phenotypes (Chen and Courey, 2000; Lu et al., 2008; Mahadeveraju et al., 2020; Paroush et al., 1994; Younger-Shepherd et al., 1992).

Here we test the model that multiple low-affinity binders, capable of interfering with the interaction between activators and *Sxl-Pe*, function as regulators of *Sxl*. As the repression depends on several low affinity binders, it is relatively straightforward to explain why the forward genetic screens were unable to identify corresponding mutants. Single mutations typically isolated in the genetic screens will not display the requisite phenotype.

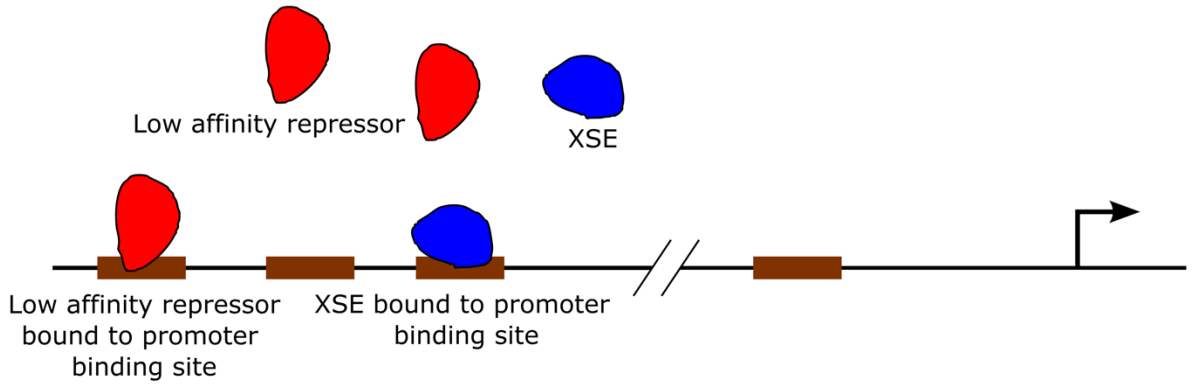


Figure 4: Components of our model of *Sxl-Pe* regulation. The blue objects represent XSEs while the red objects represent the low affinity repressors. The brown boxes represent protein binding sites on Sxl-Pe. The arrow depicts the transcription start site.

Our model posits that sex determination involves one tight binding complex, *AB*, regulated by weaker complexes AB_i (refer to section 2.1.1, equations E1 and E2, and figure 4). Here, the DNA binding sites in *Sxl-Pe* and XSEs constitute tight binders (*A* and *B*) and form the activating complex (*AB*). We assume that binding of the low affinity repressors inhibits the promoter from binding to the activator, directly or indirectly. Low affinity interactors (B_i) are repressors of *Sxl-Pe* that form the repression complex (AB_i). We assumed that there are 50 binding sites in *Sxl-Pe* (Jinks et al., 2003), 200 XSE activators and 1000 low affinity repressors. The dissociation constant (K_D) of the active complex is 10^{-8} M. The repression complex has a 100 fold lower dissociation constant (K_{D_i}) of 10^{-6}M. Here again, we considered identical forward reaction rates, whereas the reverse reaction rates are determined by the dissociation constant. Both XSEs and repressors compete for DNA binding sites in *Sxl-Pe*. We consider *Sxl-Pe* to be engaged in active transcription when 90% (45) of the binding sites are occupied by XSEs. Note that the values that we have used are estimations due to unavailability of biochemical characterization to obtain such values.

We used Gillespie stochastic simulations to study the kinetics of this system. We calculated the amount of XSEs and repressors bound to *Sxl-Pe* binding sites after 100 units of simulation time at different starting levels of XSEs and repressors. The simulation time scales are relative to the rate constants and may not correspond numerically to biological times.

We first assessed the activity of *Sxl-Pe* at different starting values of XSEs and repressors. As expected, the promoter is predominantly occupied by the repressors when activator levels are low. Increasing XSE levels elevates the number of activating complexes (figure 5A). Subsequently, we examined the effect of changing repressor levels on *Sxl-Pe* activity. Increasing the starting levels of repressors decreases the number of activator-bound *Sxl-Pe* sites. The binding sites are occupied by an equal number of activators and repressors when the starting levels of repressors are twice that of the starting levels of XSEs. This is despite a 100-fold difference in binding affinities (figure 5B).

Next, we focused on the time taken for the activation of *Sxl-Pe,* both in males and females. In wild-type *Drosophila melanogaster* embryos, activation of Sxl-Pe results in female fate whereas it remains off in males. Furthermore, this is a time-sensitive process as *Sxl-Pe* is active only between nuclear cycles 11-14 and the sex of the embryo is determined by early cellularization. (This is a narrow time window spanning just three nuclear cycles and takes about 30 minutes or ~2000 seconds.). For this, we analyzed the ratio of time taken for *Sxl-Pe* activation in male nuclei and female nuclei in *Drosophila* embryos where sex determination occurs autonomously. The calculations accounted for the fact that the XSE levels in females are twice that in males. We found that this ratio is in a narrow range between 2 to 3 even when activator levels are varied between 200-1000 (figure 5C). Thus, activation of *Sxl-Pe* in males takes approximately twice as long as compared to females. We arbitrarily selected a test case where the starting level of XSE was 200 (for males) to study the time dependence of sex determination. We analyzed this over a population of 10,000 male and female simulations, each corresponding to a single nucleus. The time distributions in males and females for *Sxl-Pe* activation show two distinct peaks corresponding to the two sexes, with males (on average) taking appreciably more time for activation than females (figure 5D). Note that Sxl activation occurs over a range of time. Importantly, this pattern of nearly non-overlapping distributions indicates that within a given time interval, *Sxl-Pe* is activated only in females. There is however a small overlap between the male and female distributions (purple region in Figure 5D). This indicates that in a small subset of male nuclei, *Sxl-Pe* is turned on and vice versa in females. If the cellularization event occurs between the two peaks (green dotted line in Figure 5D), our model can temporally distinguish male and female *Sxl-Pe* activation.

Thus, our simulations suggest that the time constraint imposed upon the activation of *Sxl-Pe* is significantly responsible for its female-specific activation. Furthermore, in the absence of such constraints, Sxl-Pe can be active even in male embryos, which has been demonstrated to lead to detrimental consequences (Cline and Meyer, 1996). Altogether, our analysis recapitulates both the sex-specificity and the temporal dynamics of *Sxl-Pe* activation. Increasing amounts of low affinity repressors increase the time taken for activation for both males and females. The effect of 1000 (half), 4000 (double) and zero low-affinity repressor on sex determination is shown in figure A2.

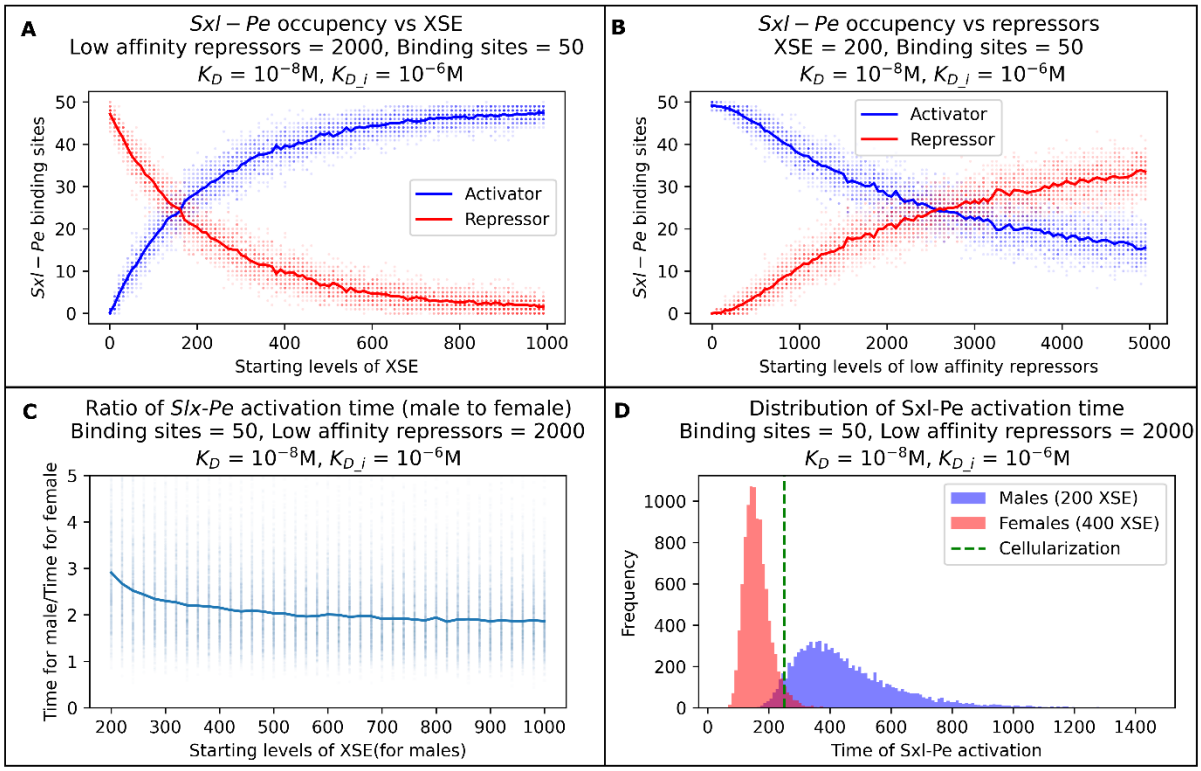


Figure 5: Model of sex determination with one set of low affinity binders. The amount of activating and repressing complexes formed when time = 100 units, at different starting values of XSEs and repressors (Panels **A** and **B** respectively). The ratio of time taken by males vs females to activate *Sxl-Pe* (XSEs occupy 90% of all available *Sxl-Pe* binding sites) (Panel **C**). Panel **D** shows the distribution of time taken for male and female cells to activate across a population of 10000 cells. The simulations were replicated 50 times for each data point in panels **A** and **B** (individual data points shown as scatter of the same colour), and 10000 for panel **C**.

2.2.2 Example 2: Signaling thresholds in biological systems can be explained by herd regulation

In general, molecular thresholds are crucial in several biological signalling events ranging from cell fate determination in early development to neuronal action potentials. Without such thresholds, signals resulting from stochastic noise (inherent to crowded

cellular environments) would be able to aberrantly activate (or repress) biological processes. In this section, we describe a general molecular mechanism of how herd regulation can be instrumental in establishing such thresholds in biological systems.

Consider the example where receptor *A* is activated by binding a signalling molecule *B* to form the receptor-signal complex *AB*. The system also contains low affinity binders for the receptor and signalling molecule (B_i and A_i respectively) as we have explored in the previous sections. The receptor-signal complex is assumed to have a dissociation constant of 10^{-8} M whereas both the low affinity complexes have a dissociation constant of 10^{-6} M. We modelled the kinetics of this system using the Gillespie stochastic simulation with 50 receptors and 500 signalling molecules. This is in keeping with the observation that signalling molecules are usually in excess compared to the number of receptors (Cima, 1994). The system also contained 500 copies of each of the two low affinity binders. We monitored the levels of the receptor-signal complex (amount of *AB* formed) after 10000 units of simulation time, starting with different levels of the signal (*B*). The time of 10000 units was chosen to represent a phase of the reaction that is significantly distant from the fast-starting kinetics (see Figure A1A for a comparison of different time cutoffs). We simulated this system at different starting receptor levels and at levels of the low affinity receptors.

We observed a sigmoidal-like relationship between the number of receptor-signal complexes and the starting levels of the signal. When the number of signaling molecules in the system is below a threshold the rate of increase of the number of receptor-signal complexes is gradual. As soon as the number of signaling molecules exceeds the threshold the number of receptor-signal complexes increases sharply (sigmoid-like). Increasing the levels of the receptor molecule does not alter the threshold but only increases the final levels of the active receptor (figure 6A). Interestingly, we see that the threshold value is regulated by the number of low affinity receptors in the system (figure 6B). Whereas, varying the levels of low affinity signalling molecules does not affect the threshold (figure A1C). Nevertheless, the probability of collisions that lead to the tight binding complexes should approach zero on either increasing the low affinity signals or low affinity receptors. Without low affinity binders the relation between B_0 and *AB* is linear until *A* is exhausted (figure A1D). Thus, the signal threshold is dependent on the level of low affinity receptors in the system, whereas the amplitude of the signal is determined by the number of receptors.

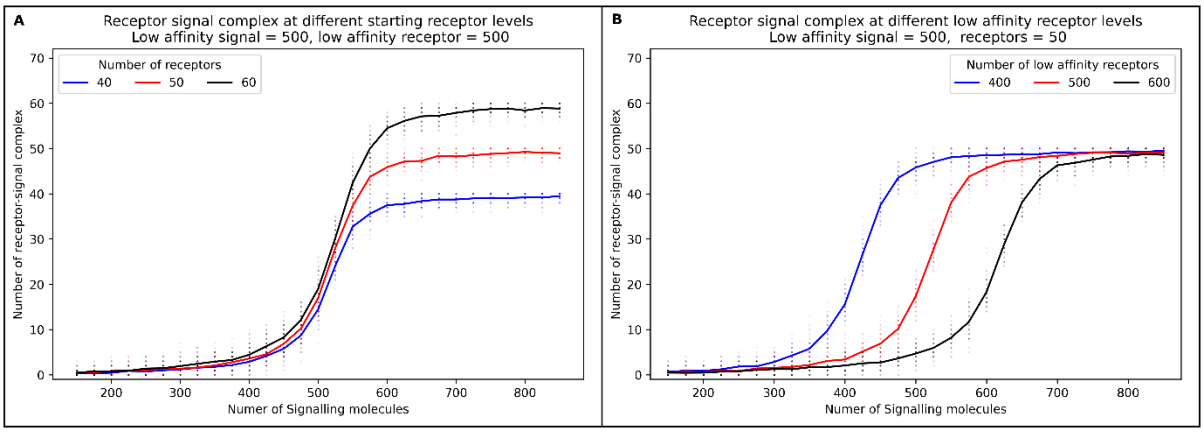


Figure 6: Modelling signalling thresholds in biological systems. All traces (averages of 50 replicates, individual data points shown as scatter) show the relationship between the active receptor and the number of signal molecules and low affinity complexes at different starting levels of the signal. Panel **A** shows this relation at three different starting receptor levels. Panel **B** shows this relation at different starting levels of the low affinity receptor. Plots are averages of 50 replicates and the scatter of the same colours shows the individual data points.

Iyer et al. (2022) analyzed the specification of different cell types under the influence of the Wingless morphogen gradient. They proposed that cellular compartmentalization and receptor promiscuity were critical to elicit a specific response to confer unique cell type identities. Interestingly, this was experimentally verified using the modulators of *Wingless* activity, *Dally* and *Dally like* (B_i) which compete with *Frizzled-2* (*B*) receptor for *Wingless* (*A*) binding. The herd regulation model supports and extends these observations by providing straightforward mechanistic explanation for the observed phenomena. We also suggest the presence of undocumented low affinity binders of *Frizzled-2* (A_i). In general, our model could be applied to a number of such biological systems where the outcome is contingent upon molecular thresholds.

In this section, we have investigated two biological scenarios where herd regulation might play a role. In the next section, we provide the design of an experiment which can validate the general concept of herd regulation.

2.3 Experiment to validate herd regulation based on DNA-DNA interaction

Our model of herd regulation holds for all types of molecular associations such as protein-protein, protein-DNA, DNA-DNA interactions etc. The binding affinity between single-stranded DNAs can be designed based on complementarity and melting temperature. Here we present the results of herd regulation in DNA-DNA interactions. In this example, each of *A*, *B*, A_i and B_i are single-stranded DNA (Figure 7). The

sequences are such that the binding affinities are in the order, $AB > AB_i$,~= $A_iB > A_iB_i$. Also, the sequences are designed to minimize self-complementarity and binding affinity of AA_i and BB_i.

The binding affinities between the species are controlled by mismatches between the sequences i) At the experimental temperature, the ΔG of binding of each combination corresponded to the binding energy. ii) Hybridization of ssDNA from two different species results in a unique restriction site (i.e. this location in the sequence has mismatches in all other combinations)

Let the unique restriction sites in AB, A_iB, AB_i and A_iB_i be, RE1, RE2, RE3 and RE4 respectively. Before hybridisation, each of the ssDNA (easily procured as oligonucleotides) is converted into circular ssDNA using a ssDNA ligase. These circular ssDNA which correspond to A, B, A_i and B_i are incubated together at the experimental temperature till equilibrium. After this, the reaction is quickly cooled to a temperature well below the lowest melting temperature to avoid any deviations from equilibrium. This should result in a mix of AB, A_iB, AB_i and A_iB_i products. To measure the concentrations of each product, four equivalent aliquots are made from this reaction mixture. To measure the amount of AB, the enzymes RE2, RE3 and RE4 are added to one of the aliquots to linearise double-stranded DNA corresponding to A_iB, AB_i and A_iB_i. The linear dsDNA is then removed by digestion using exonucleases. This is repeated for the other 3 products by using the enzyme combinations; RE1, RE3 and RE4 for A_iB; RE1, RE2 and RE4 for AB_i and RE1, RE2 and RE3 for A_iB_i. The products of this step contain the circular dsDNA of only one complex. DNA purification removes protein and buffer remains from enzyme reactions. The concentration of DNA is then measured using UV spectrophotometry at 260nm wavelength or by realtime PCR experiments.

This experimental method allows us to measure the steady state of the system after manipulating the concentrations and binding affinities of each of the reactants individually. Different reactions with different combinations of binding affinities and concentrations can be set up such that our results from steady-state (figures 1E, 2E and 6) can be replicated.

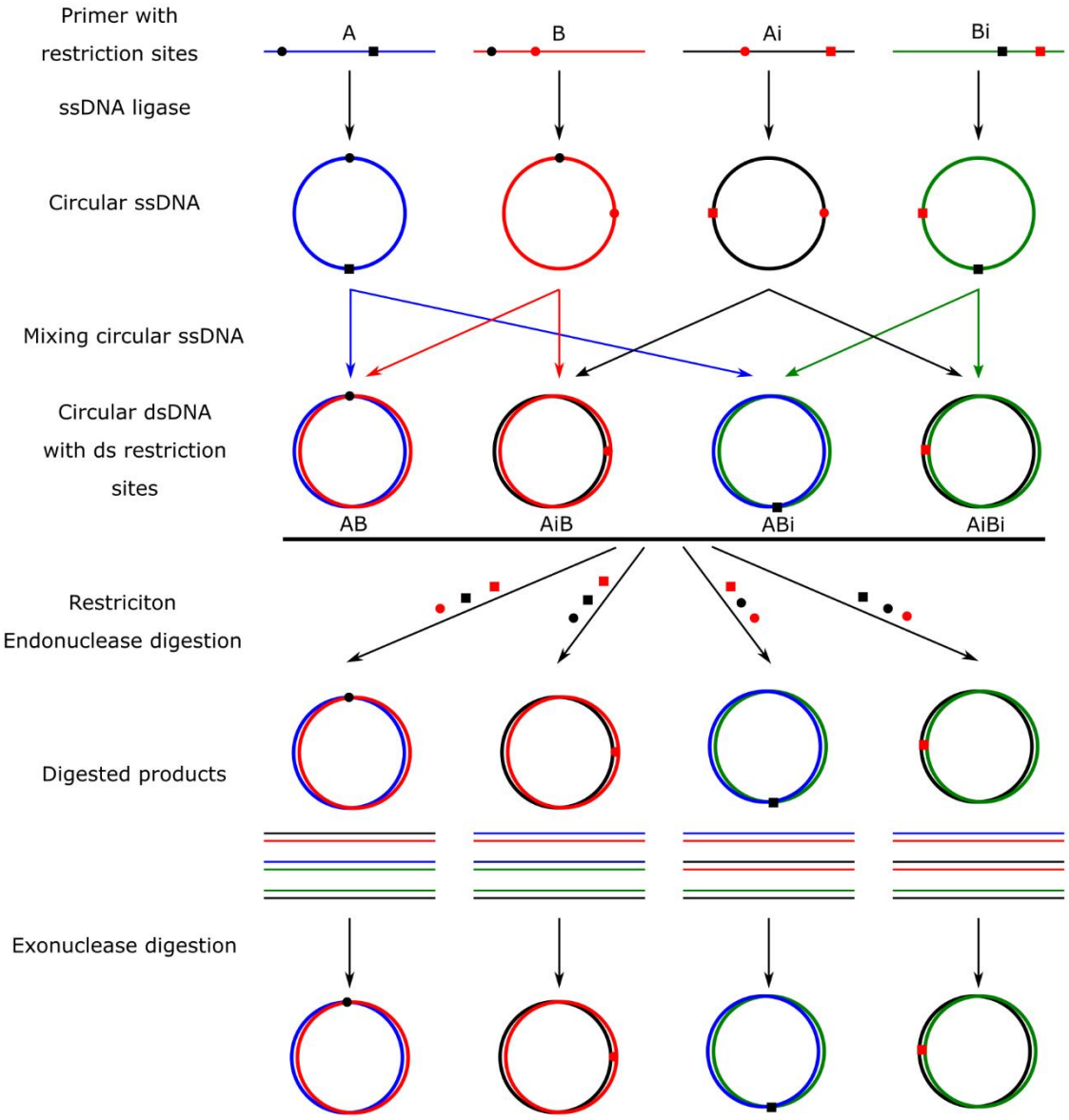


Figure 7: Schematic of the experiment to validate the effect of herd regulation. DNA strands are coloured blue, red, black, and green for *A*, *B*, A_i and B_i. red and black squares and circles represent different restriction sites and corresponding restriction enzymes. Different steps of the reaction are shown in the left-hand side.

3 Discussion

Molecular investigations of biological processes almost always focus on interactions that are high affinity and specific. Seldom, if at all, is any importance given to interactions of low affinity/specificity. In part, this could be due to our inability to detect such interactions. In this study, we examine how the presence of low affinity interactors

modulates the association of high affinity interactors. We term such modulation of tight binding events by low affinity interactors as ‘herd regulation’.

To illustrate herd regulation, we have considered a simple system of two interactors (*A* and *B*) that come together to form a dimeric complex (*AB*). The kinetics of AB formation was modulated in the background of low affinity interactors to both *A* and *B* (B_i and A_i respectively). In this case, we found that the rise time of the *AB* complex increases with changes in the levels of the low affinity interactors. Surprisingly, we find situations where *AB* is not the dominant product of the reaction, despite K_D of *AB* several orders of magnitude higher. A_i and B_i when acting alone, without the presence of the other (section 2.1.1), are weak binders and offer weak competition. Only when acting in consort (section 2.1,2) do they become potent competitors. This can be explained by the lack of monomeric *A* and *B* in the system, as all available *A* and *B* are consumed by the formation of *AB*, A_iB and AB_i. This is the case since *A* and *B* contribute to the formation of two complexes each, whereas A_i and B_i are a part of only one complex each. Thus, the products of the dissociation of A_iB or AB_i have a higher probability of encountering A_i or B_i compared to *A* or *B*. This study effectively predicts the ranges of both the concentrations and dissociation constants at which herd regulation would be effective. In general, herd regulation is affected by the quantity of the low affinity binders as well as their respective binding efficacies (quality). Interestingly, our data argues that the quality is more impactful than the quantity. We successfully reproduce all of these results almost exactly using a separate continuous method.

In our simulations, we have demonstrated herd regulation of a binary interaction of *A* and *B* (giving *AB*) with interference from low affinity competitors A_i and B_i. In the crowded cellular environment, reactions could be more complex, involving multiple reactants with varying stoichiometry. It is thus relatively straightforward to imagine that herd regulation would impact such complex interactions more than it does the simple *AB* system. The other important aspect here is that the low affinity competitors could be multiple species. For instance, the interactor *A* could have many homologous or structurally analogous molecules in the cell, all of which could collectively constitute the A_i. Given the sizes and abundances of protein families, and the comparatively low number of folds, our estimates of the levels of the low affinity competitors are

conservative. Please note that similar arguments would also hold for herd regulation involving different types of molecules such as DNA, RNA etc.

Unlike our simulations, in a cell all the potential interactors may not coexist or be active at the same time. It is conceivable that before one of the strong binding interactors is introduced into the system, the other binding partner and the low affinity interactors are already present. Such a situation would only magnify the effects we have discussed above. i.e., the relative rates of reactions could be slower than what we have simulated, and the time taken to equilibrate the system would be longer. For instance, during sex determination in *Drosophila*, the DNA binding sites in *Sxl-Pe* exist in the cell even when the XSEs are absent. In this situation, it is reasonable to assume that these DNA binding sites are bound to a cohort of low affinity repressors. The XSEs produced during early development (nuclear division cycles 7-13) will have to displace the low affinity binders for the DNA binding site to activate *Sxl*.

In our computations that study kinetics, we have assumed that the forward rate constants ($k_{forward}$ or k_f) for the high and low affinity interactors are the same. This is not an unusual circumstance in biological processes. For instance, proteins that recognize specific sequences of DNA initially bind non-specifically and diffuse along the DNA to its cognate binding site (Kim and Larson, 2007; Shimamoto, 1999; Wang et al., 2006). Even, non-specifically binding proteins would also have a similar rate of association with the DNA as specific binders (Cima, 1994; Schreiber et al., 2009). The rates of dissociation, however, are vastly different and determine the binding affinities. A similar argument can be made regarding protein-protein (or protein-peptide) and protein-RNA interactions, especially if the binding was predominantly due to complementarity in electrostatics. In the studies of equilibrium/steady state this assumption of comparable forward rate constants is not required. This is because the equilibrium state is only dependent on the dissociation constant and starting concentrations of the reactants.

It should be noted that the herd regulation model presented in this study is basic. We built such a model to set up a platform where one could build in additional constraints for future investigations. Such future directions include: i) The effect of herd regulation on oligomerization and/or polymerization reactions; ii) Our models demonstrate regulation when there is a fixed quantity of the reactants. A more realistic model would

also incorporate rates of their production and degradation; iii) Modeling interactions where there are transition states and more involved kinetics. iv) Proving herd regulation experimentally would constitute a significant challenge as low affinity interactors are mostly undetected. We have addressed this by suggesting an *in vitro* experiment involving circular single stranded DNA. We believe that these experiments could validate the findings reported in results section 2.1. We also have a suggestion for *in vivo* investigation (see below). The present study lays down the basis to investigate such questions, which is otherwise beyond the scope of this study.

We have illustrated herd regulation in biological context with two examples and also suggest one 'in vitro' experimental stratergy to verify the model. In one example, we use protein-protein interactions akin to signal-receptor activation and in the other, we use DNA-protein interactions to illustrate sex determination in *Drosophila melanogaster*.

In the example of signalling, we resort to our simple *AB* model (*A* = receptor, *B* = signal or a ligand). The signal (*B*) can bind its cognate receptor *A* or any of the low affinity competitive/homologous receptors A_i. Once *B* is at a level (threshold) where it has saturated all the *A* and A_i's the kinetics is only dependent on the level of B_i (low affinity competing signals). Thus, exhaustion of A_i switches the system, from herd regulation influencing both the tight binders, to one where it affects only a single tight binder. Thus, the threshold is primarily dependent on the amount of A_i, the low affinity receptor, present in the system. This formulation also helps explain earlier observations and computations where the presence of promiscuous receptors (A_i) determines cell fate in *Drosophila melanogaster* development (Iyer et al., 2023). Note that our model is significantly simpler than the model proposed by Iyer et al which required invoking compartments and significantly more rate equations to explain this phenomenon. We can explain this with three simple equations (E9, E10 and E11) (section 2.1.4 and appendix section 9.4). This also simplifies the evolution and design of molecular switches. Consider *A*, *B*, A_i and B_i where *A* and *B* interact strongly whereas A_i and B_i are homologs of *A* and *B* which have partially similar binding sites and thus interact with *A* and *B* and with each other. This system will become a molecular switch if there are mutations (engineered or evolved) in the binding interface such that A_iB_i is not formed (see sections 2.1.2 and 2.1.3).

In the example of sex determination, we analysed a binary switch gene, *Sxl*, whose activity confers female identity. *Sxl* activity is primarily affected by the presence of X-linked signalling elements (XSEs) and autosomal repressors. XSEs are derived from the X chromosome. Thus, females with two X chromosomes activate *Sxl-Pe* while the *Sxl-Pe* of males remains inactive. Sex determination is dependent on the activity of *Sxl-Pe* at the end of the syncytial blastoderm stage of embryogenesis (NC10-13). We have modelled sex determination considering the binding sites in *Sxl-Pe* as *A*, XSEs as *B* and the autosomal repressors as low affinity interactors, B_i. Note that many of the *Sxl-Pe* binding sites are concentrated within 400 base pairs in the promoter. (Estes et al., 1995). We found that the presence of low affinity binders temporally resolves *Sxl-Pe* activation between males and females. XX and XY genotypes would determine female and male states respectively as long as the cellularization occurs at a time point between male and female activation. The primary factor that allows for this temporal decision is the amount of XSEs, the cell cycle at which cellularization happens and the amount of low affinity autosomal repressors present in the system. Among these, the amount of XSEs and timing of cellularization has been experimentally perturbed. Mutations in XSE genes are detrimental to females (Cline and Meyer, 1996). Delayed cellularization generates embryos with more *Sxl-Pe* activation and vice versa (Erickson and Quintero, 2007). It would be interesting to see the effect of perturbing the low affinity binder levels *in vivo*, although this would require the identification of multiple low affinity binders which would need to be mutated simultaneously. Hypothetically, changing the level of the low affinity repressors (herd regulators) could affect the male-female ratio in a population (see figure A3 for distributions of *Sxl-Pe* activation at different low affinity repressor levels). Since the sex determination decision is made individually in each nucleus in an autonomous manner, this effect should manifest as a mosaic phenotype as those shown in (Erickson and Quintero, 2007). The presence of such repressors is our prediction. The role of deadpan (*dpn*) and groucho (*gro*) as weak negative regulators is already reported (Chen and Courey, 2000; Lu et al., 2008; Mahadeveraju et al., 2020; Paroush et al., 1994; Younger-Shepherd et al., 1992). However, *gro* does not bind DNA directly and needs to be recruited to the promoter and neither deadpan nor *gro* active in all the cells (sex determination decision happens in all the cells). We speculate that the unknown entities responsible for the recruitment of *gro* along with other yet unknown

entities have a combined ‘herd’ repressive effect on the promoter. Our model of sex determination can be validated only after these low affinity repressors are identified.

It will be important to consider these data in a broader, more general context. Even a cursory examination should reveal that binary decisions constitute important cornerstones of biological systems both at the ‘micro’ and ‘macro’ levels. It is noteworthy that many of the early foundational discoveries, especially in the field of developmental genetics were aimed at uncovering the presence of a single master determinant that functions as a molecular switch. (Sxl is just one such example of the several well documented ‘master switch genes’). As the proper execution of the downstream molecular processes is critically dependent on the early choice, these binary switches have proven to be highly effective means to regulate major outcomes of a variety of determinative events. Notwithstanding the benefits however, one ought to consider a few important caveats that are readily overlooked while considering biological relevance and regulation of binary mechanisms that underlie fate determination. The genetic and biochemical strategies aimed at the identification and characterization of canonical bio-switches (and their regulatory components) invariably rely on phenotypes that are extreme. For instance, the genetic screens that identified regulators of sex determination relied on either male or female specific lethality. (The nomenclature of the relevant mutations *daughterless*, *sisterless a, b* etc. effectively underscores this feature). As the detection of relevant mutations relied on highly penetrant sex-specific lethality, a number of potentially interesting albeit weak mutations were likely dismissed. Their detection is only possible upon simultaneous deletion or addition of multiple weak affinity factors as approximated in our computational approach.

We have also observed an overlap in the distribution of time taken for male and female *Sxl* activation. This suggests that binary switch systems are not strictly binary. Their outcome is actually two states that in many cases can have small (but not insignificant), overlaps. This becomes especially important in the case of cell autonomous determinative events. For instance, the sex of every single cell in a *Drosophila* embryo is independently determined by the total concentration of X-chromosomal elements. As a result, gynandromorphs consist of cells of both sexual identity and, on rare occasions can survive even up to adulthood. While every instance does not result in a dramatic and demonstrable outcome such as this, it is conceivable

that several cells of the 'wrong sex' may be determined in an organism and are eventually eliminated (or rendered ineffective) without any detectable consequence. Analogous scenarios can be envisaged in other biological contexts i.e., a few haltere-like cells could co-exist in wings or some neurons may exhibit partial epithelial traits. Such heterogeneity has been well documented in primordial germ cells in blastoderm stage *Drosophila* embryos while its relevance remains to be determined.

Canonical models suggest that both the initiation and maintenance of the binary switch genes follow a stereotypical path toward a unified outcome. Altogether, while binary decisions may lead to invariable and robust outcomes at a macro- or organismal level, cellular heterogeneity likely exists at a micro- or cellular level. Mechanisms ought to be in place that do not allow such variability to exceed permissible limits leading to detrimental biological consequences. Importantly, the tolerable threshold may differ vastly depending on the cellular or organismal contexts. The mechanistic underpinnings and biological relevance of such mechanisms need closer scrutiny and careful consideration as these represent molecular hurdles that are likely essential components of homeostatic mechanisms.

4 Materials and Methods

4.1 Reactions

Consider the following reaction,

$$A + B \rightleftharpoons AB \tag{E1}$$

Consider the following reactions happening in proximity to E1,

$$A + B_i \rightleftharpoons AB_i \tag{E2}$$

$$A_i + B \rightleftharpoons A_iB \tag{E3}$$

$$A_i + B_i \rightleftharpoons A_iB_i \tag{E4}$$

Here A, B, A_i and B_i, each represent ensembles of biomolecules. The Interactions between any pair of A and B have high affinity whereas the interaction between A_i and B_i, have low affinity. Note that although these interactions are low affinity, they are specific.

4.2 Gillespie stochastic simulations

Differential equations can describe the progression of reactions deterministically. But they assume that the reaction volume is infinitely large, and the stochastic behaviour of molecules is cancelled out. This also means that the concentration terms are continuous. But a cell is a small system with relatively few molecules. In such a system the concentration is no longer continuous, and randomness could often play an important role. We use the direct method of the Gillespie stochastic simulation algorithm (Gillespie, 2013, 1977) to simulate such a system. This is a numerical method for simulating the temporal evolution of systems of chemical reactions with randomness. It involves iteratively sampling reaction events and updating the state of the system based on the chosen reaction and the current state of the system. And thus can be used to study the behaviour of complex chemical systems over time, including the effects of noise and uncertainty on the system's dynamics.

We implemented the Gillespie algorithm, in python3. First, the initial state of the system is set, including the number and types of reactants and the reaction rates for each chemical reaction. Then, for each possible reaction, the propensity, a measure of the likelihood that the reaction will occur in the next time step, is calculated based on the reaction rate and the current state of the system. *A* random number generator is used to choose one of the possible reaction events based on the propensities calculated. The state of the system is then updated by adding or subtracting reactants as appropriate. The simulation time is advanced by a time step calculated based on the inverse of the sum of propensity and a random number. The process is repeated until the system reaches a) a pre-specified time b) a specified number of iterations c) when a product crosses a pre-specified level, or d) steady state (described in section 4.2.2).

4.3.1 Parameters used to generate the results

Section 2.1.1: A_0 = 500, B_0 = 500, K_D = 10^{-8} M and C_0 = 10^{-5}M and N_0 = 1000. B_{i0} and K_{D_i} were varied in the section.

Section 2.1.2: A_0 = 500, B_0 = 500, K_D = 10^{-8} M and C_0 = 10^{-5}M and N_0 = 1000. A_{i0}, B_{i0} and K_{D_i} were varied in the section. Note that in all the cases A_{i0} = B_{i0} and their binding affinities were also kept the same and represented by the common term K_{D_i}.

Section 2.1.3: A_0 = 500, B_0 = 500, A_{i0} = 5000, B_{i0} = 5000, K_D = 10^{-8} M, K_{D_i} = 10^{-6} M and C_0 = 10^{-5}M and N_0 = 1000. K_{D_cross} was varied in the analysis.

Section 2.2.1: A_0 = 50, B_0 = 200, B_{i0} = 2000 and K_D = 10^{-8} M, K_{D_i} = 10^{-6} M and C_0 = 10^{-5}M and N_0 = 1000. B_{i0} and B_{i0} were varied in the section.

Section 2.2.2: A_0 = 50, and K_D = 10^{-8} M, K_{D_i} = 10^{-5} M and C_0 = 10^{-3}M and N_0 = 100.. A_0, B_0 and B_{i0} were varied in the section. The simulations were carried out till the simulation time of 10000 units.

Also, see section 4.3.3 and appendix 9.3 for the justification for the values provided.

4.3.2 Steady state calculations

For reactions with only B_i present, steady-state was assumed to have been reached when the average fluctuation over 1000 steps for every reactant and product is less than 1 (molecule or molecular complex). This was evaluated in intervals of 100 steps.

For reactions with both B_i and A_i, steady-state was assumed to have been reached when the average fluctuation of over 1,000,000 steps for every reactant and product is less than 1 (molecule or molecular complex). This was evaluated in intervals of 100,000 steps.

4.3.3 Calculation of propensities

The propensities were calculated using the following equations

$$k_f = 10^5\ \mathrm{M}^{-1}s^{-1}$$

$$k_r = k_f \times K_D$$

$$p_f = \frac{C_0}{N_0} \times k_f$$

$$p_r = k_r$$

Where k_f and k_r are the forward and reverse rate constant respectively. k_f is diffusion controlled and set at 10^5 $M^{-1}s^{-1}$ (Schlosshauer and Baker, 2002).p_f and p_r are the forward and reverse propensities used in the Gillespie stochastic simulations. C_0 and N_0 are used to scale the second-order forward rate constant (k_f) based on concentration.

For sections 2.1 and 2.2.1, $C_0 = 10^{-5}M$ and $N_0 = 1000$. For section 2.2.2 $C_0 = 10^{-3}M$ and $N_0 = 100$ (Traces for other values of C_0 can be found in figure A1B) (justifications for the values are provided in appendix section 9.3).

4.4 Parameters for generating results from differential equations

The parameters used are the same as those described in 4.3.1, except that the number of molecules were converted into concertation using the following formula based on effective concentrations of the number of particles in the Gillespie simulations, calculated as follows.

$$effective\ concentration = \frac{C_0}{N_0} \times number\ of\ particles$$

Initial concentration of Ai is set to 0 for calculations in sections 9.4.1. The K_{Dcross} for the dissociation of AiBi is set to infinity in all sections except section 9.4.3.

Trajectories were calculated by Runge-Kutta 4th Order numerical integration. The equilibrium states were calculated by solving equations E10, E11 and E12 for *AB*, AB_i and A_iB. The system of equations was solved by sympy module in python3. Real solutions that satisfy stoichiometry are selected.

4.5 Data availability

All scripts/programs used in this study along with the data they have produced can be accessed at

http://sites.iiserpune.ac.in/~madhusudhan/Herd_regulation/figures.tar.gz

8 References

Abrami L, van Der Goot FG. 1999. Plasma membrane microdomains act as concentration platforms to facilitate intoxication by aerolysin. *J Cell Biol* **147**:175–184. doi:10.1083/jcb.147.1.175

Chen G, Courey AJ. 2000. Groucho/TLE family proteins and transcriptional repression. *Gene* **249**:1–16. doi:10.1016/s0378-1119(00)00161-x

Chen Y, Belmont AS. 2019. Genome organization around nuclear speckles. *Curr Opin Genet Dev* **55**:91–99. doi:10.1016/j.gde.2019.06.008

Cima LG. 1994. Receptors: Models for binding, trafficking and signaling. By Douglas A. Lauffenburger and Jennifer J. Linderman, Oxford University Press, 1993, $70.00. *AIChE J* **40**:1089. doi:https://doi.org/10.1002/aic.690400624

Cline TW. 1988. Evidence that sisterless-a and sisterless-b are two of several discrete “numerator elements” of the X/A sex determination signal in Drosophila that switch Sxl between two alternative stable expression states. *Genetics* **119**:829–862. doi:10.1093/genetics/119.4.829

Cline TW, Meyer BJ. 1996. Vive la différence: males vs females in flies vs worms. *Annu Rev Genet* **30**:637–702. doi:10.1146/annurev.genet.30.1.637

Deshpande G, Stukey J, Schedl P. 1995. scute (sis-b) Function in Drosophila sex determination. *Mol Cell Biol* **15**:4430–4440. doi:10.1128/MCB.15.8.4430

Erickson JW, Cline TW. 1993. A bZIP protein, sisterless-a, collaborates with bHLH transcription factors early in Drosophila development to determine sex. *Genes Dev* **7**:1688–1702. doi:10.1101/gad.7.9.1688

Erickson JW, Cline TW. 1991. Molecular Nature of the Drosophila Sex Determination Signal and Its Link to Neurogenesis. *Science (80-)* **251**:1071–1074. doi:10.1126/science.1900130

Erickson JW, Quintero JJ. 2007. Indirect Effects of Ploidy Suggest X Chromosome Dose, Not the X:A Ratio, Signals Sex in Drosophila. *PLOS Biol* **5**:e332.

Estes PA, Keyes LN, Schedl P. 1995. Multiple response elements in the Sex-lethal early promoter ensure its female-specific expression pattern. *Mol Cell Biol* **15**:904–917. doi:10.1128/MCB.15.2.904

Galaway F, Wright GJ. 2020. Rapid and sensitive large-scale screening of low affinity extracellular receptor protein interactions by using reaction induced inhibition of Gaussia luciferase. *Sci Rep* **10**:10522. doi:10.1038/s41598-020-67468-7

Gillespie DT. 2013. Gillespie Algorithm for Biochemical Reaction Simulation. *Encycl Comput Neurosci* **1**:1–5. doi:10.1007/978-1-4614-7320-6

Gillespie DT. 1977. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* **81**:2340–2361. doi:10.1021/j100540a008

Hu W-S. 2017. Kinetics of Biochemical Reactions. *Eng Princ Biotechnol*, Wiley Online Books. doi:https://doi.org/10.1002/9781119159056.ch4

Huh W-K, Falvo J V, Gerke LC, Carroll AS, Howson RW, Weissman JS, O'Shea EK. 2003. Global analysis of protein localization in budding yeast. *Nature* **425**:686–691. doi:10.1038/nature02026

Iyer KS, Prabhakara C, Mayor S, Rao M. 2023. Cellular compartmentalisation and receptor promiscuity as a strategy for accurate and robust inference of position during morphogenesis. *Elife* **12**:e79257. doi:10.7554/eLife.79257

Iyer KS, Rao M, Prabhakara C, Mayor S. 2022. Cellular compartmentalisation and receptor promiscuity as a strategy for accurate and robust inference of position during morphogenesis. *bioRxiv* 2022.03.30.486187. doi:10.1101/2022.03.30.486187

Jiang L, Barclay AN. 2009. New assay to detect low-affinity interactions and characterization of leukocyte receptors for collagen including leukocyte-associated Ig-like receptor-1 (LAIR-1). *Eur J Immunol* **39**:1167–1175. doi:10.1002/eji.200839188

Jinks TM, Calhoun G, Schedl P. 2003. Functional conservation of the Sex-lethal sex determining promoter, Sxl-Pe, in Drosophila virilis. *Dev Genes Evol* **213**:155–165. doi:10.1007/s00427-003-0304-1

Kanehisa M, Furumichi M, Sato Y, Kawashima M, Ishiguro-Watanabe M. 2022. KEGG for taxonomy-based analysis of pathways and genomes. *Nucleic Acids Res* gkac963. doi:10.1093/nar/gkac963

Kappes G, Deshpande G, Mulvey BB, Horabin JI, Schedl P. 2011. The Drosophila Myc gene, diminutive, is a positive regulator of the Sex-lethal establishment promoter, Sxl-Pe. *Proc Natl Acad Sci U S A* **108**:1543–1548. doi:10.1073/pnas.1017006108

Keyes LN, Cline TW, Schedl P. 1992. The primary sex determination signal of Drosophila acts at the level of transcription. *Cell* **68**:933–943. doi:https://doi.org/10.1016/0092-8674(92)90036-C

Kim JH, Larson RG. 2007. Single-molecule analysis of 1D diffusion and transcription elongation of T7 RNA polymerase along individual stretched DNA molecules. *Nucleic Acids Res* **35**:3848–3858. doi:10.1093/nar/gkm332

Li Y-C, Rodewald LW, Hoppmann C, Wong ET, Lebreton S, Safar P, Patek M, Wang L, Wertman KF, Wahl GM. 2014. A Versatile Platform to Analyze Low-Affinity and Transient Protein-Protein Interactions in Living Cells in Real Time. *Cell Rep* **9**:1946–1958. doi:10.1016/j.celrep.2014.10.058

Lu H, Kozhina E, Mahadevaraju S, Yang D, Avila FW, Erickson JW. 2008. Maternal Groucho and bHLH repressors amplify the dose-sensitive X chromosome signal in Drosophila sex determination. *Dev Biol* **323**:248–260. doi:10.1016/j.ydbio.2008.08.012

Mahadeveraju S, Jung Y-H, Erickson JW. 2020. Evidence That Runt Acts as a Counter-Repressor of Groucho During Drosophila melanogaster Primary Sex Determination. *G3 (Bethesda)* **10**:2487–2496. doi:10.1534/g3.120.401384

Maul GG, Deaven L. 1977. Quantitative determination of nuclear pore complexes in cycling cells with differing DNA content. *J Cell Biol* **73**:748–760. doi:10.1083/jcb.73.3.748

Paroush Z, Finley RLJ, Kidd T, Wainwright SM, Ingham PW, Brent R, Ish-Horowicz D. 1994. Groucho is required for Drosophila neurogenesis, segmentation, and sex determination and interacts directly with hairy-related bHLH proteins. *Cell* **79**:805–815. doi:10.1016/0092-8674(94)90070-1

Perkins JR, Diboun I, Dessailly BH, Lees JG, Orengo C. 2010. Transient Protein-Protein Interactions: Structural, Functional, and Network Properties. *Structure* **18**:1233–1243. doi:https://doi.org/10.1016/j.str.2010.08.007

Roth S, Stein D, Nüsslein-Volhard C. 1989. A gradient of nuclear localization of the dorsal protein determines dorsoventral pattern in the Drosophila embryo. *Cell* **59**:1189–1202. doi:10.1016/0092-8674(89)90774-5

Salz HK, Erickson JW. 2010. Sex determination in Drosophila: The view from the top. *Fly (Austin)* **4**:60–70. doi:10.4161/fly.4.1.11277

Schlosshauer M, Baker D. 2002. A General Expression for Bimolecular Association Rates with Orientational Constraints. *J Phys Chem B* **106**:12079–12083. doi:10.1021/jp025894j

Schreiber G, Haran G, Zhou H-X. 2009. Fundamental aspects of protein-protein association kinetics. *Chem Rev* **109**:839–860. doi:10.1021/cr800373w

Shimamoto N. 1999. One-dimensional Diffusion of Proteins along DNA: ITS BIOLOGICAL AND CHEMICAL SIGNIFICANCE REVEALED BY SINGLE-MOLECULE MEASUREMENTS *. *J Biol Chem* **274**:15293–15296. doi:10.1074/jbc.274.22.15293

Simons K, Ehehalt R. 2002. Cholesterol, lipid rafts, and disease. *J Clin Invest* **110**:597–603. doi:10.1172/JCI16390

Wang YM, Austin RH, Cox EC. 2006. Single molecule measurements of repressor protein 1D diffusion on DNA. *Phys Rev Lett* **97**:48302. doi:10.1103/PhysRevLett.97.048302

Wartlick O, Kicheva A, González-Gaitán M. 2009. Morphogen gradient formation. *Cold Spring Harb Perspect Biol* **1**:a001255. doi:10.1101/cshperspect.a001255

Younger-Shepherd S, Vaessin H, Bier E, Jan LY, Jan YN. 1992. deadpan, an essential pan-neural gene encoding an HLH protein, acts as a denominator in Drosophila sex determination. *Cell* **70**:911–922. doi:10.1016/0092-8674(92)90242-5

9 Appendix

9.1 Effect of changing levels of low affinity signaling molecules on the threshold.

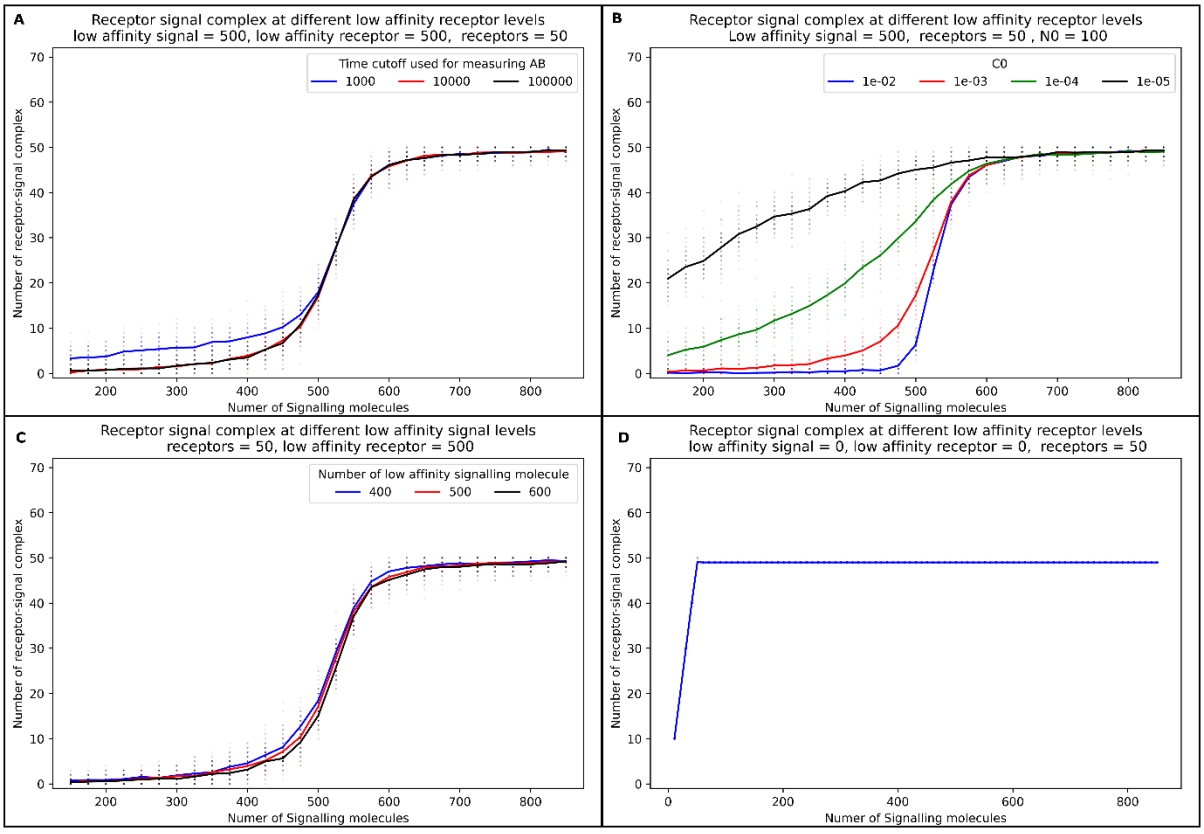


Figure A1: Dependence of receptor activation in varying levels of decoy signalling molecules in panel A, different C_0 values in panel B and different time cutoffs used for sampling in panel C. Panel D shows a control case of herd regulation when there is no low-affinity binder present. The difference in the scale of x-axis was required show show the the results between 0-50. In the range of A, B and C this plot would be a flat line with saturated AB . The parameters used were the same as that in Figure 5.

9.2 Male and female Sxl-Pe activation at different starting level of low affinity repressors

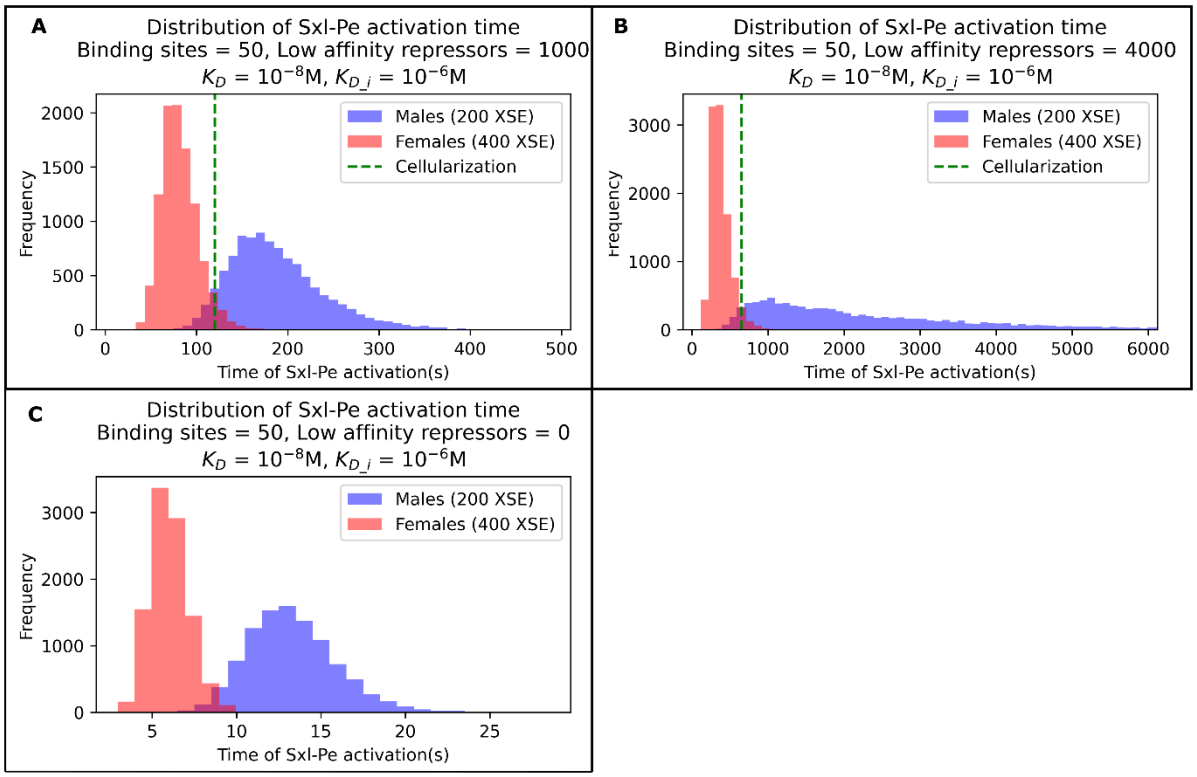


Figure A2: Effect of low affinity repressors on male and female activation times. Panel A shows male and female activation when the starting levels of low affinity repressors are 1000 (half that of figure 5D). Panel B shows the effect of 4000 (double that of figure 5D) low affinity repressors on male and female activation. Please not the variation in scales of x axis.Panel C shows male and female activation when there are no low affinity binders present. Parameters are shown in the top of the plots.

9.3 Local concentrations of reactants.

Here we are addressing the physiological relevance of the concentrations of interactors used in the study. To do this, we estimate the volume that contains a single interactor. This in turn gives us the distance that separates two interactors.

The average volume (m^3) occupied by one interactor is calculated from the concentration as

$$Volume = \frac{N_0}{C_0 \times 6.022e23} \times 10^{-3}$$

Where N_0 is the number of interactors and C_0 is the concentration. 6.022e23 is Avogadro's number and 10^{-3} is the conversion factor from dm^3 to m^3.

Assuming that the volume containing the interactor is spherical

$$Volume = \frac{4}{3}\pi r^3$$

Where *r* is the radius of the sphere. The average distance (*d*) that separates two interactors is 2*r*

$$d = 2 \times \sqrt[3]{\frac{3}{4\pi} \times Volume}$$

In this study, we have considered reactant concentrations ranging from 10^{-5}M to 10^{-3}M. We have also worked with the number of interactors in the range from (of the order of) 100 – 1000. Substituting these values in the equation above gives us interactor distance ranging from 68.2nm to 682 nm. Considering that the sizes of these interactors are of the order of a few nm and also that cellular and sub-cellular compartments are micrometre scale objects (Maul and Deaven, 1977), the concentrations do not represent overly crowded or unphysical scenarios. In fact, in several instances, the concentrations of cellular interactors are shown to be much higher than our somewhat conservative estimates.

Regulatory processes in biological systems usually occur under conditions of local enrichment, e.g. localization in subcellular compartments (Roth et al., 1989), nuclear speckles (Chen and Belmont, 2019) and lipid rafts in the cell membrane (Simons and Ehehalt, 2002). Such local enrichments have been shown to increase effective concentrations by as much as 5 orders of magnitude (Abrami and van Der Goot, 1999). In comparison, the values of concentrations used in this study are conservative.

9.4 Kinetics and steady state of herd regulation can be replicated by differential equations

9.4.1 Only B_i affect the formation of AB

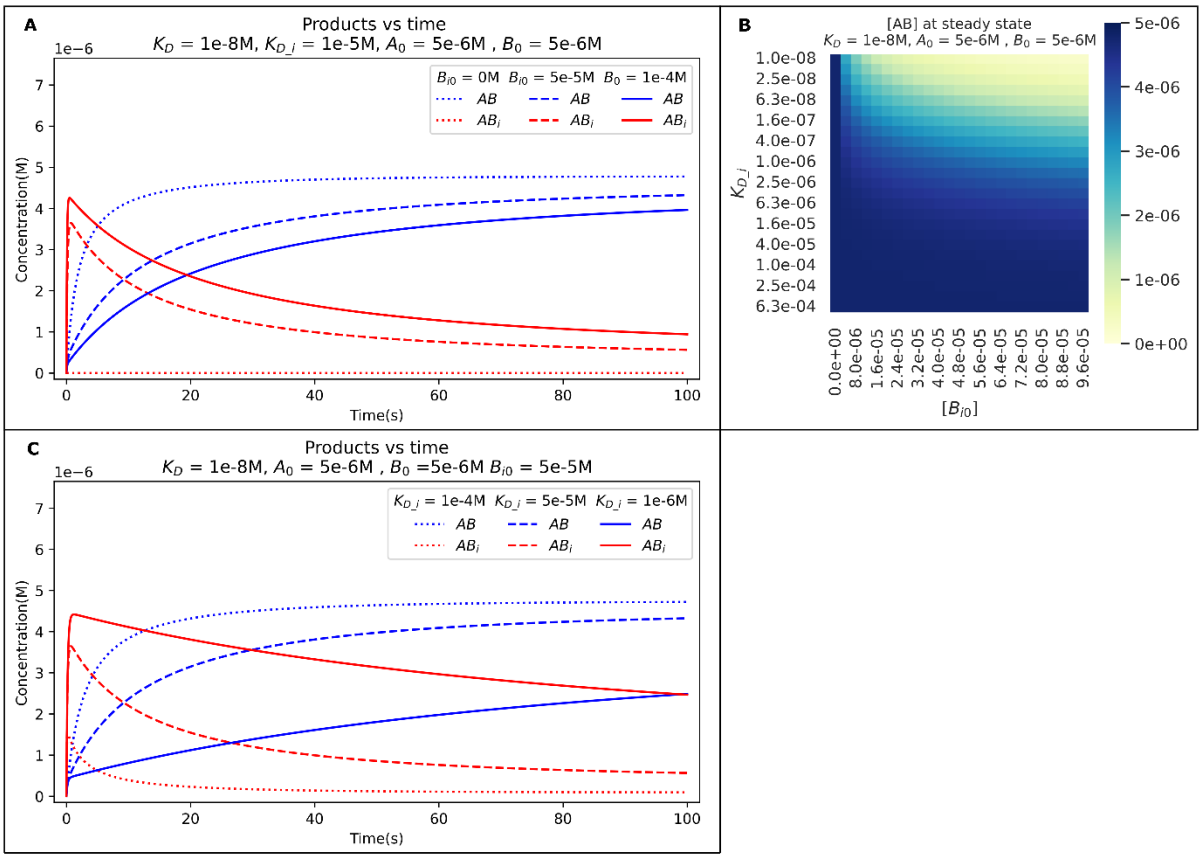


Figure A3: Panel A shows the effect of different concentrations of B_i0 on AB formation. Panel B shows the concentrations of AB at different levels of K_{D_i} and B_i0. Panel C shows the effect of K_{D_i} on AB formation. The concentration scales is noted in the top left corner of the plot

9.4.1 B_i and A_i affect the formation of AB

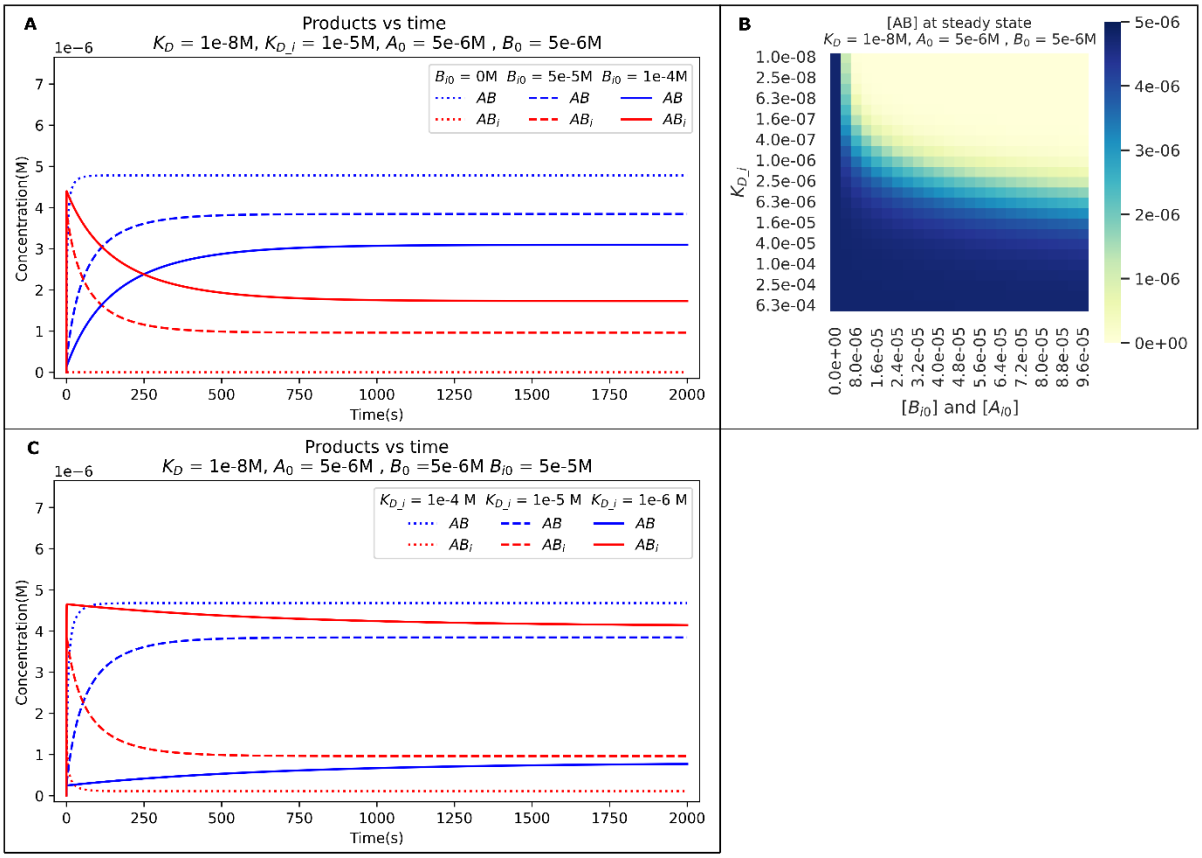


Figure A4: Panel A shows the effect of different concentrations of B_i0 on AB formation. Panel B shows the concentrations of AB at different levels of K_{D_i} and B_i0. Panel C shows the effect of K_{D_i} on AB formation. Parameters used are shown above the graph. The concentration scales is noted in the top left corner of the plot

9.4.3 Effect of cross reaction between B_i and A_i on the formation of AB

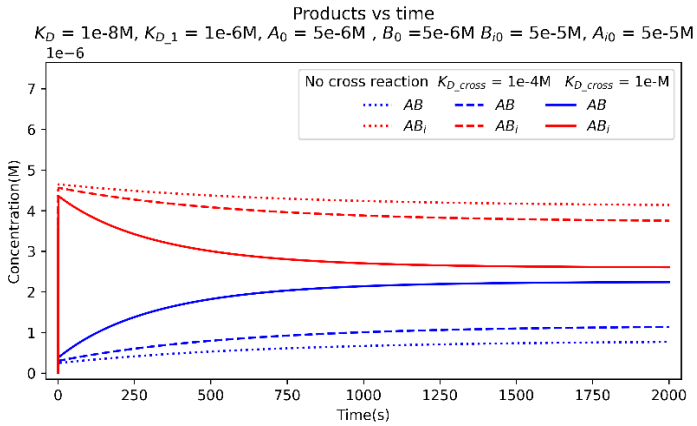


Figure A5: Figure shows the effect of cross reaction on AB formation.

9.5 Effect of herd regulation in higher order when reverser rate constants are fixed

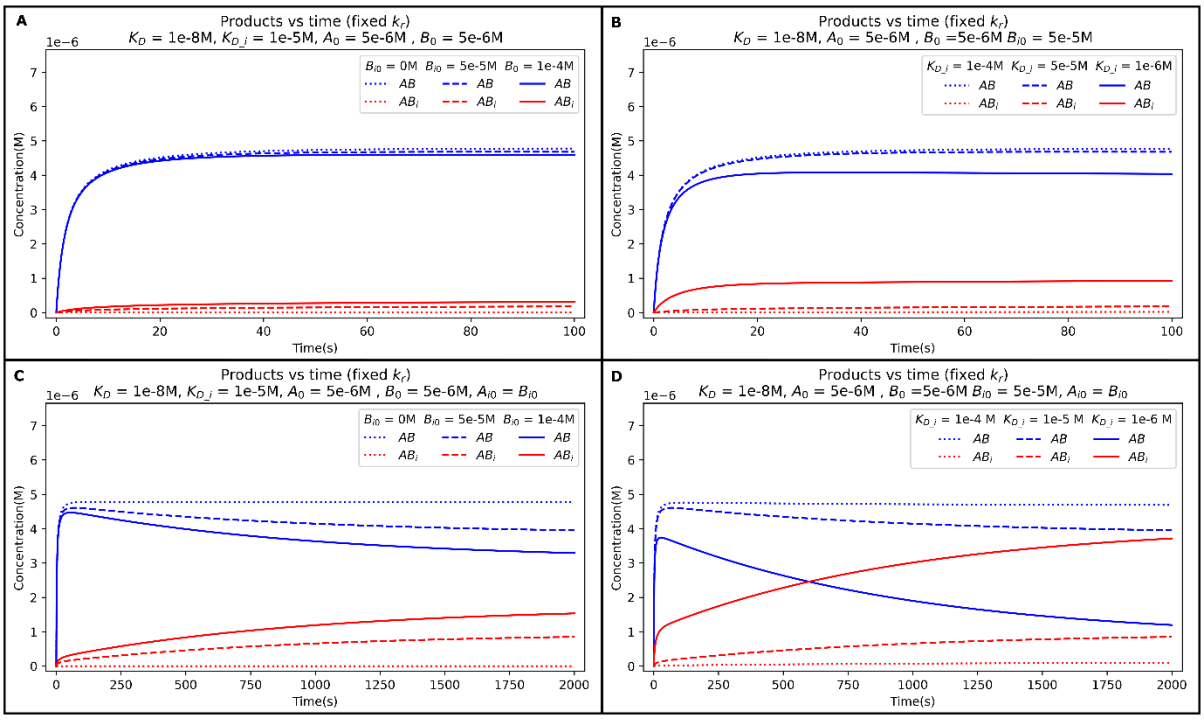


Figure A7: Trajectories with fixed reverse rate constant. The reverser rate constant for all the reactions in this plot are set to 10^{-3} s^{-1}. The equivalent trajectories of reactions show in figure A3 A and C are shown in panels A and B and A4 A and C are shown in panels C and D. The parameters used to plot the graphs are mentioned in the in the respective graphs.

Chapter 6: Conclusions and Future Perspectives

Proteins and their interactions execute most functions in biological systems. This thesis examined three aspects of proteins.

Tree based clustering of local neighbourhoods

We aimed to analyse the structure of proteins by studying their local structural neighbourhoods (Chapter 1). We formulated a new approach to store the structural information about these local neighbourhoods, which clusters and superimposes them based on their structural similarity and, in the process, arranges the clusters in a hierarchical order. This hierarchical order in the form of a tree enables a fast search of such databases. We then used these databases to create a method to search large protein databases in a topology-independent manner to identify partial structural matches. The capabilities of our approach are more significant than the state-of-the-art methods like TM-align and FoldSeek proteins (van Kempen et al. 2024; Zhang and Skolnick 2005) and faster than CLICK, which has similar capabilities (Nguyen and Madhusudhan 2011).

The tree arrangement has a multitude of applications. Nodes in the tree represent similar structural motifs. The similar structural motifs could be interpreted as alternate locations of atoms in a structural motif. These alternate locations can be sampled to infer the dynamics of the clique from a static structure without expensive MD simulations.

Structural motifs have previously been used in various protein design and engineering applications (Jacobs et al. 2016; Kuhlman et al. 2003; Zhou, Panaitiu, and Grigoryan 2020). Recent approaches using attention and graph neural networks resulted in significant headways in accurate protein design (Dauparas et al. 2022; Hayes et al. 2024; Watson et al. 2023). However, these methods train on whole proteins, which might cause biases in the training as structural motifs repeat in different ratios. Our approach clusters local neighbourhoods. The cluster representation forms a non-redundant training dataset for better versions of these neural networks.

The clusters in the tree are formed and represented using the first instance of the structural motif encountered. This implies that the tree depends on the order in which structural information is indexed. Moreover, this also leads to the formation of duplicate nodes if the first motif (the representative) is an outlier, especially in cases with tight cutoffs. To overcome this, the centroid of nodes in the trees can be calculated to form

‘virtual’ representative motifs after eliminating outliers using Mahalanobis distance. The duplicate can be removed after this step by comparing it with child nodes. The membership of the tree can be removed after this step to create a ‘skeleton tree’. A database can now be indexed into this ‘skeleton tree’ instead of creating a new tree from scratch. Moreover, the pattern in which a structure fills the skeleton tree can be used to infer structural information about the structure.

Highly populated nodes with more members should be more thermodynamically stable than sparsely populated nodes, assuming the database was created from a complete structural database. Cliques from models of mutants of proteins can be queried to determine their structural stability. A clique in a protein can also be swapped to a clique from a more stable node to increase the protein's overall thermodynamic stability.

Our program can also be used to identify multiple structure alignments. These alignments can be further processed to identify more extensive ‘domain-like’ features in super-cliques protein structures. The amino acid positions in these super-cliques can be limited in sequence space. The amino acid of one position would also be dependent on other positions. This provides sequence-structure mapping and information about the coevolution of amino acids from topology-independent structural motifs. The analysis of sequence to structures maps, as well as the coevolution of amino acids from multiple sequence alignments by neural networks, is attributed to the success of programs like AlphaFold (Abramson et al. 2024; Jumper et al. 2021). The sequence mappings derived from our structure alignments would provide more extensive and higher-quality data for similar endeavours in the future.

It is important to note that each step in our method is human interpretable as every superimposition performed to arrive at the result can be individually visualised. This contrasts with FoldSeek, which uses a neural network to encode structural motifs as sequences, making interpreting the process near impossible.

The tree data structure as we describe here only requires the cartesian coordinates and a type for indexing and searching. This means that any structural data can be indexed, be it biomolecules like DNA, and RNA; larger coarse-grained assemblies like chromosome coordinates, cyro-EM maps; various chemical entities like drugs and small molecule ligands; non-biological data like stars in a galaxy, buildings in a city, topography, shape of objects in X-ray scans; etc. Moreover, our approach for three-

dimensional coordinates can also be expanded to N dimensions to cluster and index N-dimensional data.

One of the applications of the tree database described in this chapter is to engineer proteins. Please note that the results presented in this part of the thesis were done prior to the development of the tree. Though the tree database would have been handy, the protein engineering described in the next section, was done without it.

Engineering proteins based on local neighbourhoods

In Chapter 2, we use our understanding of analysing local neighbourhoods to add new functionalities to a protein. For this purpose, we selected the function of green fluorescence from Green fluorescent protein. We identified similar structural motifs as the GFP chromophore forming residues. We then checked if the residues (or their mutants) around the structural motifs can recapitulate the interaction and catalytic sites found in the GFP. We selected three proteins to test experimentally from the structures that fall in this category. We used Golden Gate assembly to generate our mutational libraries. To help with this process, we created Overhang Optimizer for Golden Gate Assembly (OOGGA, Chapter 3). Evaluation of two of the three structures resulted in no fluorescence. We are in the process of testing the last proteins.

We could search structural motifs using the tree approach, providing a more robust search method. Our original approach did not use the tree as it was conceptualised after starting the experiments.

As a sperate project, recent advances in computational protein design using neural networks (Dauparas et al. 2022; Watson et al. 2023) can be used to generate de novo designs that are fluorescent.

Effect of low affinity binders in biomolecular interactions

Chapter 4 is distinct from the others as it analyses the interaction between proteins. Here, we speculated that low-affinity binding that is seldom analysed or reported ($KD > 10^{-6}$ M) could affect tight binding biomolecular interaction. We analysed the kinetics and steady states of such systems and found that low-affinity interactions can affect both the kinetics and steady states of biomolecular interaction. We coined the term ‘herd regulation’ for this process. We have then successfully modelled sex determination in Drosophila melanogaster using the effect of herd regulation on

kinetics. We also created the simplest system that forms a binary switch (Mukundan, Deshpande, and Madhusudhan 2024).

We have limited ourselves to the simplest case of herd regulation. Future studies can also explore different/higher order stoichiometries. In addition, many herd-regulated systems can be assembled to create larger functional systems for synthetic biology applications.

In this study, we establish the importance of weak binders in biological systems. Experiments and simulations exploring biological pathways in the future should account for the effect of such weak binders.

References

Abramson, Josh, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. 2024. "Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3." *Nature* 630(8016):493–500. doi: 10.1038/s41586-024-07487-w.

Dauparas, J., I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. 2022. "Robust Deep Learning-Based Protein Sequence Design Using ProteinMPNN." *Science (New York, N.Y.)* 378(6615):49–56. doi: 10.1126/SCIENCE.ADD2187.

Hayes, Thomas, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil

Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. 2024. “Simulating 500 Million Years of Evolution with a Language Model.” *BioRxiv* 2024.07.01.600583. doi: 10.1101/2024.07.01.600583.

Jacobs, T. M., B. Williams, T. Williams, X. Xu, A. Eletsky, J. F. Federizon, T. Szyperski, and B. Kuhlman. 2016. “Design of Structurally Distinct Proteins Using Strategies Inspired by Evolution.” *Science* 352(6286):687–90. doi: 10.1126/science.aad8036.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. 2021. “Highly Accurate Protein Structure Prediction with AlphaFold.” *Nature* 596(7873):583–89. doi: 10.1038/s41586-021-03819-2.

van Kempen, Michel, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. 2024. “Fast and Accurate Protein Structure Search with Foldseek.” *Nature Biotechnology* 42(2):243–46. doi: 10.1038/s41587-023-01773-0.

Kuhlman, Brian, Gautam Dantas, Gregory C. Ireton, Gabriele Varani, Barry L. Stoddard, and David Baker. 2003. “Atomic-Level Accuracy.” *Science* 302(November):1364–68.

Mukundan, S., Girish Deshpande, and M. S. Madhusudhan. 2024. “High-Affinity Biomolecular Interactions Are Modulated by Low-Affinity Binders.” *Npj Systems Biology and Applications* 10(1):85. doi: 10.1038/s41540-024-00410-z.

Nguyen, Minh N., and M. S. Madhusudhan. 2011. “Biological Insights from Topology Independent Comparison of Protein 3D Structures.” *Nucleic Acids Research*

39(14). doi: 10.1093/nar/gkr348.

Watson, Joseph L., David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. 2023. “De Novo Design of Protein Structure and Function with RFdiffusion.” *Nature* 620(7976):1089–1100. doi: 10.1038/S41586-023-06415-8.

Zhang, Yang, and Jeffrey Skolnick. 2005. “TM-Align: A Protein Structure Alignment Algorithm Based on the TM-Score.” *Nucleic Acids Research* 33(7):2302–9. doi: 10.1093/nar/gki524.

Zhou, Jianfu, Alexandra E. Panaitiu, and Gevorg Grigoryan. 2020. “A General-Purpose Protein Design Framework Based on Mining Sequence–Structure Relationships in Known Protein Structures.” *Proceedings of the National Academy of Sciences* 117(2):1059–68. doi: 10.1073/pnas.1908723117.

Appendix

1. GFP PDB IDs

1B9C,1C4F,1EMA,1EMB,1EMC,1EME,1EMG,1EML,1EMM,1HCJ,1HUY,1JBY,1JBZ,1JC0,1JC1,1KP5,1Q4A,1Q4B,1Q4C,1Q4D,1Q4E,1Q73,1QYF,1RMM,1RMO,1RMP,1W7S,1W7T,1W7U,1YFP,2AH8,2AHA,2AWK,2AWL,2AWM,2B3P,2B3Q,2DUI,2GX2,2H5O,2H5P,2H5R,2H6V,2H9W,2HJO,2HPW,2HQZ,2HRS,2O24,2O29,2O2B,2OKW,2OKY,2Q6P,2QLE,2QU1,2QZ0,2RH7,2WUR,2Y0G,2ZO6,2ZO7,3CB9,3CBE,3CD1,3CD9,3ED8,3EK4,3EK7,3EK8,3EKH,3EKJ,3EVP,3EVR,3EVU,3EVV,3G9A,3GJ1,3GJ2,3K1K,3LVA,3LVC,3LVD,3OGO,3P28,3SG2,3SG3,3SG4,3SG5,3SG6,3SG7,3U8P,3UFZ,3UG0,3WLC,3WLD,4EUL,4HE4,4IK1,4IK3,4IK4,4IK5,4IK8,4IK9,4J8A,4KA9,4KAG,4KEX,4KF5,4KW4,4KW8,4KW9,4L12,4L13,4L1I,4LQT,4LQU,4LW5,4N3D,4OGS,4ORN,4P1Q,4P7H,4PA0,4PFE,4W69,4W6A,4W6B,4W6C,4W6D,4W6F,4W6G,4W6H,4W6I,4W6J,4W6K,4W6L,4W6M,4W6N,4W6O,4W6P,4W6R,4W6S,4W6T,4W6U,4W72,4W73,4W74,4W75,4W76,4W77,4W7A,4W7C,4W7D,4W7E,4W7F,4W7R,4W7X,4XBI,4XGY,4XL5,4XOV,4XOW,4XVP,4ZF3,5AQB,5BT0,5BTT,5DPG,5DPH,5DPI,5DPJ,5DY6,5EHU,5F9G,5HZO,5HZS,5HZT,5HZU,5J3N,5JZK,5JZL,5LEL,5LEM,5MA3,5MA4,5MA5,5MA6,5MA8,5MA9,5MAD,5MAK,5MFC,5MSE,5N9O,5NHN,5NI3,5WTS,6B7R,6B7T,6DGV,6DQ0,6DQ1,6EFR,6FLL,6FWW,6HUT,6IR6,6IR7,6ITC,6JGH,6JGI,6JGJ,6KKZ,6KL0,6KL1,6KRG,6L26,6L27,6LNP,6LR7,6M63,6MB2,6MZ3,6NHV,6OFK,6OFN,6OFO,6PER,6QUH,6QUI,6QUJ,6S67,6UHJ,6UHK,6UHL,6UHM,6UHN,6UHO,6UHP,6UHQ,6UHR,6URU,6WV3,6WV4,6WV5,6WV6,6WV7,6WV8,6WV9,6WVA,6WVB,6WVD,6WVE,6WVF,6WVG,6WVH,6WVI,6XU4,6XZF,6YLP,6YLQ,7A7K,7A7L,7A7M,7A7N,7A7O,7A7P,7A7Q,7A7R,7A7S,7A7T,7A7U,7A7V,7A7W,7A7X,7A7Y,7A7Z,7A80,7A81,7A82,7A83,7A84,7A85,7A86,7A87,7A88,7A89,7A8A,7A8B,7A8C,7A8D,7A8E,7A8F,7A8G,7A8H,7A8I,7A8J,7A8K,7A8L,7A8M,7A8N,7A8O,7C03,7CD7,7CD8,7E53,7E9Y,7KM4,7LG4,7PCA,7PCZ,7S7T,7S7U

2. GFP PDB IDs (resolution < 1.5 Å)

1JBZ,1Q4A,1Q4B,1Q4E,1QYF,2AWK,2B3P,2DUI,2H5O,2H5P,2H6V,2HJO,2HQZ,2HRS,2O24,2QZ0,2RH7,2WUR,2Y0G,2ZO6,3CB9,3CBE,3CD1,3CD9,3EVP,3LVA,3LVC,4EUL,4J8A,4KAG,4L1I,4LQT,4N3D,4P1Q,4XGY,4XOV,4XOW,5AQB,5DPH,5EHU,5MA4,5NI3,6FWW,6HUT,6IR7,6JGH,6JGI,6JGJ,6KKZ,6KL0,6KL1,6KRG,6L26,6L27,6MZ3,6OFK,6QUH,6UHJ,7A7L,7A7N,7A7P,7A7W,7A7Y,7A7Z,7PCA,7PCZ

3. Donor and acceptor list of finding hydrogen bonds

```
Format <Residue name> <Main chain or side chain> <Donor or
acceptor> <Atom id> <Antecedent atom id>

ALA MC DONOR    N   H
ALA MC ACCEPTOR O   C

ARG MC DONOR    N   H
ARG SC DONOR    NE  HE
ARG SC DONOR    NH1 HH1
ARG SC DONOR    NH2 HH2
ARG MC ACCEPTOR O   C

ASN MC DONOR    N   H
ASN SC DONOR    ND2 HD2
ASN SC ACCEPTOR OD1 CG
ASN MC ACCEPTOR O   C

ASX MC DONOR    N   H
ASX SC DONOR    ND2 HD2
ASX SC ACCEPTOR OD1 CG
ASX MC ACCEPTOR O   C

ASP MC DONOR    N   H
ASP SC ACCEPTOR OD1 CG
ASP SC ACCEPTOR OD2 CG
ASP MC ACCEPTOR O   C

CYS MC DONOR    N   H
CYS SC DONOR    SG  HG
CYS MC ACCEPTOR O   C

CSS MC DONOR    N   H
CSS MC ACCEPTOR O   C

GLN MC DONOR    N   H
GLN SC DONOR    NE2 HE2
GLN SC ACCEPTOR OE1 CD
GLN MC ACCEPTOR O   C

GLX MC DONOR    N   H
GLX SC DONOR    NE2 HE2
GLX SC ACCEPTOR OE1 CD
GLX MC ACCEPTOR O   C

GLU MC DONOR    N   H
GLU SC ACCEPTOR OE1 CD
GLU SC ACCEPTOR OE2 CD
GLU MC ACCEPTOR O   C
```

```
GLY MC DONOR    N   H
GLY MC ACCEPTOR O   C

UNK MC DONOR    N   H
UNK MC ACCEPTOR O   C

HIS MC DONOR    N   H
HIS SC DONOR    ND1 HD1
HIS SC ACCEPTOR NE2 CD2
HIS MC ACCEPTOR O   C

HSE MC DONOR    N   H
HSE SC DONOR    NE2 HE2
HSE SC ACCEPTOR ND1 CG
HSE MC ACCEPTOR O   C

HSP MC DONOR    N   H
HSP SC DONOR    ND1 HD1
HSP SC DONOR    NE2 HD2
HSP MC ACCEPTOR O   C

ILE MC DONOR    N   H
ILE MC ACCEPTOR O   C

LEU MC DONOR    N   H
LEU MC ACCEPTOR O   C

LYS MC DONOR    N   H
LYS SC DONOR    NZ  HZ
LYS MC ACCEPTOR O   C

MET MC DONOR    N   H
MET MC ACCEPTOR O   C

PHE MC DONOR    N   H
PHE MC ACCEPTOR O   C

PRO MC ACCEPTOR O   C

SER MC DONOR    N   H
SER SC DONOR    OG  HG
SER SC ACCEPTOR OG  CB
SER MC ACCEPTOR O   C

THR MC DONOR    N   H
THR SC DONOR    OG1 HG1
THR SC ACCEPTOR OG1 CB
THR MC ACCEPTOR O   C
```

```
TRP MC DONOR    N   H
TRP SC DONOR    NE1 HE1
TRP MC ACCEPTOR O   C

TYR MC DONOR    N   H
TYR SC DONOR    OH  HH
TYR SC ACCEPTOR OH  CZ
TYR MC ACCEPTOR O   C

VAL MC DONOR    N   H
VAL MC ACCEPTOR O   C

CRO MC DONOR    OG1 HG1
CRO SC ACCEPTOR OG1 CB1
CRO SC ACCEPTOR N2  C1
GYS MC ACCEPTOR N3  C2
CRO SC ACCEPTOR O3  C3
CRO SC ACCEPTOR O2  C2
CRO SC DONOR    OH  HH
CRO SC ACCEPTOR OH  CZ
CRO MC ACCEPTOR O   C

GYS MC DONOR    OG1 HG1
GYS SC ACCEPTOR OG1 CB1
GYS SC ACCEPTOR N2  C1
GYS MC ACCEPTOR N3  C2
GYS SC ACCEPTOR O3  C3
GYS SC ACCEPTOR O2  C2
GYS SC DONOR    OH  HH
GYS SC ACCEPTOR OH  CZ
GYS SC ACCEPTOR OH  CZ
GYS MC ACCEPTOR O   C

HOH MC ACCEPTOR O   O
HOH MC DONOR    O   O

DOD MC ACCEPTOR O   O
DOD MC DONOR    O   O
```

4. Description of rings of residues

```
Format <Res_name> <space separated list of atoms of a ring in
order of preference for calculating plane>

TYR CG CE1 CE2 CD1 CD2 CZ

TRP CG NE1 CD1 CD2 CE2
TRP CD2 CZ2 CZ3 CE3 CE2 CH2

HIS CG NE2 CE1 CD2 ND1
```

```
HSE CG NE2 CE1 CD2 ND1

HSP CG NE2 CE1 CD2 ND1

PHE CG CE1 CE2 CD1 CD2 CZ

CRO CA3 CA2 N2 C1 C2
CRO CB2 CE1 CE2 CD1 CD2 CZ

GYS CA3 CA2 N2 C1 C2
GYS CB2 CE1 CE2 CD1 CD2 CZ
```

5. Description of charges of residues

```
Format <Res_name> <anion or cation> <atom_name> <space separated
atoms_in_resonance>
atoms_in_resonance are atoms across which charge is delocalised
and space separated

ARG CAT NE NH1 NH2
ARG CAT NH1 NH2 NE
ARG CAT NH2 NH1 NE

ASP ANI OD1 OD2
ASP ANI OD2 OD1

GLU ANI OE1 OE2
GLU ANI OE2 OE1

HIS CAT ND1 NE2
HIS CAT NE2 ND1

HSE CAT ND1 NE2

HSP CAT ND1 NE2
HSP CAT NE2 ND1

HSD CAT NE2 ND1

LYS CAT NZ
```

6. List of PIP binding proteins

Residue name	No of structures	PDB ids

IPD	12	1IMA,7JS7,7JS5,2ORK,6LFJ,1AWB,1LBX,4RW3,5J16,1G0H,5F24,3IKP
4PT	5	7YIS,6FJC,1W1G,2Z0P,3W68
XJ7	13	7MZ9,7L2H,7L2U,7MZA,7MZE,7L2S,7L2J,7MZB,7L2R,7L2P,8FXG,7MZ6,7L2T
PIO	87	8T1O,7B2L,8ED8,2H3V,5L0H,7OVR,6W7E,8E4L,8IAB,8T47,6CDS,7TU9,8DDU,6UP7,8CRR,8CRQ,6XIT,8J02,8JRU,5L0G,8W4U,1HFA,6I53,7V07,7V19,6DMU,7SKU,4KFM,6XEU,8CTE,5LO8,8IAD,6VM0,5L0C,8U2Z,8I5M,8E4Q,3SPH,8T3R,8DDS,5ON7,8T44,3SYA,6M84,7T6M,6CS9,7T6Q,6HUG,4CQK,5L0D,7XNL,8CT3,8T45,8J01,6HUP,8E4M,3SPI,6HUO,8T3U,6PW4,6HUK,8E4O,6XEV,8I5N,3SPG,8FFO,6UBS,8JRV,6VM2,4PR9,8ED9,7QNE,8U30,5VYP,8DDT,8DDV,3SYQ,6MFS,6HUJ,6PW5,6E7Y,7XNN,4NS0,7UZ3,8E4N,5KUM,8DDX
2Y5	9	8T3O,7OH6,4PH7,7OH7,7OH4,7OH5,6ROJ,6ROI,7PEM
I3P	40	6CV8,4WU2,5J68,8T8Y,3T8S,4NP9,4O4D,3C5N,8QBB,5W2H,5XA1,1N4K,5J67,2P0D,7Z3J,3UJ0,1W2C,8EAR,1DJX,7F1X,5HJQ,7JXA,3V0H,5W2I,4QJ4,8QBF,1GC6,4QJ5,7T3U,5GUG,1OQN,1NU2,1H0A,8QBD,1BTN,1MAI,3W9F,2A98,5X1O,1U29
IBS	1	1I7E
EUJ	9	7M5V,7SQ7,7M5Y,6NQ0,6NQ2,7SQ9,7M5X,8PVR,6C9A
5P5	1	3RGQ
PT5	27	8X91,8WE6,7X1J,6V01,8TI2,8EPL,7BYN,7VFW,8ZLE,7VFV,7MIY,7VFS,8JNI,5EGI,3GPE,8X90,7X1I,7BYL,7MIX,8TI1,7BYM,7VFU,8X93,8WE9,8WE8,7VNP,5EIK
T7X	21	7A24,7ZKP,8CMO,6Y79,7BGI,7AQQ,6RFQ,8K66,7P1K,8K69,7ARB,6RFR,7O71,7A23,7BLZ,7LP9,7LPC,5HYM,7AR8,7AR7,7O6Y
PIB	6	7BLQ,1OCU,1H6H,2RAK,1ZSQ,7BLO
B7N	6	3B7N,4J7Q,6SLD,3B7Z,7WVT,7WWG
PII	2	3QI9,1GZQ
DB4	1	4MXP
52N	1	4CML
J40	9	8FYT,7E2Y,8FYE,8FYL,8FY8,7E2Z,8FYX,7E2X,8JT6
IP9	6	7A17,3MDB,8OX9,3LJU,8OX8,8OX7
PIK	3	4QK4,6PW4,6PW5
4IP	29	3O3L,1FHX,1U27,1BWN,2UZS,4WTY,1W1D,4WU3,2R0D,6U3G,6BBP,2LKO,1B55,6BBQ,7KJZ,1H10,1FAO,5D3Y

		,6U3E,1W2D,2R09,1UNQ,5D3X,8TUA,1UPR,1FGY,7SDD,4KAX,3AJM
ITP	3	4AVX,1JOC,1HYI
I3S	2	2P0H,1Z2P
T7M	3	8PJK,3SPW,8PKM
PIE	10	2XSV,2XSR,8DV4,6GYO,8DV3,1KB9,1UW5,8OXC,2XSU,6LUM
2IP	2	1I9Z,7KIR
YBG	1	7LQY
85R	2	8T3M,7X2U
0J1	2	7JM7,7JM6
LIP	4	4XF6,3C4V,1IMB,6WMV
LPY	1	4XPJ
PIF	3	4INQ,7DEI,3MTC
3PT	2	4FYG,3W67
PIZ	2	4QJR,4RWV
PBU	7	4OVV,2K4I,5C79,2H3Q,6FJD,2H3Z,3W68
8IJ	6	8T10,8T0E,8T0Y,8T3L,8GF8,8GF9
WES	1	7KHT
IEP	3	6W8C,5ZM8,5ZM6
KXP	2	6NR2,6NR3
HZ7	2	6E7P,6E7Z
3PI	1	1ZVR
8IO	1	7VKT
6ES	1	5IRZ
EIJ	2	7SHE,7SHF
KYG	1	6NR7
I35	1	6KOJ
P3H	1	6SL5
9Y5	1	5OMH
PWE	1	6WHG