USING EVOLUTIONARILY BASED CORRELATION MEASURES TO IMPROVE BCL::FOLD PROTEIN STRUCTURE PREDICTION

By

Pedro Luis Teixeira, Jr.

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Biomedical Informatics

August, 2014

Nashville, Tennessee

Approved:

Jens Meiler Ph.D.

Thomas A. Lasko M.D., Ph.D.

Terry P. Lybrand Ph.D.

ACKNOWLEDGEMENTS

First, I would like to thank my committee for their guidance and support. Their help has been indispensable throughout this process. Professor Meiler and I have had many stimulating scientific discussions and I am grateful for the opportunity I have had to work in his lab. Professor Lybrand has been a wonderful committee member - always providing insightful comments and suggestions. Dr. Lasko, I have greatly appreciated your perspective and always had such wonderful meetings with you to discuss my progress. Our discussions have always stoked my scientific curiosity.

Dr. Dermody, you have been an inspiration and a continual source of helpful advice and motivation. Regardless of circumstance, I always feel enthusiastic after every conversation we have. Professor Gadd, thank you so much for everything you have done for me. I have always enjoyed our chats and appreciate you always making time to for me.

I would also like to thank Jeff Mendenhall, Sten Heinze, and Brian Weiner for the many great scientific discussions we have had, as well as for their programming help. I have learned so much from you and my computer science knowledge is so much greater now thanks to our interactions and time coding together.

The Departments of Biomedical Informatics has always made me feel at home, for which I am very grateful. My student colleagues have been wonderful sources of help and encouragement. The faculty and staff have also been tremendously supportive. I would especially like to thank Rischelle Jenkins for her endless helpfulness.

My funding through the Vanderbilt MSTP T32 (GM07347 NIH/NIGMS) and through NIH R01 Protein-Ligand Grant (NIGMS 1R01GM099842-01) and Cheminformatics R01 Grant NIH (NIMH 1R01MH090192-01) have made this work possible.

I am deeply appreciative of my family who has always supported me. It means so much to me to know that are you always there.

Finally, I would like to thank my wonderful wife, Carolyn. She has been understanding, supportive, and helpful beyond words. Her presence has always kept me in good spirits regardless of scientific obstacles. I could not have done this without her.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

ANN ..... artificial neural network

AUC.....area under the (receiver operating characteristic) curve

BCL .....Biochemical Library

Cα ..... α-carbon on the backbone of the amino acid

Cβ ..... β-carbon on the backbone of the amino acid

CPU..... central processing unit

DI .....direct information

DT ..... decision tree

KBP ..... knowledge based potential

MC.....Monte Carlo

MI ..... mutual information

ML ..... machine learning

MSA..... multiple sequence alignment

NMR .....nuclear magnetic resonance

PDB..... protein data bank

PPV.....positive predictive value

PvFPP.....precision vs. fraction predicted positive

PSP ..... protein structure prediction

RMSD.....root mean square distance

ROC.....receiver operating characteristic

SFFS ..... sequential forward feature selection

SSE..... secondary structure element

USING EVOLUTIONARILY BASED CORRELATION MEASURES TO IMPROVE BCL::FOLD PROTEIN STRUCTURE PREDICTION

Introduction

Overview

The objectives of this thesis are (1) to use site-site correlation data in addition to primary sequence properties to predict more accurately long-range protein contacts; (2) to use predicted long-range contacts to enrich for native-like models during *de novo* prediction using BCL::Fold; (3) to explore modifications to our method that may further enhance *de novo* prediction confidence based scoring and sampling subsets of predicted contacts.

Protein Structure Prediction

One of the primary purposes of DNA is to store information regarding the sequence of amino acids in each of the proteins that carry out the necessary functions to maintain and perpetuate life. RNA transfers this sequence information from DNA to sequences of amino acid via the machinery of the ribosome. The organization of amino acids in sequence is known as the primary structure. During folding the sequence forms simple local structures, secondary structural elements (SSEs). SSEs then aggregate into the final fold, also known as the tertiary structure. Thus the sequence of amino acids dictates the tertiary, or three-dimensional structure, of each protein with the caveat that some proteins, especially ones of larger size, require chaperone molecules to successfully surmount intermediate states and arrive at their final thermodynamically stable

conformation (Anfinsen, 1973). A folded protein can then carry out its function – enzymatic, structural, signaling, or otherwise (Baker, 2000; Floudas, Fung, McAllister, Mönnigmann, & Rajgaria, 2006; Levinthal, 1968; Schwede, Sali, & Eswar, n.d.; Zwanzig, Szabo, & Bagchi, 1992).

Thus, protein structure dictates function, and knowledge of a protein's fold informs our understanding of normal activity and dysfunction. However, elucidating the structure for non-trivial proteins *de novo* (using sequence information alone) is a challenging problem due to the sheer size of the fold search space. This search space increases exponentially as amino acid sequence length increases. Thus, *de novo* structure prediction remains especially difficult for proteins larger than 150 amino acids (Bonneau, Strauss, et al., 2002; Yarov-Yarovoy & Schonbrun, 2006). Myriad different methods have addressed this challenge computationally, including comparative modeling, which uses the determined structures of similar sequences (Kopp & Schwede, 2004); fragment-based methods, which search for similar portions of a protein in other sequences with known structures and assemble the matching fragments (Rohl, Strauss, Chivian, & Baker, 2004); fold recognition methods, which leverage the fact that entire protein folds are less diverse than sequence structure suggests, and often very different sequences can fold to the same 3-dimensional structure. Once an approximate placement is determined using either the combinations of sub-elements (fragment-based methods) or entire sequences threaded into similar related known structures (homology modeling), other methods are used to score and refine the predicted structures (Floudas et al., 2006; D. Kim, Xu, Guo, Ellrott, & Xu, 2003; Przybylski & Rost, 2004).

As computational power has continued to increase, simulating the protein folding pathway at varying levels of complexity has been attempted with more resource intensive *de novo*

methods. Such approaches predict the final fold from the primary amino acid sequence without structural information from related proteins. These include molecular dynamics methods (Alder & Wainwright, 1959; Ding, Tsao, Nie, & Dokholyan, 2008; Proctor, Ding, & Dokholyan, 2011; Shaw, Deneroff, & Dror, 2008) and BCL::Fold which simplifies the complexity of protein models by generating models with a Monte Carlo Metropolis simulated annealing algorithm that samples the placement of idealized secondary structural elements and scores each model using several knowledge-based scoring potentials (Karakaş et al., 2012; Woetzel et al., 2012).

One can increase protein structure prediction sampling efficiency by constraining the search space with long-range contact restraints (positions distant in the primary sequence but known to be in close proximity within the tertiary structure). Given a sufficient number of accurate contact restraints, it is possible to solve structures deterministically using distance geometry. BCL::Fold can leverage sparse experimental and computational restraints to improve accuracy. Such restraints include NMR chemical shifts, residual dipolar couplings, medium-resolution cryo-electron microscopy data, electron paramagnetic resonance, small-angle X-ray scattering, and distance restraints (Sanders & Sönnichsen, 2006; B. E. Weiner, Woetzel, Karakas, Alexander, & Meiler, 2013).

For all *de novo* protein structure prediction methods, the greatest challenge is effectively sampling the conformational search space. No method can recognize a native-like topology that it does not sample. Thus, constraints that effectively limit the search space, such as contact restraints derived from evolutionary site-site couplings, are invaluable for improving current methods. An efficiently constrained search space reduces the effects of the sampling bottleneck (D. E. Kim, Blum, Bradley, & Baker, 2009).

Contact Prediction

De novo structure prediction begins with only the primary sequence. The initial search set is the entirety of conformational space for a given sequence. Constraints reduce the computational complexity of identifying the native fold. Knowing that two amino acid sites that are distant in the sequence are close in space provides useful information. In other words, the "information content" is directly proportional to the predicted contact's sequence separation (Alexander, Bortolus, Al-Mestarihi, Mchaourab, & Meiler, 2008). The frequency of these long-range contacts within a given protein varies, and one can compare proteins' contact order to understand the relative complexity of the fold. Specifically, contact order is calculated as the average sequence separation between amino acid pairs that form contacts in the folded 3-dimensional structure (Bonneau, Ruczinski, Tsai, & Baker, 2002). As long-range contacts form during actual or simulated protein folding, they are more likely to be disrupted by the long intervening amino acid chain. Thus, protein structures with higher contact order fold more slowly and are more challenging to predict for methods like Rosetta, MD-based methods, and many other methods which maintain a continuous primary structure (Lindorff-Larsen, Piana, Dror, & Shaw, 2011; Punta & Rost, 2005). This is in contrast to the BCL, which focuses on placing and scoring separate SSEs, enabling it to better leverage long-range contact restraints than other methods.

While other definitions have been used, a contact, as per the Critical Assessment of Protein Structure Prediction, is an amino acid pair with their Cβ carbons within 8Å or less (Cα is used in the case of glycine) and with a minimum separation of six amino acids in the primary sequence (Ezkurdia, Graña, Izarzugaza, & Tress, 2009; Izarzugaza, Gran, Tress, Valencia, & Clarke,

2007). A long-range contact is traditionally separated by 12 amino acids along the primary sequence. In addition to improving protein fold prediction, contact prediction has also proved useful for estimating the rate of protein folding (Punta & Rost, 2005).

Most available contact prediction methods achieve accuracies insufficient for robust *de novo* protein folding. Earlier methods developed to predict long-range contacts have accuracies that peak at around 20%. Newer methods, utilizing deep architectures, only improve accuracy to 30% (Di Lena, Nagata, & Baldi, 2012). Current contact prediction methods include the use of statistical methods, genetic algorithms (Chen & Li, 2010), machine learning methods (ANN, Deep Architectures) (Fariselli, Olmea, Valencia, & Casadio, 2001), sequence conservation (Jones, Buchan, Cozzetto, & Pontil, 2012), predicted secondary structures (Karakaş, Woetzel, & Meiler, 2010), mutual information, and more recently direct information (DI) (Olmea & Valencia, 1997; Shackelford & Karplus, 2007).

Previous work leveraged many different data points to predict contacts. These data types include primary sequence position, site separation, site-site correlation such as mutual information, predicted secondary structure, predicted solvent accessible surface area, amino acid property profiles, position-specific scoring matrices PSIBLAST), as well as predicted solvent accessibility (Altschul et al., 1997; Di Lena et al., 2012; Karakaş et al., 2010). Earlier work also utilized complexity composition of the intervening region (Xue, Faraggi, & Zhou, 2009). However, using these descriptors alone only yields maximum accuracies of up to 42% in some cases (Karakaş et al., 2010). This level of performance has been beneficial but further increases in accuracy can further bolster *de novo* prediction.

Direct Information (DI), has been particularly promising. Morcos *et al.* have used DI to

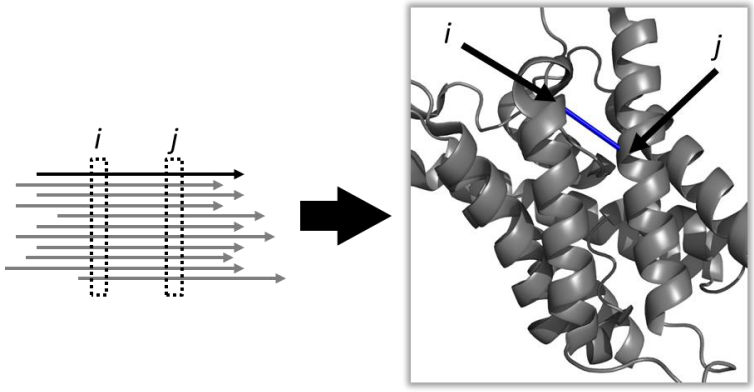


Figure 1: Depiction of Correlation Between Positions in a MSA and the Related Physically Proximal Site-Site Pair

In a MSA (panel A), a correlated pattern of mutations at *sites i and j* is indicative of the two amino acids having some mutually shared evolutionary pressure. This is often indicative of both sites being near one another in the final folded structure.

determine the structures of some membrane and soluble proteins with enough homologous sequences to form a deep and accurate alignment. However, additional filtering and secondary structure-based constraints are necessary to attain the published performance (Morcos et al., 2011).

Mutual Information

Mutual information (MI) is one algorithm for determining pairwise correlation. Pairwise correlation, denoted here as C_{ij}^{ab}, captures the statistical correlation of two amino acid sites *i* and *j* based on mutations observed at each site respectively within a multiple sequence alignment

(MSA). This correlation often captures pairs correlated due to an important role in maintaining local structure, or clusters of correlated sites near functionally essential regions, such as enzymatic active sites. Consequently, sites determined to have higher MI values are enriched for "contacts", put simply, amino acid pairs in closer proximity to one another than random selections (Gloor, Martin, Wahl, & Dunn, 2005). As described in Equation 1., C_{ij}^{ab} is the deviation of the pairwise frequency $f_{ij}^{(ab)}$ of amino acids *a* and *b* at positions *i* and *j* respectively, from the expected pairwise frequency based on the individual frequencies f_i^a and f_j^b of *a* at *i* and b at *j* (Gloor *et al*. 2005; Shackelford and Karplus 2007).

$$MI(i,j) = \sum_{a=1}^{20} \sum_{b=1}^{20} f_{ij}^{(\mathrm{ab})} \log_{20}\left(\frac{f_{ij}^{(\mathrm{ab})}}{f_i^a f_j^b}\right) \qquad (1.)$$

Equation 1: Equation Used to Determine the Mutual Information at Site *i,j*

MI is calculated by summing the observed frequency of amino acid types *a* and *b* multiplied by the log of the observed frequency of each pair *a*, *b* divided by the expected pair probability given the observed frequencies for *a* and *b* individually at the columns i *and j,* respectively.

Direct Information

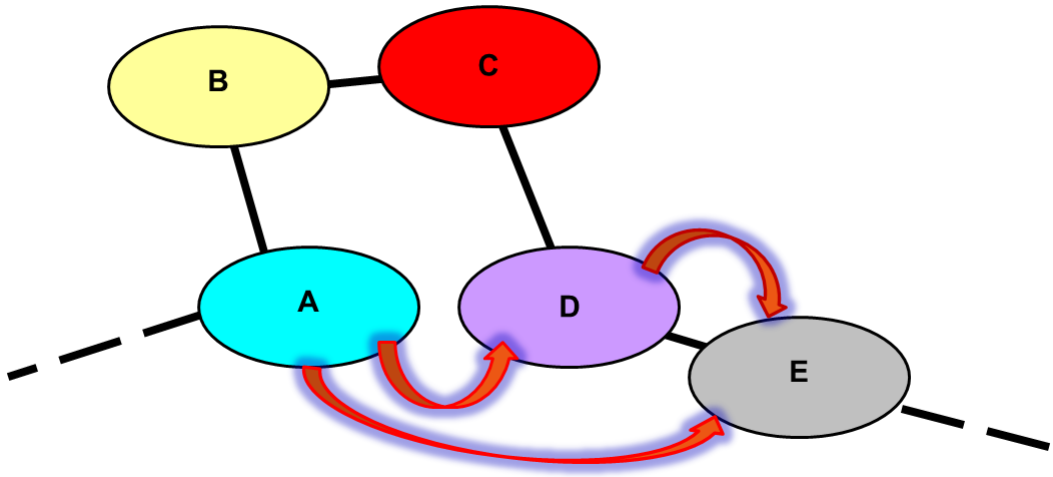


Figure 2: Schematic of Direct information and Transitive Correlations

DI isolates direct correlation, the detected correlation between sites "A" and "D" as well as "D" and "E", while filtering out false transitive correlations as symbolized above with the connection between sites "A" and "E". DI accomplishes this goal via an entropy maximization step, which attempts to explain all couplings seen with a minimal set of correlated sites.

Correlations can be direct or transitive. A network of interacting sites can couple two distant sites, resulting in a correlation between sites that are distant in the final folded structure. As depicted in Figure 2, correlations between site "A" and "D" in addition to "D" and "E" may also lead to a significant detected correlation between sites "A" and "E". This is commonly true for the MI algorithm and consequently decreases the accuracy of contact prediction methods that rely on MI. DI's entropy maximization step searches for the minimal set of correlations that explain all statistical couplings seen from the internal MI calculation. In so doing, it filters out the indirect or transitive correlations such as the correlation depicted below between site "A" and "E" (Marks, Colwell, Sheridan, Hopf, Pagnani, & Sander, 2011).

Recent work using DI has achieved positive predictive values (the proportion of contact predicted to be in contact which actually are in contact) of up to 0.8 for the top-ranked predictions of some proteins by including a global correlation approach (Marks, Colwell, Sheridan, Hopf, Pagnani, Sander, et al., 2011). Evolution selects for new mutations at sites adjacent in the folded structure that can compensate for prior deleterious mutations. Given a sufficient set of related but evolutionarily diverse protein sequences, it is possible to detect these "coupled" mutations. DI minimizes transitive correlations by examining correlations simultaneously across an entire protein (Marks, Colwell, Sheridan, Hopf, Pagnani, & Sander, 2011). Thus, DI achieves unprecedented accuracy by filtering correlations most likely to arise from "direct" association.

Morcos *et al.* and Marks *et al.* have combined DI with distance geometry methods to predict the structures of larger soluble and membrane-bound proteins of varying sizes. They achieve especially high accuracy near key functional areas, which suggests DI is particularly useful for determining structure around important protein sites (Hopf et al., 2012a; Morcos et al., 2011). The entropy maximization step improves the accuracy of DI in comparison to previous correlation measures – specifically local methods such as mutual information. The global nature of this approach leverages the information from all sites, resulting in significant improvement for protein structure predictions that use DI.

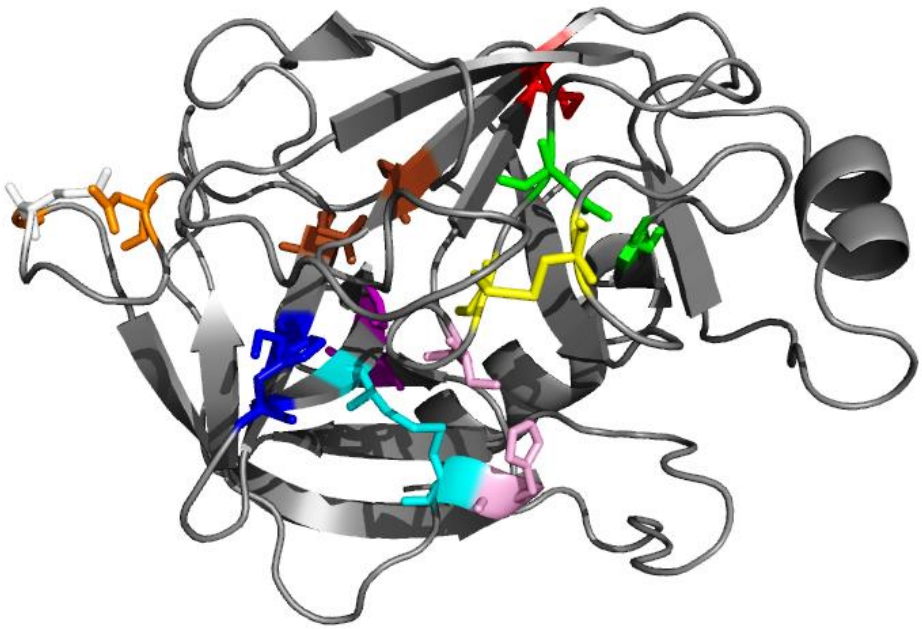


Figure 3: Top 10 Direct Information-based contact predictions for serine protease

The top 10 DI pairs are mapped onto the native structure of S1A serine protease. I have highlighted the top ten DI pairs in matching colors. All top ten pairs are within the 8Å cutoff traditionally used to determine contacts. Pairs are well distributed across the protein structure. However, some pairs are within loop regions, and therefore of no utility to BCL::Fold which only scores contact restraints within SSEs.

Limitations of Direct Information

The dynamic nature of proteins and their interactions complicates correlation-based contact prediction. Sites on opposite sides of a folded protein may be "adjacent" in a homodimer. Similarly, different protein conformational states can drastically alter site-site proximity. One would have to determine which subset of correlations was due to each conformational state to avoid confounding structure prediction with a mixed set of evolutionary constraints for different structures (Hopf et al., 2012a).

Recent publications have shown DI to produce more accurate contact predictions. However, DI cannot distinguish between subsets of contacts from different conformation states, thereby limiting potential accuracy. Results in this work were promising, but required manual contact filtering based on topological predictions, as well as predicted secondary structure and a conservation filter. Furthermore, additional constraints were included during model folding that were based on the predicted secondary structure (Hopf et al., 2012b; Marks, Colwell, Sheridan, Hopf, Pagnani, & Sander, 2011; Morcos et al., 2011). These filtering steps must be performed in addition to the calculation of DI and are non-trivial to implement.

Significance

Overview

Sequencing significantly outpaces structure determination. The protein family database (Pfam) is a database of known protein families. It is sub-divided into a curated set, Pfam-A, and automatically generated families, Pfam-B. There are approximately 12,000 well-characterized protein families within Pfam-A and only approximately half of them have an associated 3D structure. Pfam-B contains approximately 200,000 families and is growing briskly. *De novo* prediction that leverages this rapidly expanding pool of sequence information is poised to increase our knowledge of protein structure. Such computational methods can scale more favorably than current experimental methods to keep pace with the increasing number of protein sequences (Finn et al., 2010). Most importantly, the difficulty of experimentally determining membrane protein-structure further hampers progress. This is of special importance as membrane proteins are the targets of over 50% of current therapeutics. Understanding protein structure is key to identifying druggable targets and developing pharmaceuticals for the ultimate benefit of human health (Bakheet & Doig, 2009).

Membrane Proteins

Determining membrane protein (MP) structures is difficult as they are often too large for nuclear magnetic resonance experiments and are difficult to crystallize. Understanding MP structure is critical as *in silico* methods estimate that MPs comprise 15-39% of the human proteome and are particularly important players in cell signaling pathways often disturbed in

human disease (Ahram, Litou, Fang, & Al-Tawallbeh, 2006). However, only about 2.0% of reported tertiary structures (Raman, Cherezov, & Caffrey, 2006) and 100 unique MP topologies ("Scop Classification Statistics," 2009) consisting of more than one TM span are represented in the Protein Data Bank (PDB) (Berman et al., 2002; Raman et al., 2006; Tusnády, Dosztányi, & Simon, 2004). Furthermore, MPs are targeted by a majority of therapeutics and as such are especially important for the successful understanding and treatment of human disease (Bakheet & Doig, 2009; Fagerberg, Jonasson, von Heijne, Uhlén, & Berglund, 2010). Finally, the advent of personalized medicine necessitates non-promiscuous and targeted therapeutics. Computational methods can address these challenges and assist in *de novo* MP structure elucidation. Predictions that are more accurate will enable the development of pharmaceuticals and the determination of disease mechanisms.

Innovation

Current methods for contact prediction rely on sequence information, which includes the amino acid's biophysical properties and conservation, or site-site correlation information gleaned from multiple sequence alignments (MSA) (Di Lena et al., 2012; Jones et al., 2012; Karakaş et al., 2010; Olmea & Valencia, 1997; Shindyalov, Kolchanov, & Sander, 1994; Xue et al., 2009). Some correlation measures have also been used with machine learning methods to predict contacts (Shackelford & Karplus, 2007). However, no measure combines correlation information from Direct Information (DI), the current gold standard, and other sequence information with machine learning methods. Prior literature has shown that a small number of constraints derived from evolutionary information can be sufficient to predict some smaller protein structures (Ortiz, Kolinski, Rotkiewicz, Ilkowski, & Skolnick, 1999; Ortiz, Kolinski, & Skolnick, 1998; Skolnick, Kolinski, & Ortiz, 1997; Wu, Szilagyi, & Zhang, 2011). I have also shown that contact restraints derived from DI incorporating methods improves protein fold prediction in BCL::Fold for transmembrane proteins.

Furthermore, DI yields promising results and successfully separates most indirect from direct correlations. However, certain interactions – especially direct interactions between sites within homomultimers and interactions between sites in different conformational states – are not distinguishable by DI. Manual filtering addresses some of these issues, but the approach described in this manuscript provides automated topology-based filtering as part of the normal course of contact prediction.

Results and discussion

Benchmark Dataset Analysis

This work focuses on the contact and structure prediction of a set of 25 membrane proteins with known structure and enough known homologous sequences to construct sufficiently deep and accurate MSA to calculate DI values for the majority of amino acid sites in each protein.

Table 1: The 25 membrane proteins used as the benchmark set for this work

Protein	Initial Sequence				Filtered			Unfiltered		
Name	PDBID	L	Opt. E_{val}	TM_{helix}	M_{align}	M_{eff}	Cov	M_{align}	M_{eff}	Cov
ADIC_SALTY	3NCYA	422	1.00E-20	12	1215	1205	0.891	16975	5560	0.884
ADRB2_HUMAN	2RH1A	442	1.00E-20	8	818	451	0.652	22822	5228	0.559
ADT1_BOVIN	1OKCA	292	1.00E-40	6	1068	1043	0.890	8631	3516	0.890
AMTB_ECOLI	1XQFA	362	1.00E-05	11	1048	1021	0.961	4051	1416	0.925
AQP4_HUMAN	3GD8A	223	1.00E-10	7	1073	1062	0.964	5400	1933	0.955
BTUC_ECOLI	1L7VA	324	1.00E-10	10	1049	1045	0.914	9399	4125	0.910
C3NQD8_VIBCJ	3MKTA	460	1.00E-20	12	1072	1068	0.917	11067	5591	0.913
C6E9S6_ECOBD	3RKON	473	1.00E-10	14	1745	1722	0.831	59616	5932	0.588
COX1_BOVIN	1OCCA	514	1.00E-40	12	1157	754	0.979	47394	1289	0.089
COX3_BOVIN	1OCCC	261	1.00E-03	6	684	521	0.958	9105	1444	0.709
CYB_BOVIN	1PP9C	379	1.00E-03	8	1069	581	0.921	49258	855	0.272
FIEF_ECOLI	3H90A	283	1.00E-05	6	1050	1039	0.968	7610	3473	0.933
GLPG_ECOLI	3B45A	180	1.00E-05	6	1092	1073	0.867	4625	2323	0.739
GLPT_ECOLI	1PW4A	434	1.00E-30	12	1611	1604	0.878	25199	10789	0.882
METI_ECOLI	3DHWA	203	1.00E-15	5	1086	1065	0.877	13418	4788	0.877
MIP_BOVIN	1YMGA	233	1.00E-10	7	1032	1010	0.901	5431	1937	0.897
MSBA_SALTY	3B60A	572	1.00E-03	6	1576	1568	0.881	65525	29777	0.388
O67854_AQUAE	2A65A	510	1.00E-03	12	1135	1043	0.825	4351	1657	0.818
OPSD_BOVIN	1HZXA	340	1.00E-20	7	1165	1151	0.803	40460	8873	0.782
Q87TN7_VIBPA	3PJZA	468	1.00E-10	12	1019	923	0.793	3340	1587	0.791
Q8EKT7_SHEON	2XUTA	456	1.00E-10	14	1055	1040	0.706	8196	2983	0.697
Q9K0A9_NEIMB	3ZUXA	308	1.00E-10	10	1024	1005	0.899	3928	1549	0.903
SGLT_VIBPA	2XQ2A	538	1.00E-05	15	1515	1380	0.820	8075	3177	0.784

TEHA_HAEIN	**3M71A**	**306**	1.00E-03	10	822	646	0.971	1503	735	0.948
URAA_ECOLI	**3QE7A**	**407**	1.00E-03	14	1371	1355	0.818	11244	3384	0.747
Statistics										
Mean		**376**	2.42E-04	9.68	1142	1055	0.875	17865	4557	0.755
Standard Deviation		**109**	4.26E-04	3.07	244	309	0.080	18581	5691	0.218
Maximum		**572**	1.00E-03	15	1745	1722	0.979	65525	29777	0.955
Minimum		**180**	1.00E-40	5	684	451	0.652	1503	735	0.089

Protein names and Protein Data Bank IDs (PDBID) are accompanied by the length of the initial target sequence L, the optimal E-value (representing the likelihood of false positive matches allowed in the MSA – lower values are more stringent) used to generate the alignment (Opt. E_{val}) (Hopf et al., 2012a), the number of transmembrane helices predicted with SPOCTOPUS (TM_{helix}), in addition to the number of sequences in the created alignments (M_{align}), the effective number of alignment sequences, which takes into consideration sequence diversity (M_{eff}), and the percent coverage of the initial sequence by the final alignment with columns containing over 30% gaps removed (Cov). The MSA-related data is included for both filtered and unfiltered datasets.

The 25 membrane proteins listed in Table 1 are a diverse set of non-trivial (having more than four transmembrane helices) α-helical transmembrane proteins with more than 1000 homologous sequences of sufficient coverage. Site-coverage (Cov) is the percent of the target sequence sites that map onto the final MSA after one removes columns with a large number of gaps. Based on prior work (Morcos et al., 2011), a threshold of 30% gaps was used for this analysis. These 25 membrane proteins come from 23 different Pfam families, contain a maximum of 15 helices, and have a maximum initial target length of 572 (Hopf et al., 2012a).

Both filtered and unfiltered alignments are included in Table 1. Filtered MSA are significantly smaller – on average nearly 16 fold so. Maximum, minimum, and average unfiltered alignment sizes are 65525, 1503, and 17865 respectively. The maximum, minimum, and average filtered alignment sizes are 1745, 684, and 1142, respectively. Coverage increases nearly 16% from approximately 76% coverage in unfiltered to 88% in filtered alignments. Effective alignment size (M_{eff}) significantly decreases the number of MSA sequences. M_{eff} captures the number of sequences in the alignment after down-weighting sequences that are highly similar to the original

target sequence – over 80% identity (Morcos et al., 2011). M_{eff} is nearly five-fold higher in unfiltered alignments on average. However, the minimum M_{eff} for filtered alignments is still 451.

Sequence coverage is a proxy for determining whether functional pressures were similar across all sequences included in the MSA. Evolutionary pressures result in the detected DI correlations essential to accurate contact prediction. Sequence diversity is necessary to capture sufficient mutations for an accurate DI. However, the functional role and selective pressures must be similar between all proteins included, enabling DI to detect the relevant and same overarching signal due to functional short-range interactions. An ideal DI calculation would isolate evolutionary coupling produced by the selective pressures associated with the overall functions of a single protein. The results seem relatively robust to the exact coverage value used, but generally, target sequence coverage in the range of 70% indicates a high probability of similar functional pressures.

Figure 5 depicts the maximum number of contacts, amino acid pairs with C-β to C-β distances within 8Å based on their respective Protein Databank (PDB) file, that are possible across the benchmark set at various minimum separation values – 0, 3, 6, and 12. Some atoms within the PDB files have undefined coordinates and as such their positions cannot be accurately determined. I excluded all pairs that contained C-β atoms with undefined position from this analysis. A minimum separation of 0 shows the total number of possible contacts if the sequence aspect is disregarded. Separations of 3 and 6 were selected to visualize the rapidity with which the number of possible contacts decreases as sequence separation is increased. Using a minimum separation of 6 removes the vast majority of contacts that would be considered trivial and are of especially low utility to BCL::Fold, which uses idealized secondary structures during folding and as

such contacts within an SSE do not alter folding predictions. However, given an L-based fraction of contact restraints, such trivial predictions do limit the number of more informative contacts that could be included. In addition, a minimum separation of 6 is generous enough to avoid removing valuable contact information between the ends of adjacent SSEs. Finally, a minimum separation of 12 is the traditional cutoff for "long-range" contact restraints. However, the difference between the total number of contacts is relatively similar between a minimum separation of 6 and 12. The maximum decrease is 22.3%, the minimum decrease is 6.3%, and the average decrease is only 10%. The topology of alpha-helical membrane proteins and the high SSE content are such that pairs with a minimum separation between 6 and 12 are unlikely to be in contact. Such pairs are often within and thus "held" apart by a relatively linear SSE. The pairs that do form contacts within this range are either on adjacent ends of SSEs separated by short loop regions and are thus valuable for determining SSE rotation and orientation within the membrane. These pairs may also lie within long flexible loop regions that the BCL does not leverage during folding runs.

One should also note that for minimum separations of 6 and 12 the maximum number of possible contacts is on average 1.6 and 1.5 times the length of the trimmed protein. Thus, for a perfectly or mostly accurate contact ranking by direct information or some other method, taking any L-fraction beyond this range would be counter-productive as it would likely extend beyond the set of possible contacts. However, if the method used also enriches for near-contacts (contacts outside of the traditionally strict 8Å cutoff but still within approximately 14Å) an extension of the selected restraints that extends beyond traditional contacts could still be

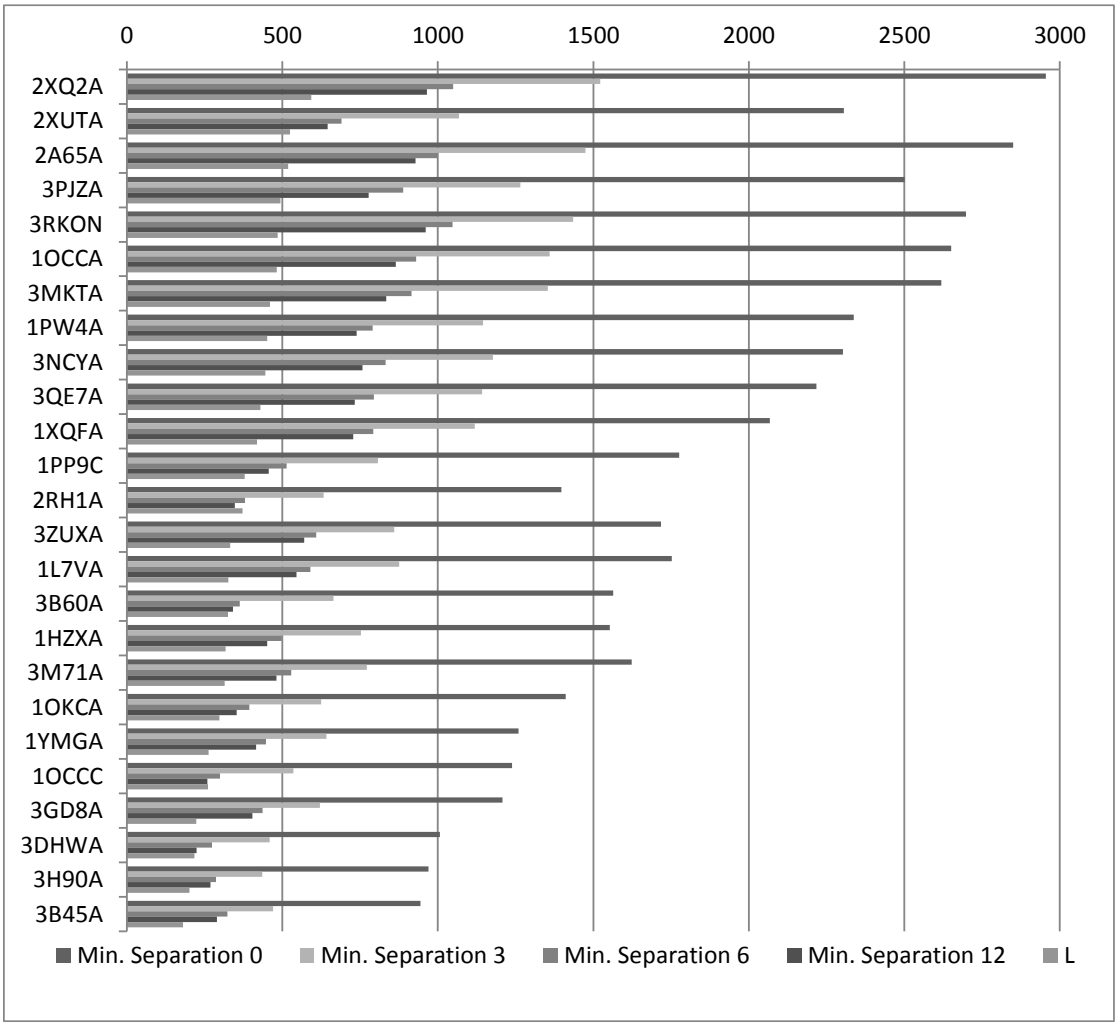


Figure 5: Number of Possible Contacts at Varying Minimum Separations Compared to Trimmed Protein Length

The total possible contacts across the benchmark set is depicted above sorted by the trimmed length L for each protein at various minimum separation thresholds – 0, 3, 6, and 12. These values were determined from Protein Databank Files that have a small number of positions with undefined coordinates. Thus, there is a small discrepancy between these totals and the ideal. The average L-fraction of contacts at the thresholds of 0, 3, 6, and 12 are approximately 5, 2.5, 1.6, and 1.5 respectively. Thus, while there is some difference between the possible contacts between a minimum separation of 6 and 12 the average decrease is only ~10%. The maximum and minimum decreases across the benchmark set from minimum separations of 6 to 12 are 22.3% and 6.3%.

beneficial if it includes near-contacts that provide information regarding areas of a protein structure not already addressed by the predicted contacts.

Naïve Direct Information-Based Contact Restraints

I calculated DI for all 25 membrane proteins listed in Table 1 for both filtered and unfiltered MSA. The filtering process removes sequences that individually do not align to cover at least 70% percent of the original target sequence used to create the alignment. This is the same cutoff used by Hopf *et al.* to determine membrane protein structure for the same set of membrane proteins (Hopf et al., 2012a). In both cases, I used e-value cutoffs for sequence aggregation of 1E-03, 1E-05, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40. In addition, Hopf *et al.* also provided a set of "optimal e-values" for prediction within the work. They selected e-values by

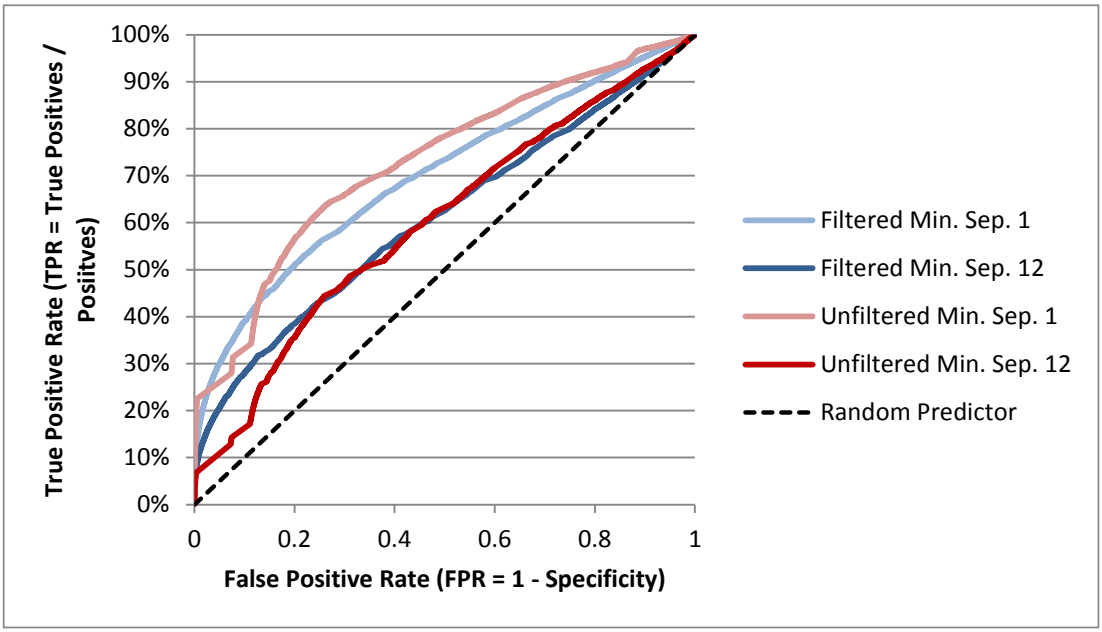


Figure 6: Direct Information Receiver Operating Characteristic Curves (Filtered and Unfiltered MSA for All Site-Site Pairs at Minimum Separations of 1 and 12)

The above graph displays ROC curves created by calculating the DI for the pairwise contacts for all 25 membrane proteins, aggregating all pairs, and ranking them by the magnitude of the DI values assuming that higher DI indicates two residues are in contact. AUC given a minimum separation of 1 is approximately 0.700 and 0.729 for filtered and unfiltered respectively. AUC given a minimum separation of 12 is approximately 0.611 and 0.601 for filtered and unfiltered respectively. A minimums separation of 6 performs very similarly to a minimum separation of 12 (not shown). Filtered outperforms unfiltered for long-range contacts, which are of special importance for structure prediction.

comparing M_{eff} against coverage and selecting the e-value that balanced between having good target sequence coverage but also sufficient sequences for a deep and accurate MSA. Table 1 contains the optimal e-values (Opt. E_{val}). This set represents the default for experiments in this thesis unless otherwise specified.

Figure 6 displays a ROC curve for DI values from unfiltered and filtered MSA, at minimum separations of 1 and 12, created using the set of optimal e-values described in Table 1. A minimum separation of 1 focuses on all predictions regardless of how trivial. The minimum separation of 12 isolates the long-range contacts, which are more informative for protein structure prediction. Knowing that two amino acids separated by only one other amino acid are near each other is not helpful. However, this threshold does exclude amino acid pairs that are directly connected via a peptide bond (a minimum separation of 0). AUC values are 0.700 and 0.729 for filtered and unfiltered at a minimum separation of 1 respectively. When one only considers long-range contacts (minimum separation of 12) the AUC values are 0.611 and 0.601 for filtered and unfiltered respectively. Thus, looking at overall prediction unfiltered outperforms the filtered set when one includess trivial contacts but when one only considers long-range contacts the filtering step results in a small improvement in AUC. The AUC is also significantly decreased for long-range contact prediction as more distant contacts are more challenging to predict. One possible explanation is that positions near one another in the final folded structure and also in primary sequence are near one another and thus influence one another more frequently. In other words, their increased interaction frequency may amplify the detected evolutinary influence in comparison to pairs distant in primary sequence. Of note, results graphed from a set with a

minimumse separation of 6 nearly superimpose upon those from a minimum separation of 12. This is due in large part to the fact that there are relatively few contacts in the intervening range.

To leverage contact prediction for protein folding one only needs to predict a relatively small subset of accurate long-range contacts. Thus, I have focused on the most confident predictions using the precision as one increases the fraction predicted positive. It is especially important to focus on this upper segment, as a relatively small set of accurate contact predictions is sufficient for accurate three-dimensional protein structure prediction. Prior work has estimated that accurately predicting 25-35% of the possible contacts is satisfactory for folding (Marks, Colwell, Sheridan, Hopf, Pagnani, Sander, et al., 2011).

To compare the precision versus fraction predicted positive graphs across this range of the most confidence contacts I have determined that the top L contacts across the entire benchmark set comprise 0.55% of all possible pairs. As such, I will use the integral of the precision versus fraction predicted positive curve from 0.01% to 0.55%, a range that should capture the number of contacts desired and focuses on only the most confident predictions. A starting point of 0.0001 minimizes the effects of noise at extremely low fractions predicted positive.

The relative ordering between filtered and unfiltered is the same for the precision versus fraction predicted positive (Figure 7) as it was for the ROC curves (Figure 6). Unfiltered outperforms filtered for pairs with a minimum separation of 1, but is outperformed by filtered for long-range contacts (minimum separation of 12). The integrals from 0.0001 to 0.0055 are 0.656 and 0.787 for filtered and unfiltered respectively, at a minmum separation of 1. For long-range contacts (minimum separation of 12) the AUC values are 0.537 and 0.494 for filtered and unfiltered respectively. Using this metric the improvement in accuracy for the most confident

predictions for filtered over unfiltered is more appropriately represented than with the area under the ROC curve. Also, there is also a signficant decrease in magnitude transitioning from results from a minimum separation of 1 to 12 – once again highlighting the increased difficulty of predicting long-range contacts. Unfortunately, contacts between pairs near one another in primary sequence are much less informative for protein folding. Many such close-range contact restraints lie within the same SSE and therefore are of no benefit to BCL::Fold, which assembles

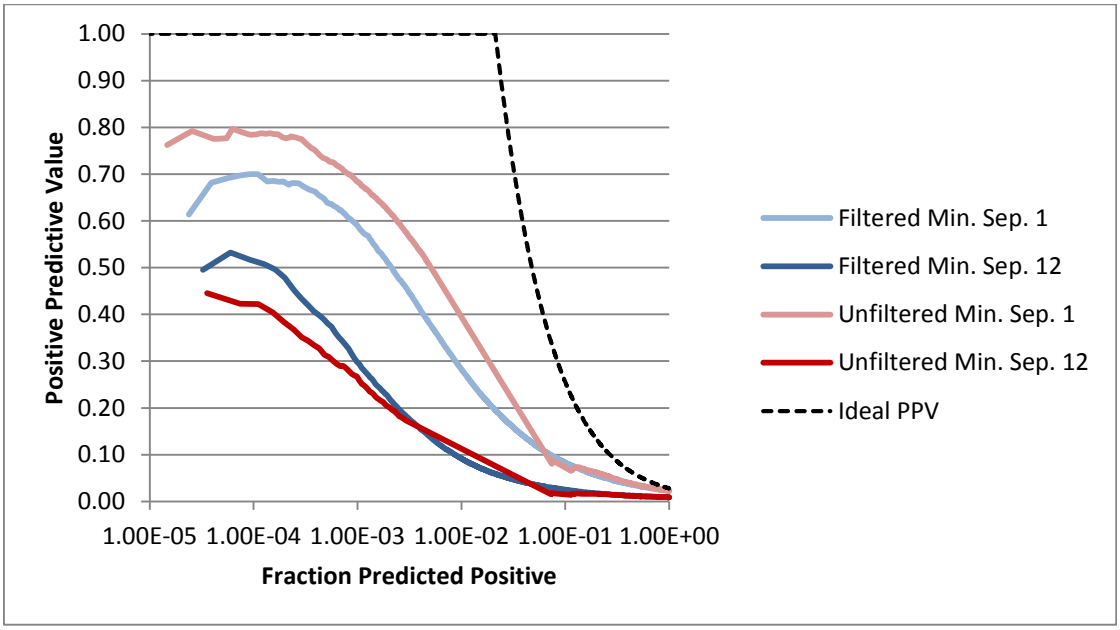


Figure 7: Direct Information Logarithmic Positive Predictive Value Curves (Filtered and Unfiltered MSA for All Site-Site Pairs at Minimum Separations of 1 and 12)

The above contains a graph showing precision as the fraction predicted positive increases. I determined each curve using DI for all pairwise contacts across the benchmark set, and ranking by the magnitude of the DI values assuming that higher DI indicates two residues were in contact. The percent of contacts selected across the benchmark for all top 1L contacts is 0.55%. Thus, integrating from 0.0001 to 0.0055 for the desired minimum separation most accurately reflects the final top L contact restraint set accuracy. The integral of the precision at a minimum separation of 1 from 0.0001 to 0.0055 is approximately 0.656 and 0.787 for filtered and unfiltered respectively. The integral of the precision at a minimum separation of 12 from 0.0001 to 0.0055 is approximately 0.537 and 0.494 for filtered and unfiltered respectively. A minimums separation of 6 performs very similarly to a minimum separation of 12 (not shown). Greater precision initially and continuing out as the fraction predicted positive increases is better. The black dashed line depicts ideal performance. Predictions derived from filtered MSA once again perform best for long-range contacts, which are most informative for protein fold prediction.

idealized SSE into protein fold predictions. As such, only contacts between SSEs, very often long-range contacts, are informative for assembling protein models.

Thus, based on these metrics across the aggregated set of predicted contacts, filtering, or isolating for sequence sets that are more likely to maintain functional similarity to the original target sequence used to create the MSA, leads to more precise top-ranked DI predicted long-range contact pairs.

Figure 8, Figure 9, and Figure 10 depict DI restraint file accuracy across the top L/10, L/5, L/2, L, and 2L for both unfiltered and filtered MSA. The protein with the highest L/10 accuracy is 3GD8A with 91% (filtered) and 54% for 3MKTA (unfiltered). For a minimum separation of 6 the average accuracies at L/10, L/5, L/2, L, and 2L are 39.8%, 34.4%, 24.3%, 17.2%, and 11.3% for filtered and 23.0%, 18.1%, 12.7%, 8.8%, and 6.0% for unfiltered respectively. For a minimum separation of 12 the average accuracies at L/10, L/5, L/2, L, and 2L are 41.1%, 35.0%, 24.1%, 16.5%, and 10.7% for filtered and 20.3%, 16.1%, 10.7%, 7.5%, and 5.0% for unfiltered respectively.

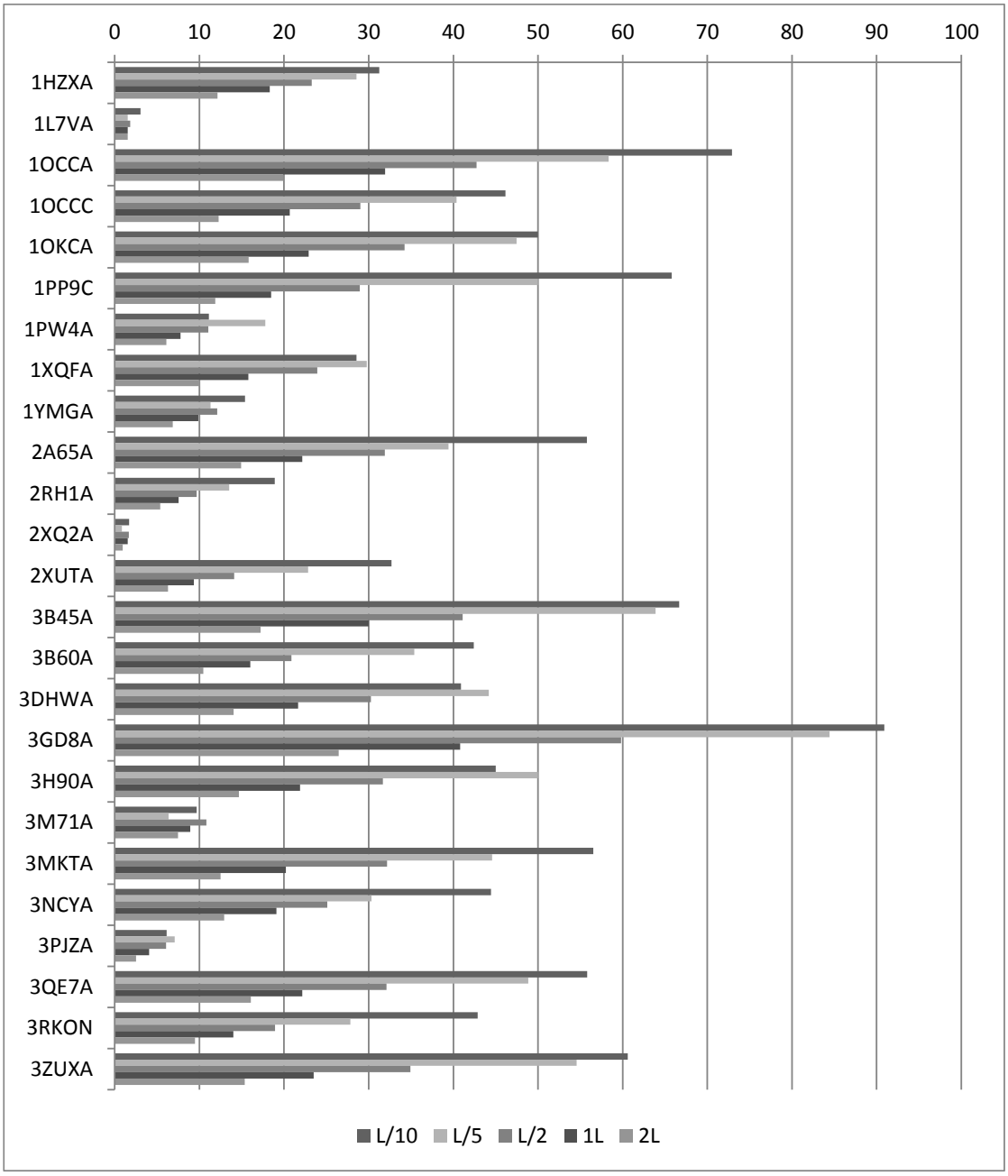


Figure 8: DI-Based Restraint Accuracy for Filtered Alignments across the Protein Benchmark Set at Various L-Fractions

I ranked contact pairs by the magnitude of their direct information and the top L-fractions were taken and predicted as contacts. I then evaluated contacts by comparing predictions to the C-β distances (within 8Å) as resolved in the Protein Data Bank structures. The accuracy for all DI-based restraints from filtered alignments achieves a maximum of 91% for protein 3GD8A. As one uses larger fractions of the top L contacts (predicts more contacts) the accuracy drops significantly in most cases. In all but 6 of the 25 benchmark proteins the top L/10 contact fraction accuracy is highest.

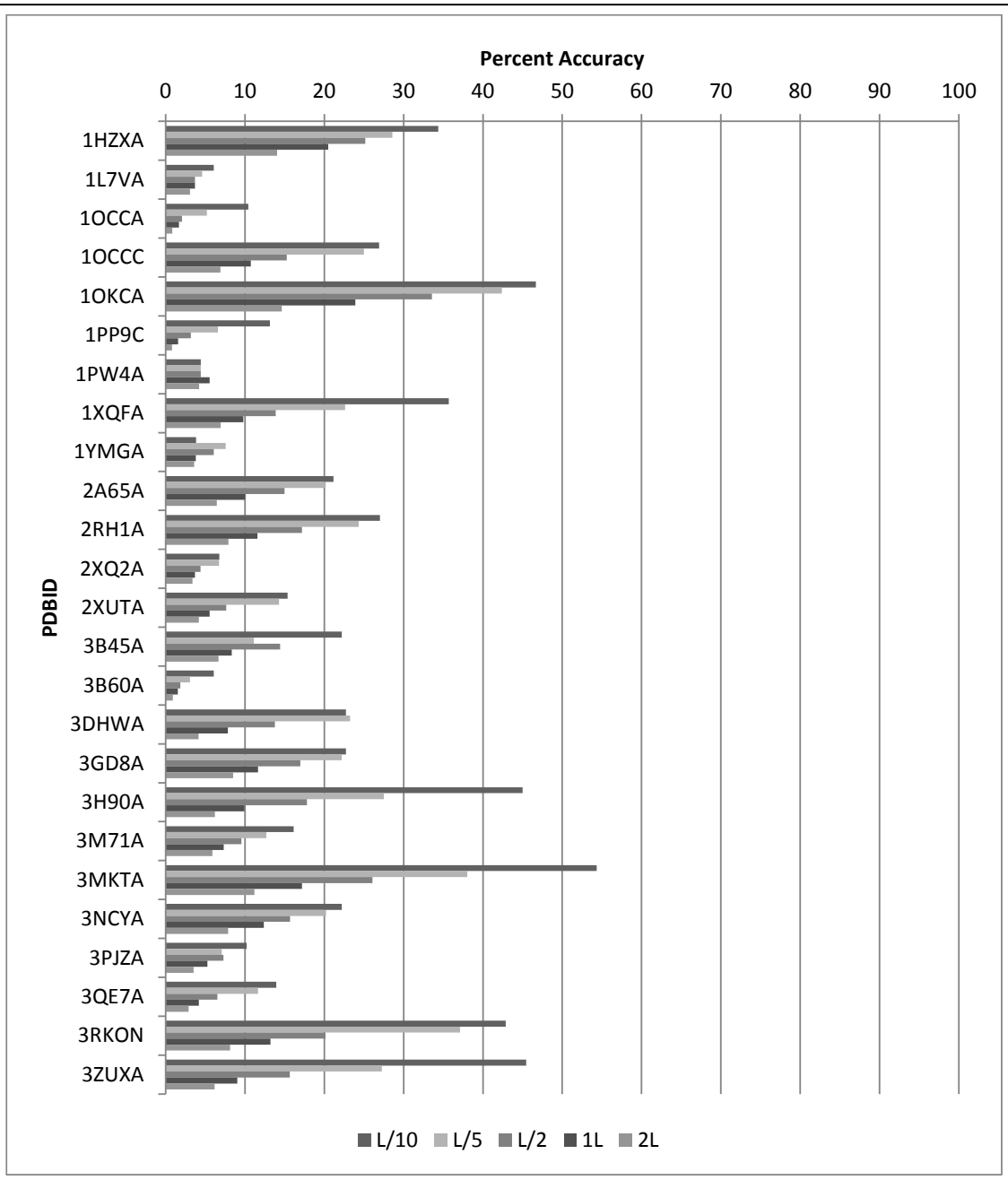


Figure 9: DI-Based Restraint File Accuracy for Unfiltered Alignments across the Protein Benchmark Set at Various L-fractions

I ranked contact pairs by their direct information and isolated the top L-fractions as predicted contacts. I determined accuracy by verifying whether C-β distances were within 8 Å as resolved in the Protein Data Bank structures. The accuracy for all DI-based restraints from unfiltered alignments achieves a maximum of 54% for protein 3MKTA. As more contacts are predicted (increasing fractions of L) the accuracy drops significantly. In all but 4 of the 25 benchmark proteins the L/10 accuracy is highest.

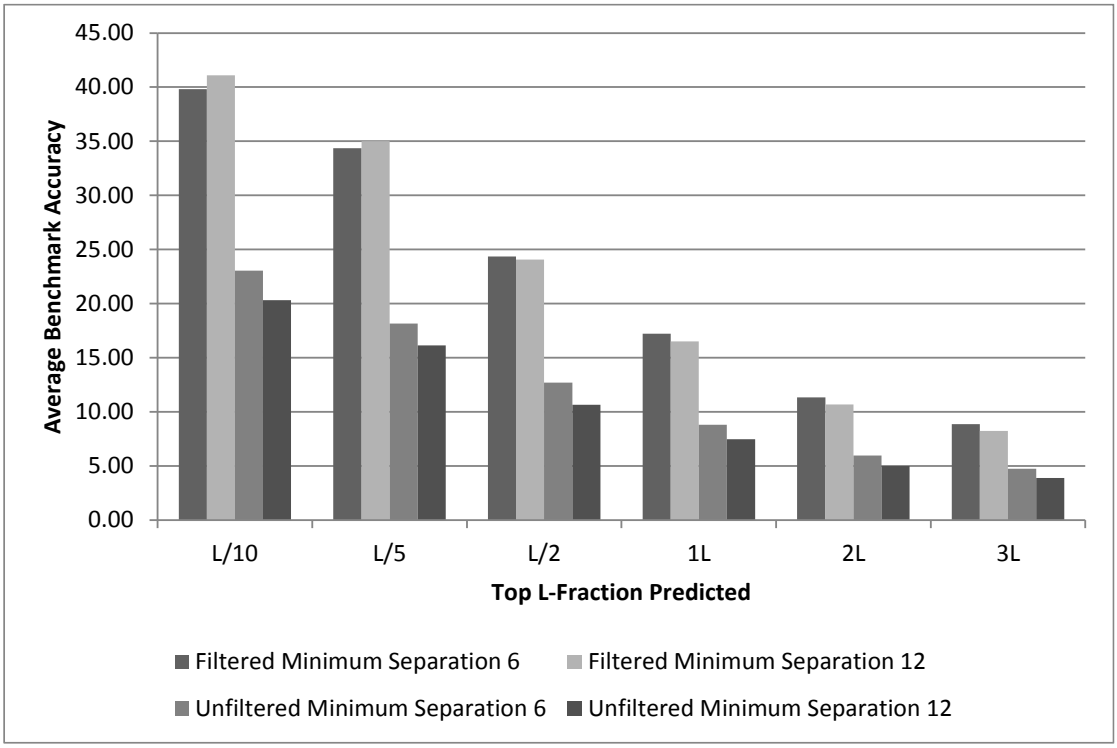


Figure 10: Filtering MSA Improves the Accuracy of Top Ranked DI Predictions at Minimum Separations of 6 and 12

Accuracies for different top L-fraction cutoffs across the entire benchmark set of 25 proteins were averaged and are depicted above. Average accuracies for DI-based contact restraints are consistently higher from filtered than unfiltered MSA. This accuracy discrepancy decreases as the total number of constraints predicted increases towards 3L. Additionally, average accuracy at a minimum separation of 6 is higher across all unfiltered direct information-based contact predictions as well as for filtered contact predictions larger than L/5.

Results in Figure 10 confirm the conclusion from Figure 6 and Figure 7, which indicate that filtering MSA improves the accuracies for the top ranked DI predictions. These top ranked predictions are used at various L cutoffs as restraints for folding and the accuracy is consistently higher across L fractions and for both a minimum separation of 6 and 12. The significant decrease in accuracy as one uses more of the top ranked direct information pairs, and thus includes less confident predictions, suggests that confidence-based scoring is a promising avenue for further

leveraging ranking information during model scoring. If higher ranked positions are more accurate they should also be weighted more highly when scoring a predicted protein model using a contact-restraint score.

After determining accuracy for all proteins using a naïve implementation of direct information (one that does not include any topology or secondary structure filtering), I examined how accuracy correlated with several protein and MSA characteristics including protein length and the effective MSA depth (M_{eff}). I compared sets individually for predictions from both filtered and unfiltered MSA (Figure 11). For both sets, as protein size increases the accuracy decreases (Figure 11 panels A and B). This was intuitive as large proteins have many more contacts and increased complexity. In addition, the shift upwards in accuracy is also prominent in Figure 11 panel B compared to A. In addition, when gauged linearly, there is a slight downward trend as M_{eff} increases for both predictions from filtered and unfiltered MSA. However, the lower portion of the range is truncated, as all benchmark proteins were selected such that a minimum total number of sequences is greater than 1,000. In addition, when analyzed using higher order trend lines, one can detect that the highest accuracies occur in the middle of the M_{eff} range. Thus, too many and too few sequences appear to hinder prediction accuracy. Too few sequences results in too little evolutionary information, while using too many increases the odds that one incorporates related but functionally distant proteins who introduce confounding evolutionary pressures. More intelligently selecting or filtering sequences from large MSA such that functional similarity is preserved can likely increase the accuracy of proteins from the middle of the range upward as confounding sequences would be excluded.

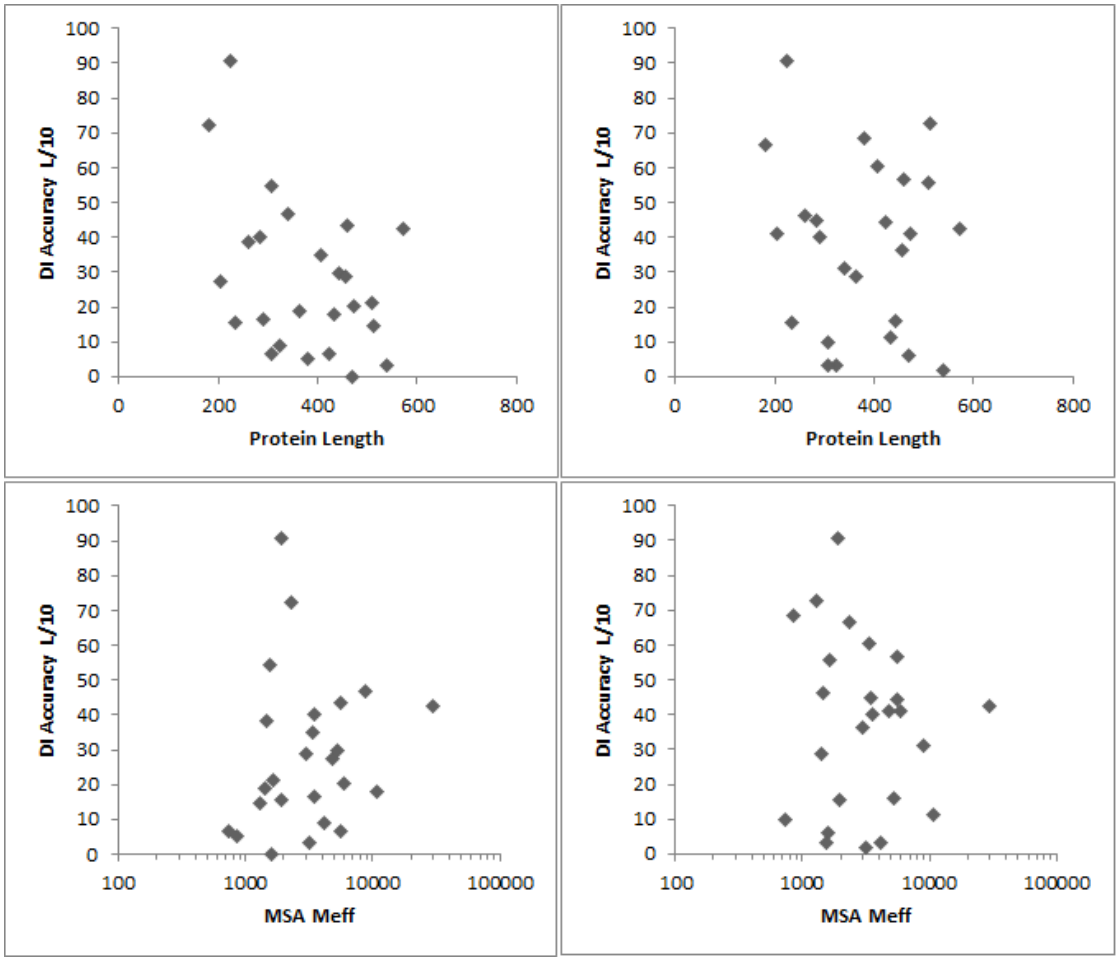
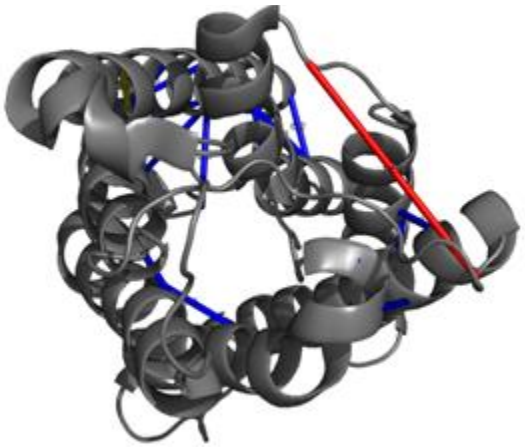


Figure 11: Top L/10 DI Accuracy vs. Length and M_{eff} for Unfiltered and Filtered MSA

Above accuracy for the top L/10 direct information-based predicted contacts is compared to protein length and the effective MSA size. Contacts from Unfiltered MSA are displayed in panels (A) and (C) while results from Filtered MSA are given in panels (B) and (D). Accuracy for the top L/10 DI restraints decreases as protein size increases for predictions from both unfiltered and filtered MSA. The decrease is less significant for predictions based on filtered MSA as accuracy in general is increased and most of all for larger proteins. Performance from unfiltered and filtered MSA also decreases as the effective number of sequences increases, if gauged by a linear trend. However, one can also see a trend of improvement followed by decline such that the maximum accuracy coincides with the center of the range of M_{eff} for this benchmark set.

One potential method could include examining whether sequences contribute to or detract from the major DI trends. For example, one could construct MSA using decreasingly strict

coverage parameters and calculate at each stage the effect on DI across sites. By evaluating by site one may also be able to preserve snippets of the sequence, which still seem to functionally match thus preventing excessive exclusion of valuable evolutionary information.



some small degree of conformational shift such state. Automated methods for detecting such rmation would increase accuracy and our egardless, DI appears to be very promising for lutionary information and functionally similar

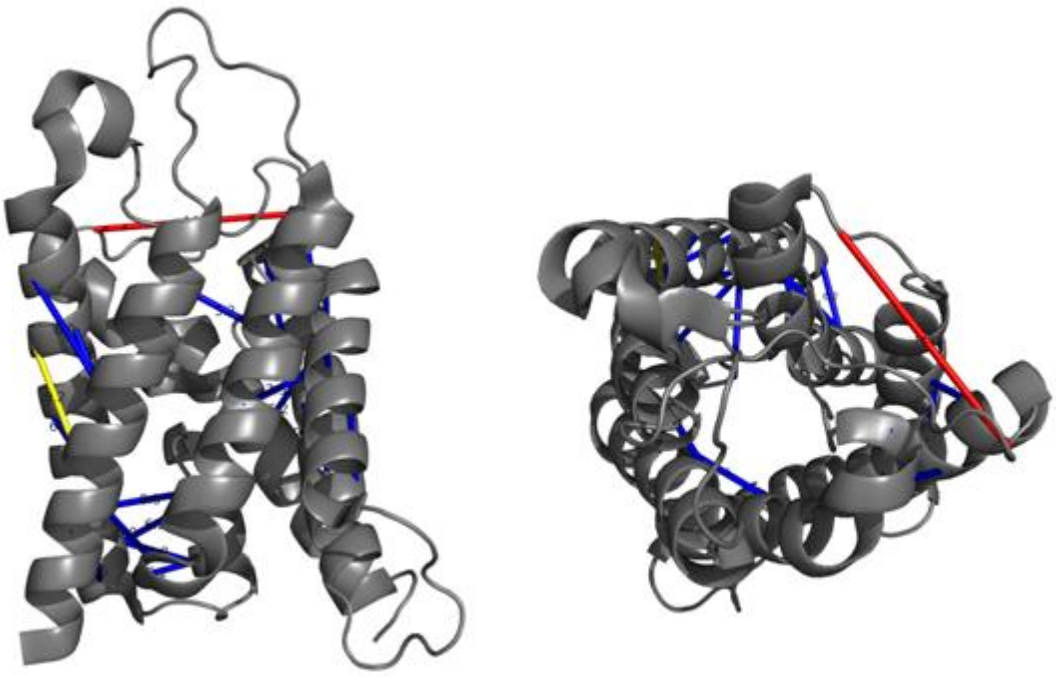
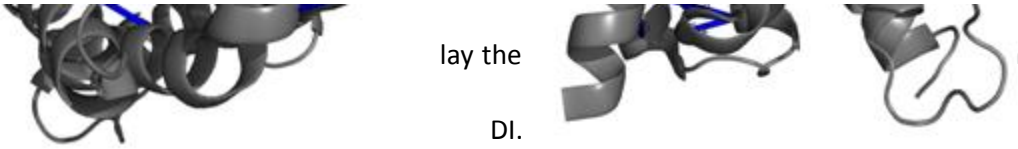


Figure 12: High Accuracy Top L/10 Visualized Contacts for 3GD8A (Filtered)

Above is an example of the top L/10 predicted contacts for 3GD8A showing both the distribution across the protein structure and the potential accuracy of the top DI pairs for contact prediction. The accuracy is approximately 91%. Blue lines connect sites correctly predicted to be within 8Å of one another. Yellow lines connect incorrect pairs that are "near-contacts" (between 8Å and 12Å). Red lines connect incorrectly predicted pairs. For this set, only two pairs are incorrect at 9.2Å and 18.1Å. The latter is a predicted contact between flexible loop regions.



lay the

DI.

Figure 12 includes one of the best examples of DI prediction, 3GD8A in this benchmark set, which has an accuracy of 91%. This particular set of constraints for the top L/10 has identical accuracy for both the filtered and unfiltered MSA. The accuracy is especially impressive given the stringency of the 8Å cutoff and the fact that one of the "incorrect" contact predictions is still within 9.2Å – only slightly outside the cutoff. Predicted constraints for the other members of this benchmark set do not perform as well using this strict cutoff. However, restraints are enriched for

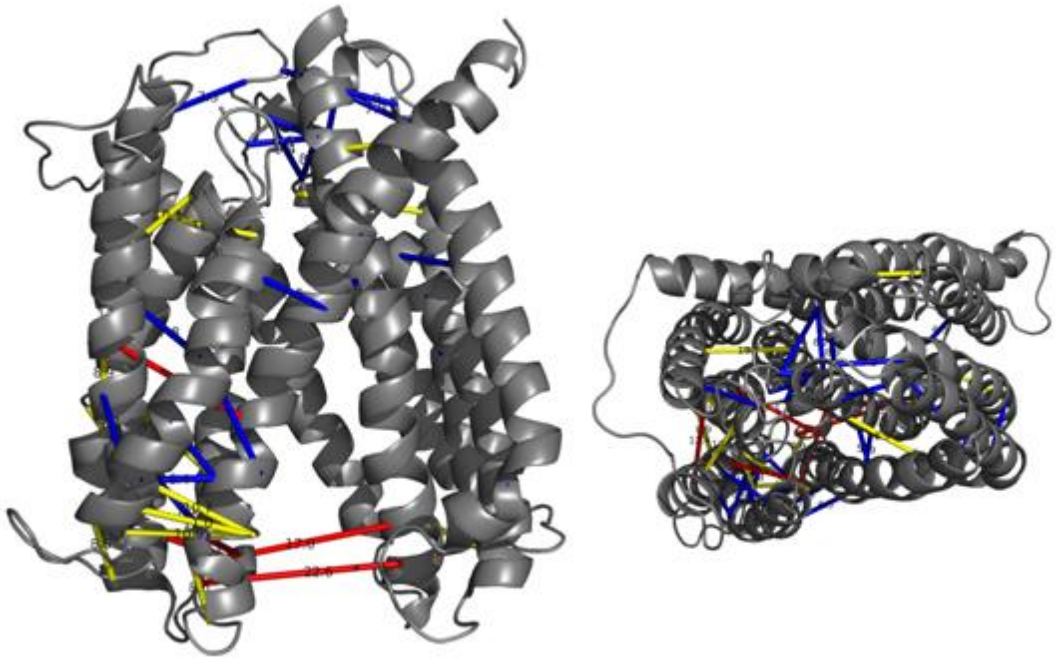


Figure 13: Medium Accuracy Top L/10 Visualized Contacts for 3MKTA by DI (Filtered)

Above is an example of the top L/10 predicted contacts for 3MKTA showing both the distribution across the protein structure and another example of the high accuracy levels possible for the top DI pairs. Accuracy for this protein is 56.5%. Blue lines connect correct contact pairs (within 8Å of one another). Yellow lines connect incorrect pairs that are "near-contacts" (between 8Å and 12Å). Red lines connect incorrect pairs. Few pairs are incorrect, and most of them are within the 8-11Å range. Many of the incorrect pairs are still "near-contacts".

pairs within enough proximity of one another to significantly improve BCL::Fold predictions as can be seen in the section on protein structure prediction with contact information.

Figure 13 contains a more representative example of top L/10 constraints. The accuracy for 3MKTA is 56.5%. Some of the proteins within this dataset have significantly lower accuracies.

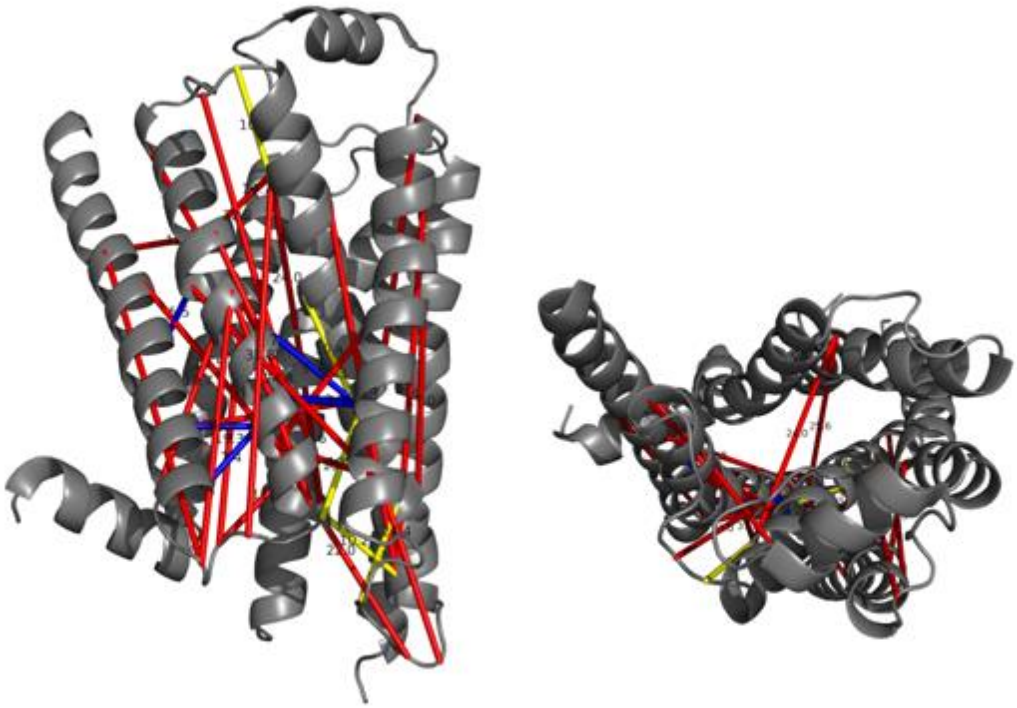


Figure 14: Low Accuracy Top L/10 Visualized Contacts for 2RH1A from DI (Filtered)

Above is an example of the top L/10 predicted contacts for 2RH1A showing both the distribution across the protein structure and poor accuracy in the absence of topology prediction assistance. Blue lines connect correct contact pairs (within the 8Å threshold). Yellow lines connect incorrect pairs that are "near-contacts" (between 8Å and 12Å). Red lines connect incorrect pairs. Accuracy for top L/10 is approximately 18.9%.

Figure 14 shows how poorly DI can perform in some cases. Accuracy for the top L/10 for 2RH1A is 18.9% with many incorrect contacts predicted between opposite sides of the membrane topology. Elimination of such predictions, which one can easily identify in the most extreme cases, can significantly improve accuracy. Hopf *et al*. manually filter out trans-topology predictions, and I address them within my machine learning models by using a transmembrane position descriptor and a descriptor indicating whether a pair is within the same SSE (Hopf et al., 2012a). In addition, I have also evaluated the improvement of such filtering techniques across the benchmark within

the section “Improving Direct Information-Based Contact Prediction Accuracy with Predicted Topology and Secondary Structure”.

While incorrect pairs within the same SSE diminish the number of informative contact restraints, they do not necessarily alter BCL::Fold model prediction once SSEs are selected. The BCL utilizes idealized SSEs and any negative impact on the contact score for such a pair would be equivalent across generated models. However, it is possible that incorrect contact restraints may alter the SSEs selected from the initial pool which would affect the final model prediction. When obvious topological false positives are removed most of the remaining incorrect restraints are outside the traditional 8Å cutoff but still in relatively close proximity to one another – potentially providing useful information to BCL::Fold. DI’s accuracy coupled with filtering and scoring functions that allow for near-contacts enables BCL::Fold to achieve higher prediction accuracies on membrane proteins.

Evaluation of Aggregated (Maximum/Mean) Direct Information Values Across Multiple Sequence Alignment Parameters

The accuracy of contacts predicted using direct information varies significantly across results computed from MSAs determined using different e-value thresholds and with or without filtering. Filtering, as has been previously shown, usually improves results. However, the loss of evolutionary sequence space by reducing the number of usable sequences sometimes results in decreased prediction accuracy. Performance across e-value thresholds can differ greatly. Hopf *et al.* chose an optimal e-value for each protein in their benchmark set by comparing the number of effective sequences in each MSA to the coverage of the original target protein used to create the MSA. This approach enabled them to approximate the balance between more data, the number of diverse sequences, and the noise introduced by proteins no longer constrained by the same functional pressures (that covered less of the target protein's sequence). Using e-values from the transition point between coverage and included sequences does not yield ideal results as can be seen from the large variability in accuracy. Thus, I attempted other simple methods to determine whether it would be possible to use all the MSA alignment data to make more consistently accurate predictions than the previous approach, which employed manual analysis. These six methods take either the maximum or mean direct information value across results produced using all filtered, unfiltered, or both sets of MSA.

Figure 15 displays a comparison of the accuracies resulting from each of these methods compared to both the filtered and unfiltered naïve direct information based methods as well as the most accurate processed and filtered direct information based method. Some parameter sets show improvement over unfiltered naïve direct information, however all six aggregated sets

perform significantly worse than a naïve filtered set of predictions. Interestingly, the mean instead of the maximum yields better results for each set of MSA. I had originally hypothesized that the maximum would perform best as random noise in an alignment created using the non-optimal e-value should result in low direct information values. In such a case, only the signal from the best aligned MSA should be detected. However, it appears that false signals from noncontact sites that have been poorly aligned due to improper e-value selection outcompete the true positives. Exceptionally poor sampling of evolutionary space, as can occur from overly restrictive e-values, can introduce selection noise that obscures correlation due to functional pressures and physical proximity. Thus, the mean may reduce the effects of noise at the margin by decreasing the impact of one high direct information value arising as an artifact of sequence selection and alignment. Nevertheless, utilizing all correlation values across MSAs and filtering status using these simple methods performs significantly worse than carefully selecting a single MSA for each protein that balances evolutionary space sampling against target sequence coverage.

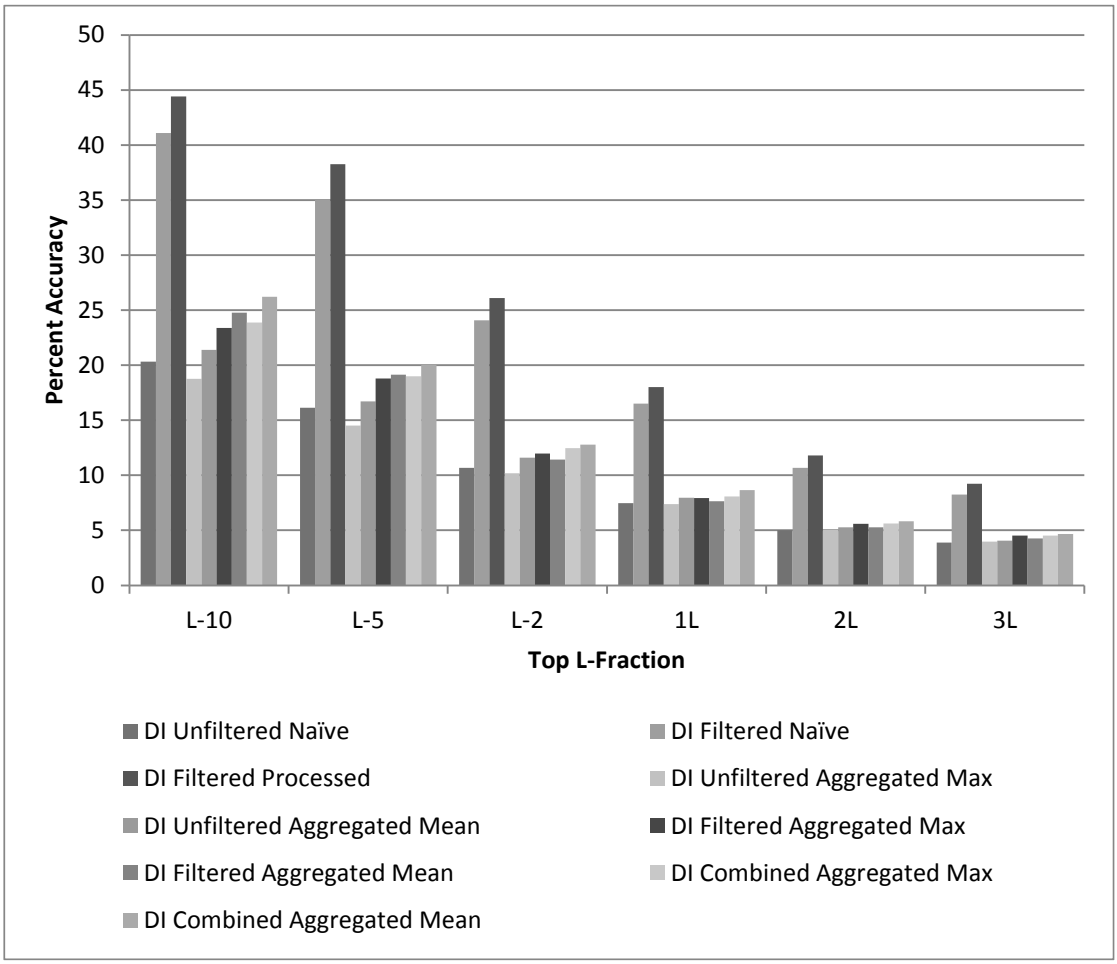


Figure 15: Highest Accuracy for DI Filtered Naïve and DI Filtered Processed When Compared across L-Fractions and Different Correlation Aggregation Methods

I calculated correlation information using the optimal set of e-values (the naïve direct information based contact predictions) from both filtered and unfiltered MSA. In addition, I have also improved accuracy by excluding contact predictions separated by too large a distance as determined by topology prediction – the processed set. These filtered processed and unprocessed sets are the most accurate predictions. The other sets combine results across all e-values (1e-03, 1e-05, 1e-10, 1e-10, 1e-15, 1e-20, 1e-30, and 1e-40) and are unfiltered, filtered, or combine both. For these sets the mean of the correlation values across sets results in higher accuracy than the max for each pair. In addition, results are also best for the combination across all sets, followed by filtered, and then unfiltered MSA sets.

Improving Direct Information-Based Contact Prediction Accuracy with Predicted Topology and Secondary Structure

While direct information is usually high for amino acid pairs in close proximity in the native protein fold there are often false positives. These occur due to several reasons including: homomultimers with contradicting evolutionary pressures confound intra-protein contact prediction, selection pressures across proteins with multiple receptor/signaling domains that are evolutionarily related but not in physical proximity, as well as other random associations possibly due to sequence selection. It is possible to increase the accuracy of direct information-based contact prediction by filtering out contacts that are not likely by leveraging secondary structure

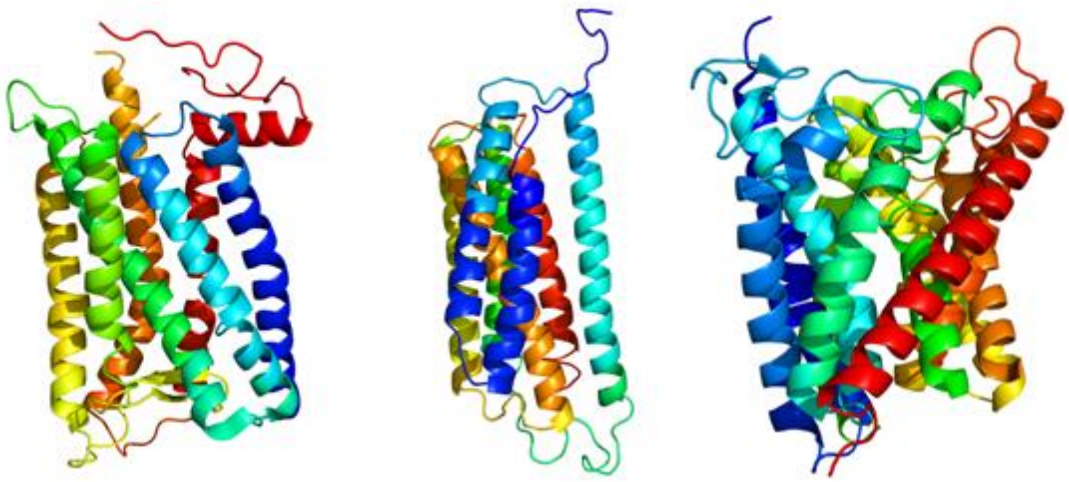


Figure 16: Secondary Structure Predicted by SPOCTOPUS with Relatively High Accuracy for Contact Prediction Filtering

1HZXA, 1OCCC, and 1XQFA (left to right) secondary structure predictions using SPOCTOPUS are depicted above in rainbow spectrum. Whenever a string of predictions changes between inner-membrane, outer-membrane, and transmembrane (i/o/M) amino acids are assigned a new index in sequence and different indices are displayed with new colors along the rainbow spectrum. I re-assigned the short re-entrant loops to inner or outer loops as appropriate to simplify SSE separation prediction. Accuracy is high with only minor overlap at the edges of secondary structure in most cases. An especially good example of this overlap can be seen in the right most helix of 1OCCC. In addition, a rare case where SPOCTOPUS predicts a transition near the center of an alpha-helix can also be seen on the left side of 1OCCC.

and topology prediction. For example, two sites predicted to be on either end of a transmembrane helix are very unlikely to be in contact. However, such a pair may yield a high direct information value if they are part of a receptor-signaling pathway, which would constrain their evolution similar to functional roles that rely on the physical proximity of both amino acid sites. Thus, removing such pairs using SSE or topology prediction may improve contact prediction accuracy. Hopf *et al.* filtered their results using such methods with the same membrane protein benchmark set that is the focus of this work (Hopf et al., 2012a).

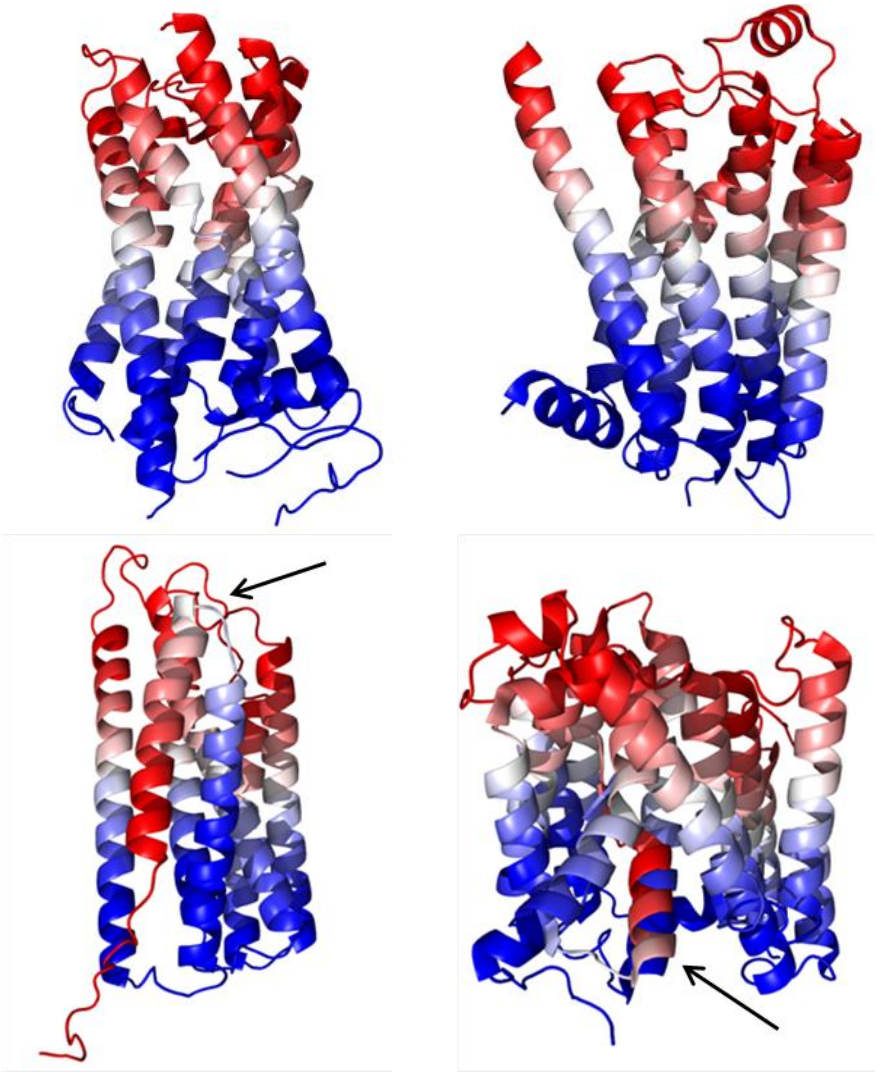


Figure 17: Predicted Transmembrane Position Descriptor Effectively Captures Membrane Topology Information

I used SPOCTOPUS to predict topology and then assigned each amino acid within all proteins a 0 for inner membrane (blue), 1 for outer membrane (red), and a value between 0 and 1 for all transmembrane helices based on the distance along the predicted transmembrane helix normalized by the size of the containing helix. Above I have included 1HZXA, 2RH1A, 1OCCC, and 3QE7A (top left to right, bottom left to right). The vast majority (23 of the 25 benchmark proteins) are similar to the top two examples – well defined and aligned gradients across the structure from inner to outer membrane portions (blue to red). The only two proteins with significant errors are on the bottom (1OCCC and 3QE7A). One can see in 1OCCC that the foremost helices do not align to the expected gradient due to their inaccurate prediction as a single unbroken helix. A similar error exists in 3QE7A where a small portion is incorrect due to a missing helix break prediction.

I have recreated this filtering by focusing on both SSE and topology prediction. Both

filtering methods leverage SPOCTOPUS, which predicts transmembrane as well as inner/outer membrane coil regions (Viklund, Bernsel, Skwark, & Elofsson, 2008). For SSE filtering, each predicted SSE was assigned a sequential index value based on ordering from N to C-terminus. All amino acids within an SSE are assigned that index value and then each pair has a calculated index difference value. Pairs predicted to be within the same SSE have an index difference of zero. Pairs predicted between adjacent predicted SSEs have an index difference of one. I evaluated secondary structure separation as a filter by comparing the final accuracy average top L-fractions of L/10, L/2, and L for direct information rankings calculated from filtered MSA using the optimal set of e-values previously described (Hopf et al., 2012a). I also filtered contact predictions based on a minimum sequence separation of 6 and 12 to determine whether there was an appreciable difference in accuracy between both sets. Results are shown in Figure 18. A minimum SSE difference of zero represents no filtering and separations up to 9 were examined. A very small increase in average accuracy can only be seen for the top L/10 contacts with a minimum separation of 6. In this case, the maximum accuracy occurs at a minimum SSE difference of 2 – 40.68% compared to 39.81% with no SS-based filtering. Otherwise, maximum average accuracies across the benchmark set occur without filtering. For the remaining L-fractions with a minimum separation of 6 these accuracies are 24.34% and 17.21% for the top L/2 and L respectively. Accuracies are similar for a minimum separation of 12 although also consistently highest without SSE-based filtering - 41.08%, 24.07%, and 16.50% for the top L/10, L/2, and L contacts. This coincides with the scatterplot in Figure 20 for secondary structure index separation, as there is not a good separation of contacts from noncontacts as the separation increases.

The improvement seen is likely negated by a minimum separation of 12 as this increased separation greatly decreases the likelihood of two amino acids being within the same SSE – limiting the number of cases where SSE-based filtering could increase prediction accuracy. Improvement for this filtering method is likely trivial in part because SPOCTOPUS SSE prediction is not perfectly accurate. As such, many adjacent transmembrane helix ends, which would be predicted to be in contact and inform protein structure prediction, are incorrectly included as part of the connecting coil region. Furthermore, removing pairs within coils is not ideal as coil regions are flexible enough to contain many intra-element contacts. However, this latter issue is irrelevant when one uses BCL::Fold as coils are not included during folding. Finally, as one excludes more possible pairs based on imperfect thresholds one also requires selection for a top L-fraction set to extend further into the direct information pair ranking and thus into smaller less confident direct information pair values. The positive predictive value of direct information rapidly decreases as the number of predictions increases and must be taken into account whenever one excludes possible contact pair predictions.

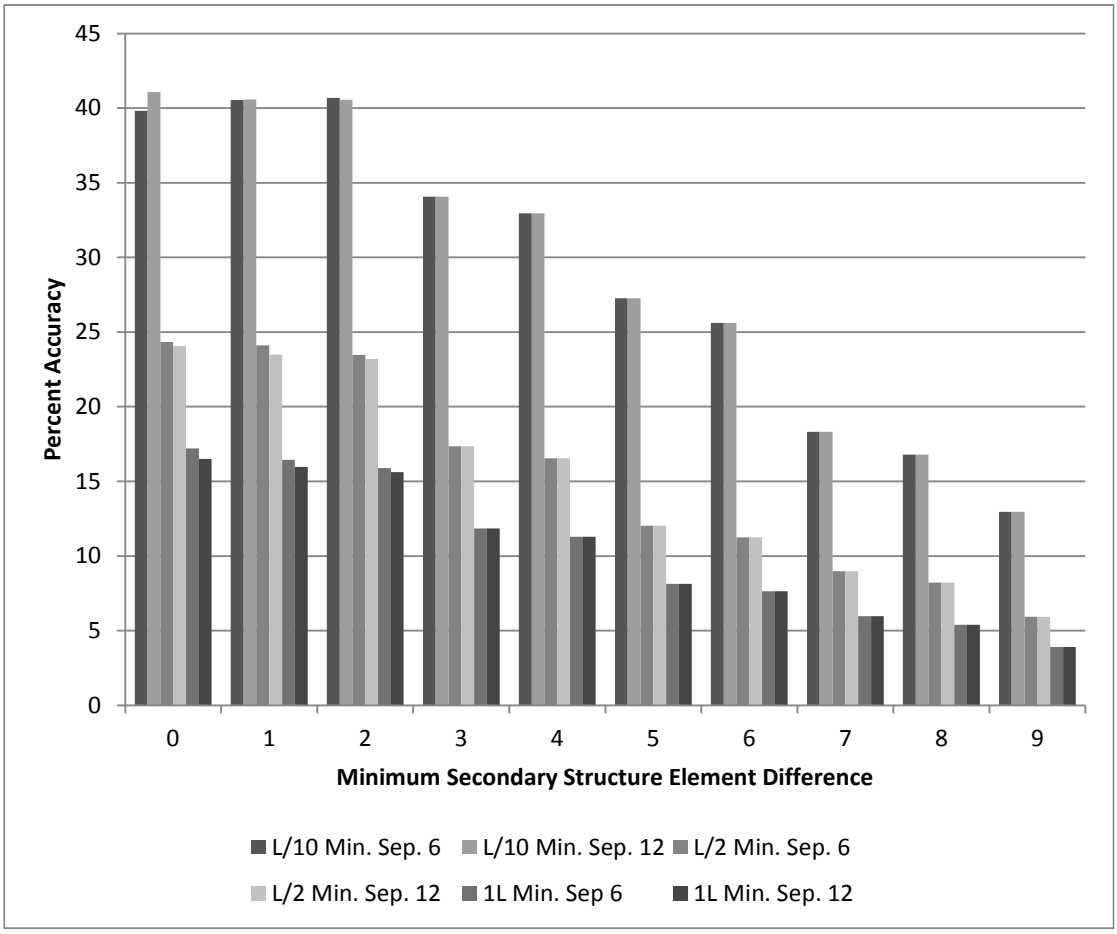


Figure 18: Filtering Based on Simple Secondary Structure Element Difference Does Not Improve Contact Prediction Accuracy

I used SPOCTOPUS to predict secondary structure and then assigned each amino acid within each SSE an index value representing the position of the SSE from N to C terminus. Amino acids within the same SSE have a minimum SSE index difference of zero. Above, each position along the x-axis indicates the minimum SSE index difference included in the top L-fraction of predicted contacts. Overall accuracy decreases as one increases the filtering threshold except for the smallest fractions of top L constraints (L/10) with a minimum separation of six. In this case, maximum average accuracy coincides with a minimum SSE difference of 2 – 40.68% compared to 39.81% with no SS-based filtering. The rest of the minimum separation 6 average accuracies are highest without SSE filtering – 24.34% and 17.21% for the top L/2 and L respectively. When one uses a minimum separation of 12 this improvement is erased. Maximum accuracy for contacts with a minimum separation of 12 are 41.08%, 24.07%, and 16.50% for the top L/10, L/2, and L contacts. A larger minimum separation for contact prediction likely decreases the number of cases where both elements in a predicted contact pair are close enough to be in the same SSE.

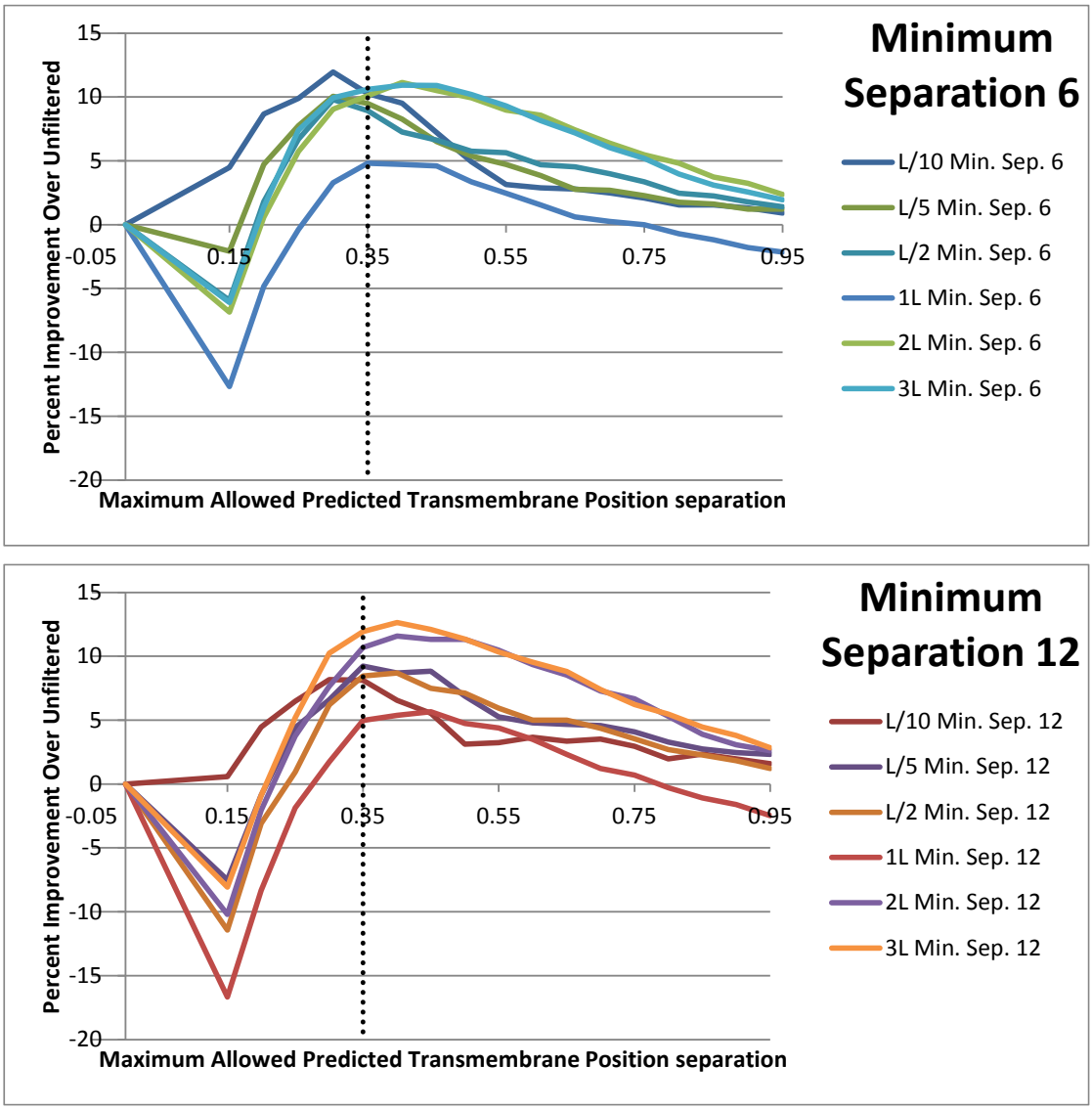
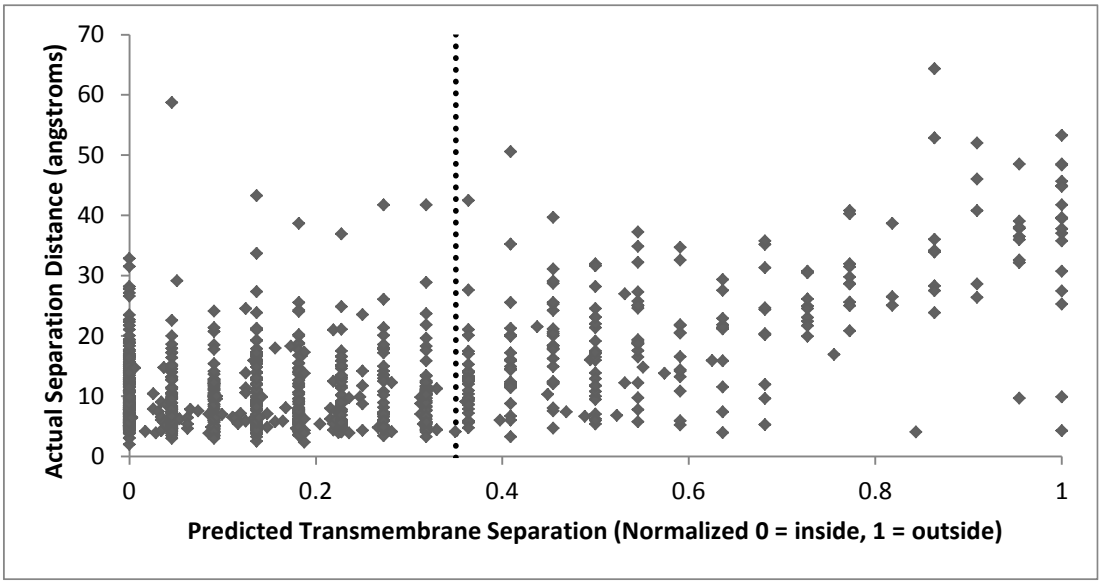


Figure 19: Predicted Transmembrane Separation Filtering Improves Contact Prediction Accuracy (Minimum Separation of 6 and 12)

I determined vertical transmembrane position using SPOCTOPUS topology predictions and then filtered contacts based on the difference between predicted transmembrane positions. Pairs predicted to be separated vertically by a significant distance are unlikely to be in contact. I calculated the optimal value to use as the filtering threshold by determining accuracies across a wide range of potential separation values and averaging the results across the entire 25 membrane protein benchmark set. Above is the percent change in accuracy across thresholds compared to the unfiltered predictions for both minimum separations of 6 and 12 across different top L fractions. The dotted line shows the threshold chosen that optimizes the improvement in accuracy across L fractions.

I also evaluated the potential of a topology-based filtering method to improve prediction

accuracy. To filter based on topology I first devised a simple normalized transmembrane position descriptor, which I have visualized and partially described in Figure 17. Positions predicted to be on the inner-membrane side as part of a coil are assigned a value of zero (consistent with a SPOCTOPUS prediction of “i”). Positions predicted to be outside of the membrane and part of a coil are assigned a value of one (consistent with a SPOCTOPUS prediction of “o”). Transmembrane helices are predicted as “M” in SPOCTOPUS and their values are determined by the position within the helix (from inner to outer membrane) and normalized based on the helix size. Given a hypothetical helix size of nineteen (nineteen “M” sites in a row in the SPOCTOPUS prediction) the first amino acid on the inner-membrane side would be assigned a value of 1/20 or 0.05. The 20 arises from the addition of 1 such that the final amino acid within this helix on the outer-membrane side would have a value of 0.95 to differentiate it from the subsequent outer-membrane coil. The few short re-entrant loops encountered were simply treated as inner or outer-membrane coils as appropriate.



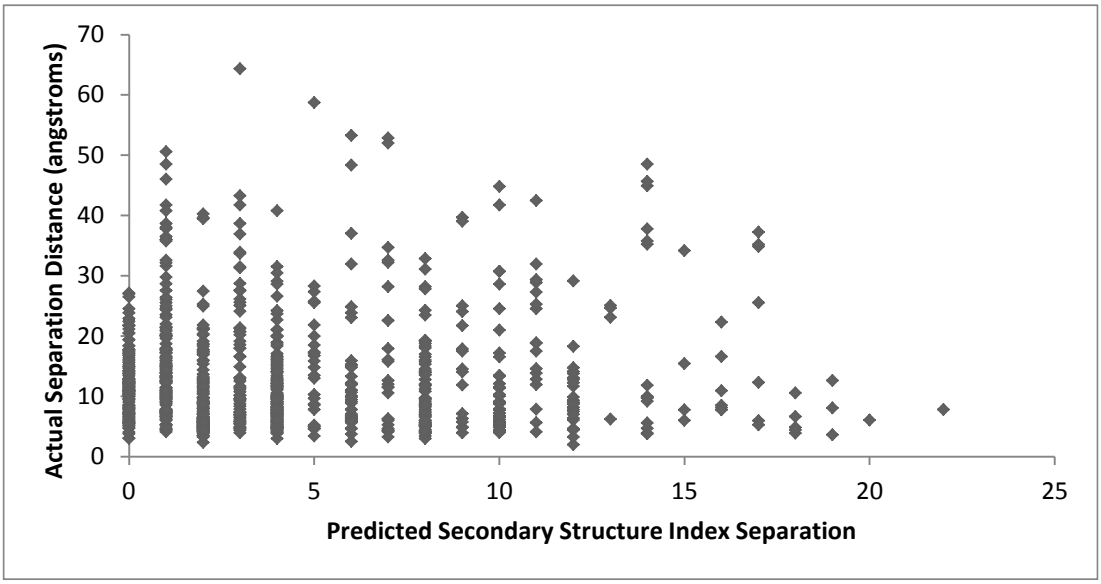


Figure 20: Scatterplots Comparing Predicted Transmembrane Position and Secondary Structure to Distance

I aggregated the top L/10 predicted contacts (using direct information calculated from filtered MSA and optimal e-values) and graphed their predicted vertical transmembrane separation/secondary structure index separation against their actual distances. A clear pattern exists for transmembrane separation where sites are much less likely to be near one another once the separation value surpasses approximately 0.35 (indicated with a vertical black dotted line). Secondary structure index separation does not separate contacts from non-contacts as well – explaining why such filtering does not result in the significant improvement seen with topology filtering.

As one can see in Figure 17, this descriptor results in a very consistent gradient along a blue-white-red spectrum (inner to outer membrane) with similar colors nearly always being very near one another with regards to their vertical membrane position. This is of course simplified by the fact that the alpha-helices within membrane proteins are perpendicularly oriented to the membrane in the vast majority of cases. Furthermore, the scatterplot in Figure 20 illustrates how distances increase for DI-based contact predictions, as there is predicted vertical separation distinguishing between contact and noncontact after the threshold of 0.35. Thus, the descriptor proves to be highly informative as amino acids cannot be in contact if they have a large difference in their vertical membrane position. It should also be noted that while two sites may have identical and accurately predicted vertical transmembrane positions they may still be on opposite sides of a protein and as such well outside the contact threshold. Furthermore, sites predicted simply to be inner or outer-membrane coils are less reliable due to the inherent flexibility of coils. Thus, position along coil SSE was not considered as it was for transmembrane helices, which further contributes to the unreliable nature of this filtering metric when solely considering positions that are inner or outer membrane. Once again, this is irrelevant to BCL::Fold, which does not include coil regions during protein fold prediction.

To determine the optimal difference threshold for filtering contacts to maximize accuracy I evaluated thresholds from zero to 0.95 for all top L-fractions and minimum separations of 6 and 12. Accuracy decreases initially for overly stringent cutoffs (near zero), which eliminate too many possible contacts for most sets of parameters. Values begin improving for most L-fractions and minimum separation values beginning around a maximum predicted transmembrane separation of 0.2. This improvement is optimal around 0.35 across L-fractions and as such, the filtering

threshold was set to a maximum predicted vertical separation of 0.35. As the threshold increases beyond 0.35, the beneficial effects of topology prediction decrease slowly towards zero. The top 1L contact predictions decreases beyond zero during this latter range likely due to a clustering of contacts with inaccurate transmembrane separation values of 1.0 due to incorrect topology predictions in the range between the top L/2 and L contacts.

Later optimization identified a top L-fraction of L/2 to be optimal for folding with BCL::Fold and improvement from filtering based on topology is 8.9% and 8.4% for minimum separations of 6 and 12 respectively. Thus, topology-based filtering significantly improves contact prediction accuracy for all L-fractions including ranges ideal for contact prediction.

Machine Learning Based Contact Prediction

DI, while promising, has many limitations. Separate topology and SSE based filtering are necessary to achieve the best results. Homomultimers and conformational changes can also confound the method. Finally, Di is calculated from a single MSA. Thus, additional information that may be present in correlation values determined from other MSA is excluded. Similarly, DI does not leverage other available information unless it is manually included via filtering. Thus, I hypothesized that a machine learning approach that incorporated correlation measures as well

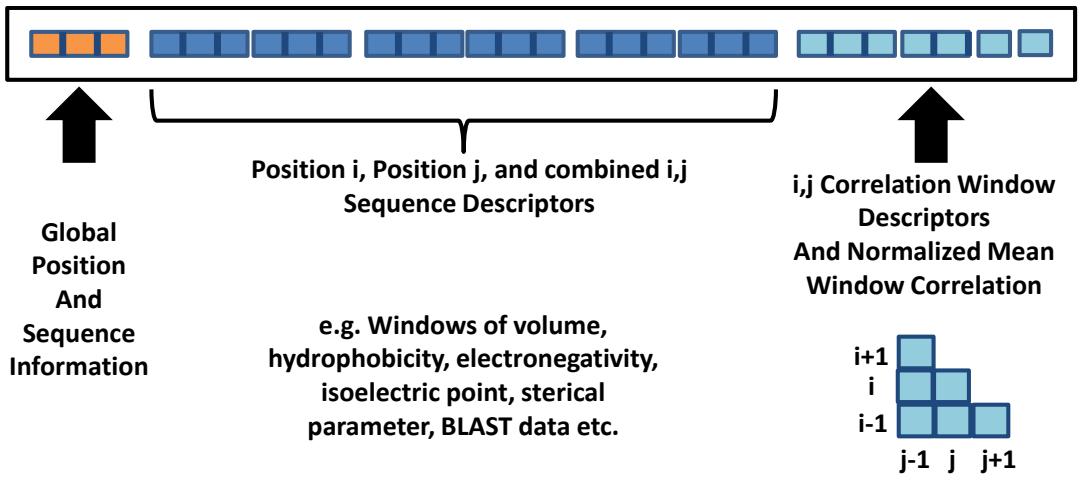


Figure 21: Diagram of Descriptors Used for Machine Learning Leveraging DI and Sequence Information

Descriptors used for model training include three categories: global, sequence information, and correlation descriptors. Global descriptors include sequence position for sites *i* and *j*, the separation from *i* to *j*, and the number of amino acids in the sequence. Sequence information descriptors include windows of biochemical properties surrounding sites *i* and *j* such as volume, hydrophobicity, sterical parameter, polarizability, isoelectric point, and their BLAST profiles. The probability of each SSE state helix/strand/coil (by JUFO9D) as well as membrane/transition/solution state are also included in sequence information. Finally, correlation information includes the symmetric matrix around sites *i* and *j*, all unique pairwise combinations from *i* ± half_window_size with *j* ± half_window_size, the mean, max, and normalized mean of this window, and the overall mean sequence correlation. All windows are set to a size of 9 for this work.

as other sequence and global position data could outperform methods relying solely on DI and a few manual filtering steps.

The descriptors assessed for model training encompass three basic categories: global, sequence information, and correlation descriptors (which indirectly result in MSA statistics that are useful for machine learning models). There were a total of 1,505 descriptors and 46,930 contacts and 1,639,181 noncontacts. I have provided a visual depiction of a descriptor vector in Figure 21. Table 2 lists all categories of descriptors. The most basic category includes global descriptors, which cover basic sequence information and element position information: sequence ID, which is indexed starting at 1, for sites *i* and *j*, the separation between *i* and *j*, and overall sequence length.

Sequence information descriptors include windows of biochemical properties surrounding sites *i* and *j*. Properties used include volume, hydrophobicity, sterical parameter, polarizability, isoelectric point, and the BLAST profiles for each site in the pair. These sequence descriptors and the aforementioned global descriptors were used previously in the Meiler lab to predict long-range contacts (Karakaş et al., 2010). I have added the probability of each SSE state helix/strand/coil (by JUFO9D) as well as membrane/transition/solution state to the set of sequence information descriptors. Prior work used an older version of the JUFO algorithm with several ANNs, each trained specifically on certain SSE type interactions to improve accuracy. I have not included that aspect of the previous approach for this thesis, as all members of this benchmark set are α-helical transmembrane proteins. In addition, I developed and included the SSE index difference and predicted vertical separation using OCTOPUS for topology prediction (covered in detail within the section on “Improving Direct Information-Based Contact Prediction

Accuracy with Predicted Topology and Secondary Structure"). Several other descriptors related to the size of the containing SSEs, the distance from the center of the SSE, and the position of each position *i* and *j* are part of the sequence descriptors.

Finally, I have included correlation information in the form of various aggregated sets of DI values. Correlation information was included from the unfiltered and filtered MSAs using the optimal set of e-values, as well as from sets of all filtered, all unfiltered, and both sets across each e-value. These sets were included via windows around the position *i* and *j* as well as by taking the maximum, mean, normalized mean, standard deviation, and sum across windows or the entire protein sequence. Windows refer to the lower triangle of the surrounding symmetric matrix for sites *i* and *j*. In other words, this set includes all unique pairwise combinations for *i* ± half_the_window_size with *j* ± half_the_window_size. All windows are set at a size of nine. The normalized mean calculation reduces the impact of variations in DI between proteins by dividing the mean DI for a symmetric matrix by the mean DI across the entire protein sequence (Equation 2. Evaluations of the descriptors using information gain, F-score, and input sensitivity all suggested that the normalized mean is one of the most useful correlation descriptors for contact prediction. This is likely due to the reduction in noise due to normalization as well as the fact that if a pair *i,j* has neighbors with high DI values, *i,j* are also more likely to be in contact. Examining a window of width nine appears to capture this relationship well – chosen because it completely encompasses two complete turns of an alpha helix. There are 7.2 amino acids, or eight after rounded up. One is added to achieve an odd number such that amino acids to be predicted are centered in the window.

$$NM(i,j) = \frac{\sum_{x=i-k}^{i+k} \sum_{y=j-k}^{j+k} \frac{\boldsymbol{DI}(\boldsymbol{x},\boldsymbol{y})}{\boldsymbol{2}}}{\left(\frac{\sum_{z=1}^{L} \boldsymbol{DI}(\boldsymbol{z})}{\boldsymbol{L}}\right)} \quad (2.)$$

Equation 2: Normalized Mean Calculation for the Given Correlation Window Surrounding Position *i,j*

Normalized mean (NM) for *i,j* is determined by calculating the average DI for the symmetric matrix surrounding *i,j* and dividing that by the average DI across the entire protein sequence. *L* is the length of the protein sequence and *k* represents half the window size desired rounded down. Window size was set to nine for this study.

Statistics derived from the many MSA alignments created are also included within the correlation descriptors. These are the length of the aligned target sequence including gaps, the depth of the alignments, the effective depths of the alignments (M_{eff}), which adjusts for sequence redundancy within the alignment, and the coverage of target sequence for the given MSA (percent of the columns with fewer than 30% gaps).

The number of descriptors balloons to 1,505 when all the categories above are used with all permutations (windows, aggregations of windows, different MSA parameters, applied to each property etc.). To optimize performance for the models chosen, it was necessary to reduce the number of descriptors significantly. Many were not beneficial for contact prediction and added noise and increased computational time. Ideally, one would use backwards elimination or forward feature selection one descriptor at a time. However, due to computational constraints it was necessary to filter descriptors using a modified method. I scored descriptors using information gain and F-score to determine their individual potential for contact prediction. One can see the F-score for all descriptors along with a focused view of the top ten in Figure 22. F-score rapidly drops

off from a maximum of 0.966 for predicted transmembrane separation and is nearly zero by the 217th descriptor. I used this descriptor ranking to train decision tree models using the top N descriptors. I evaluated performance with AUC initially. The smallest number of N descriptors was 10 and increased in increments of 50 descriptors to cover all descriptors. At each threshold I perform a fivefold cross validation and average the predictions across all generated models. During descriptor and parameter optimization I allowed data points from a single protein to span across training and monitoring or monitoring and independent datasets. However, at no point was any data point included within multiple datasets.

Table 2: Categories of Global, Sequence, and Direct Information (Correlation) Descriptors

The table above contains all the categories of descriptors initially analyzed. They are divided into the three broad categories. The first is global position descriptors - the location of each element of the pair i,j being predicted within the context of the sequence. The second category is the sequence descriptors – biochemical, BLAST, and predicted secondary structure information regarding the amino acids as well as aggregated descriptors (mean and standard deviation) calculated across the entire sequence for the given properties and BLAST data. Finally, the third lists correlation descriptors, which includes various aggregated descriptors such as the max, mean, sequence normalized mean, standard deviation, and sum across collections of the elemental correlation descriptors (by window or across e-value/filtering parameters).

AUC does not ideally reflect the end goal of predicting the top L-fraction of cutoffs. However, AUC seemed to be a more stable objective function as it is cutoff agnostic and thus seemed more appropriate for initial descriptor and parameter optimization. Performance plateaued after the top 210 descriptors and a calculation of input sensitivity across the 20 generated models is essentially zero (numerical artifact) above 210. This indicates that the

Global Position Descriptors

Sequence IDs for Positions i and j	Distance from Beginning of Sequence to i
Distance from i to j	Distance from j to End of Sequence

Sequence Descriptors

Properties	**Predicted Secondary Structure Descriptors**
Sterical Parameter	SSE Size
Polarizability	Position in SSE
Hydrophobicity	Distance from SSE Center
IsoelectricPoint	Predicted Position in Membrane
Volume	SSE Index Difference
Helix Probability	Predicted Vertical Membrane Separation (Topology-Based)
Strand Probability	JUFO Helix/Strand/Coil Probabilities
Free Energy Helix	JUFO9D Membrane Transition Solution Probabilities
Free Energy Coil	JUFO9D Membrane and SSE Probabilities
Transfer Free Energy Punta-Maritan 3D	OCTOPUS Membrane Transition Solution Probabilities
Free Energy Core	
Free Energy Transition	**Whole Sequence Descriptors (Applied to All Properties and BLAST Descriptors)**
Free Energy Solution	Sequence Mean
Free Energy Core Helix	Sequence Std. Deviation
Free Energy Transition Helix	
Free Energy Solution Helix	**Window Types (Applied to All Properties)**
Free Energy Core Coil	Window
Free Energy Transition Coil	Window Average
Free Energy Solution Coil	Window Standard Deviation
BLAST Descriptors	Window with Properties Weighted by BLAST Log Probability
BLAST Profile	Sequence Mean across All Properies
BLAST Conservation	Sequence Std. Deviation across All Properies

Direct Information Descriptors

Correlation Groups	**Window Types (Applied to All Correlation Groups)**
Filtered MSA Using Opt. E-values	Window Max
Unfiltered MSA Using Opt. E-values	Window Mean
Across All Filtered MSA (1E-03, 1E-05, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40)	Window Normalized Mean
Across All Unfiltered MSA (1E-03, 1E-05, 1E-10,	Window Std. Deviation

1E-15, 1E-20, 1E-30, and 1E-40) Across All Filtered and Unfiltered MSA (1E-03, 1E-05, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40)	Window Sum
MSA Statistics Descriptors MSA Length Effective MSA Depth (M_{eff}) MSA Depth (M) Target Sequence Coverage	

decision tree models found little additional relevant information above that threshold. As such, I only included the top 210 descriptors by F-score for further descriptor selection.

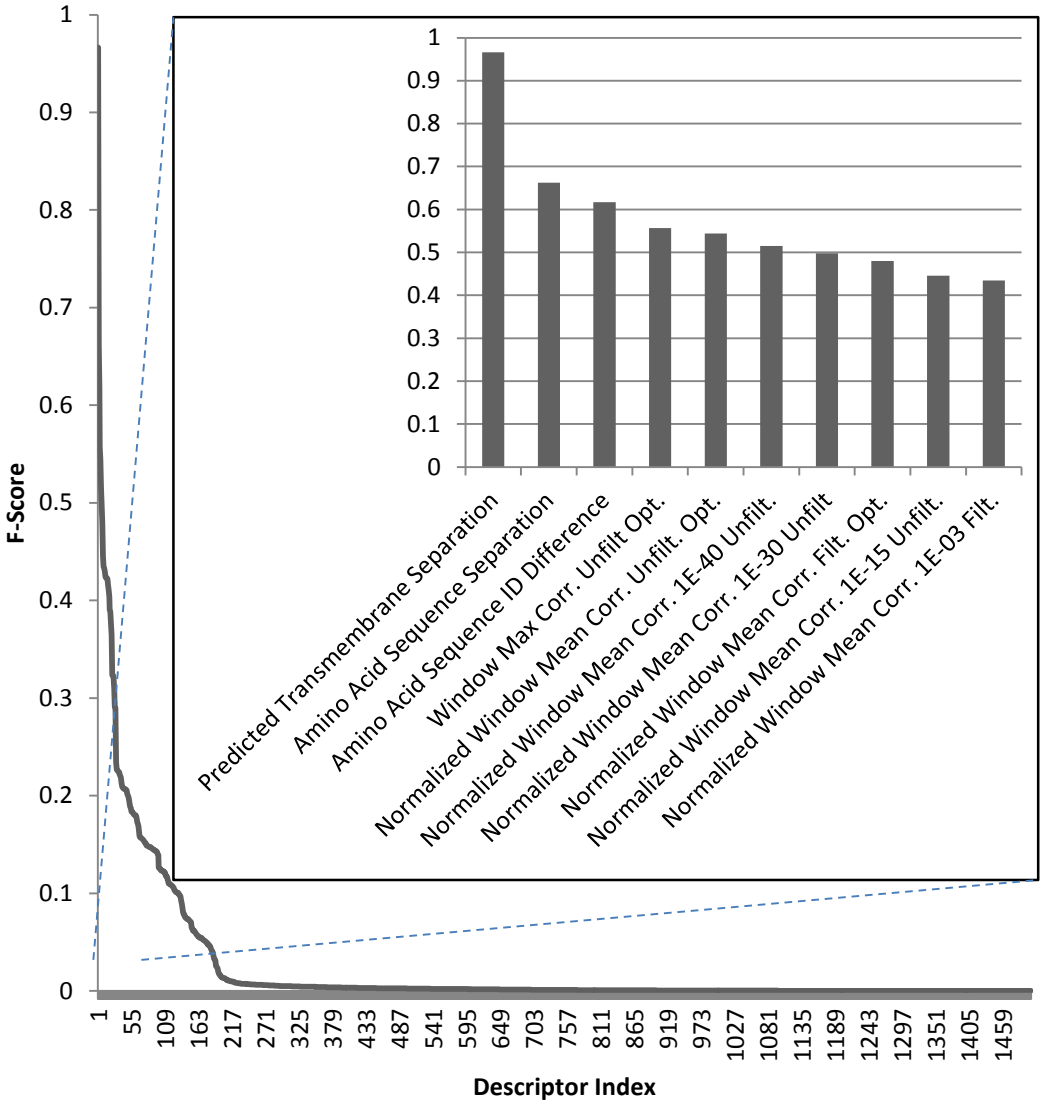


Figure 22: F-Score across all Descriptors and the Top Ten Descriptors

I ranked all 1505 descriptors by F-score and graphed those results above. In addition, I have highlighted the top 10 descriptors by F-score in the column chart above. The most useful descriptor by F-score is the predicted transmembrane separation (0.966), followed by sequence separation and amino acid sequence ID difference (which are essentially the same). The remaining elements among the top ten are correlation descriptors. The maximum of the current position window scores very highly - followed by the mean of a position's correlation window. As expected, predictions using the optimized set of e-values perform very well although surprisingly the unfiltered MSA with optimized e-values score slightly higher than the descriptors from filtered MSA. F-score drops very rapidly across the set and descriptors after approximately the first 217 have near zero scores. However, these cannot be discounted as F-score does not identify descriptors that are only valuable when combined with others.

I further reduced the number of descriptors from among the top 210 iteratively. The best scoring threshold was used to determine the next set of descriptors to be kept. I then ranked this subset using input sensitivity – calculated using the models created at this new threshold. This new ranked set of descriptors was then divided using multiple thresholds, models trained at each, and each set scored to continue the iterative descriptor selection process. To decrease the likelihood of removing useful descriptors, thresholds never eliminated more than half of the descriptors before re-ranking descriptors. I also used enrichment average as the objective function and evaluated each set of models generated by calculating the integral of the precision over the range 0.01% to 0.55% of the fraction predicted positive. This range closely captures the contacts predicted when taking the top 1L predictions across all proteins while decreasing the noise present below 0.01%. The modest number of data points results in drastic changes from small perturbations in overall predictions.

Figure 24 displays the results for input sensitivity scored for iterations with the top 160, 130, and 70 descriptors. For 160 descriptors, there is a plateau beginning at 30 descriptors and another increase in performance near 130 resulting in a second plateau. To evaluate more carefully this second plateau, I recalculated input sensitivity using the models generated with 130 descriptors and repeated the evaluation. Result variability decreases significantly after rescoring and is relatively consistent above 30. I further examined this transition point by rescoring using models produced with the top 70 descriptors from the current round and decreased the increment step size to three descriptors. In this final round, the same pattern is seen – a consistent

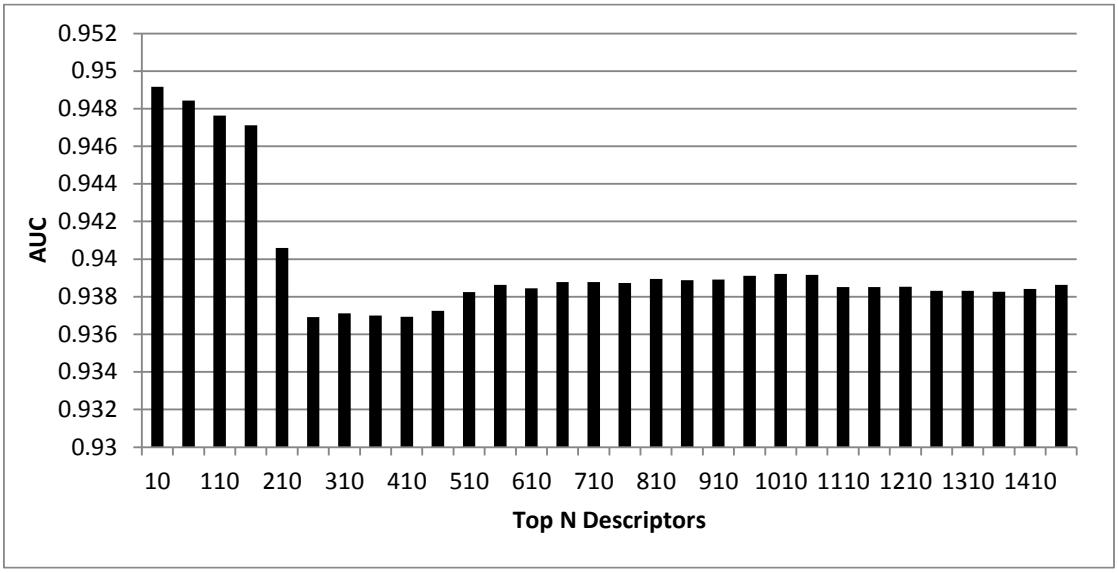


Figure 23: Performance (AUC) across Top Descriptors Ranked by F-Score

After ranking all 1505 descriptors, I used increasing thresholds of the top N descriptors to predict contacts with decision trees. Each set of the top N descriptors was used within a fivefold cross validation and the results for each set are given above. Maximum AUC is 0.949 using only the top 10 descriptors. Performance decreases significantly from the top 210 and up. Examining the input sensitivity calculated, one sees that scores above that point are numerical artifacts and thus essentially zero. Descriptors ranked that poorly by F-score were not used in further decision tree training due to computational resource limitations. This step essentially filtered down the initial descriptor set. It is important to note that AUC does not perfectly mirror the final desired performance measure.

plateau beginning around the top 30 descriptors. As such, I selected these top 30 descriptors as the optimal for training the final decision tree models.

Table 3 lists all 30 descriptors, and includes a large number of correlation-based descriptors. The highest ranked descriptor is the correlation window maximum using filtered MSA created with the set of optimal e-values. This set of DI values also performs best for naïve DI contact prediction. The second highest by input sensitivity is the sequence separation, which intuitively fits as knowing this descriptor alone provides a great deal of information regarding whether two sites are in contact. Sites adjacent in sequence are almost certainly in contact and sites at disparate positions in sequence are very unlikely to be in contact. The third highest descriptor is predicted transmembrane separation, which is derived from the topology-based filtering leveraged by Hopf *et al.* to achieve such high accuracy predictions. The normalized window mean and maximum correlation from the unfiltered optimized are also ranked very highly. Polarizability is the first traditional biochemical property to show up in the descriptor ranking but even in this case it is an aggregated sequence mean and not just the polarizability for positions *i* and/or *j*.

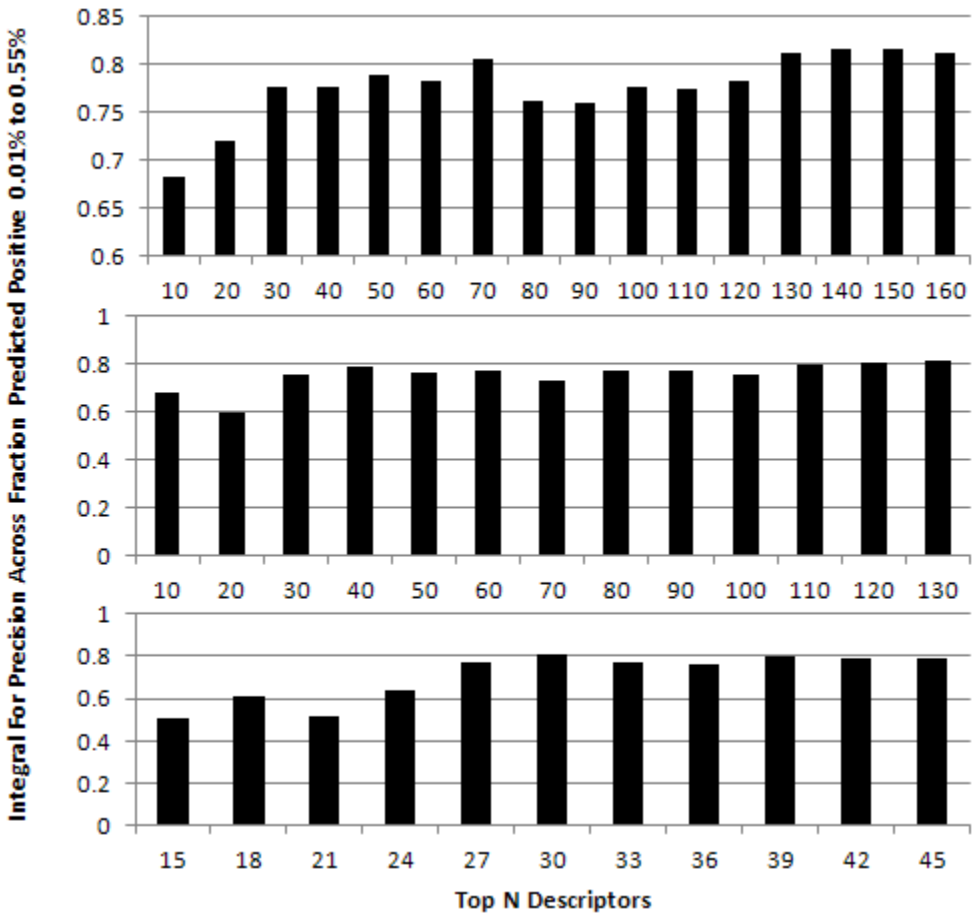


Figure 24: Integral of Precision Over Fraction Predicted Positive from 0.01% to 0.55% for Input Sensitivity Iterations Using Decision Trees

Above are the average of integrals of the positive predictive value from 0.01% to 0.55% of the fraction predicted positive from fivefold cross validated models. Panels A, B, and C cover iterations done using input sensitivity scores to rank the descriptors before taking increasing thresholds of the top descriptors. The first 210 were determined using F-score generating 20 models via a fivefold cross validation and using those models to calculate input sensitivity. Panel A shows two promising regions beginning around the top 30 and 130 descriptors. Panel B shows a large decrease in variability after rescoring the top 130 descriptors with the plateau beginning around the top 30 and upwards. Finally, the focused set above of the results, scoring the top 70, shows an overall peak and the beginning of a similar plateau at the top 30 descriptors. I used this set of 30 descriptors to train the final decision tree models for contact prediction.

Overall, decision trees favor correlation descriptors very strongly. Only 8 out of the 30 descriptors are sequence descriptors. Global position descriptors occupy 3 of the 30 and

correlation descriptors comprise the remaining 19 positions. This once again suggests that correlation descriptors, especially those aggregated or processed in some way (window max, normalized mean, or sequence mean) are a beneficial addition to the input for machine learning methods predicting protein contacts.

Table 3: Table of Top 30 Descriptors Used for Best Decision Tree Model

Descriptor Name	Input Sensitivity Score	Type	Rank
Window Maximum Correlation Filtered [Optimized]	0.02013	Correlation	1
Amino Acid Sequence Separation	0.01601	Sequence	2
Predicted Transmembrane Separation	0.01027	Topology	3
Normalized Window Mean Correlation Unfiltered [Optimized]	0.00604	Correlation	4
Window Max Correlation Unfiltered [Optimized]	0.00545	Correlation	5
Sequence Mean Polarizability	0.00298	Sequence	6
Normalized Window Mean Correlation E-value 1E-10 Unfiltered MSA	0.00193	Correlation	7
SequenceMean(Correlation(Filtered [Optimized]	0.00179	Correlation	8
Position i Sequence Position	0.00172	Sequence	9
Coverage for E-value 1E-10 Filtered MSA	0.00127	MSA Statistics	10
Probability (JUFO9D) Amino Acid j is a Transition Region Coil	0.00121	Sequence	11
Normalized Mean Window Correlation E-value 1E-40 Unfiltered MSA	0.00119	Correlation	12
Max Correlation All Filtered and Unfiltered MSA	0.00118	Correlation	13
Normalized Mean Correlation Filtered [Optimized]	0.00074	Correlation	14
Correlation Unfiltered [Optimized] Center Position	0.00072	Correlation	15
Normalized Window Mean Correlation E-value 1E-30 Unfiltered MSA	0.00067	Correlation	16
Sequence Length (L)	0.00065	Sequence	17
Correlation Filtered [Optimized] Center Position	0.00049	Correlation	18
Amino Acid Sequence ID Difference	0.00046	Global Position	19
Distance to End of Sequence	0.00040	Global Position	20
Sequence Mean IsoelectricPoint	0.00036	Sequence	21
Meff for E-value 1E-03 Filtered MSA	0.00035	MSA Statistics	22
Window Average of Blast Log Weighted Free Energy Transition Coil with Triangular Weighting at Index 7	0.00034	Sequence	23
Correlation E-value 1E-10 Unfiltered MSA	0.00031	Correlation	24
Mean Correlation All Filtered MSA	0.00031	Correlation	25
Probability (JUFO9D) Amino Acid I is in a Helix	0.00030	Sequence	26
Mean Correlation All Filtered and Unfiltered MSA	0.00030	Correlation	27
Normalized Window Mean Correlation E-value 1E-40 Filtered MSA	0.00028	Correlation	28
Correlation E-value 1E-03 Unfiltered MSA	0.00027	Correlation	29
Distance from Beginning of Sequence to Position i	0.00026	Global Position	30

Above is a table listing the final set of 30 descriptors selected for use with decision trees to predict long-range contacts. I determined this set using an iterative process whereby I scored

the usefulness of descriptors for a given model type initially using F-score and then input sensitivity. I evaluated models using increasing subsets of the ranked descriptors. I then selected optimal thresholds and repeated the scoring and evaluation process until improvement plateaued or decreased. The top 30 descriptors are described above along with the type of descriptor and the input sensitivity at the last iteration.

While decision trees perform well, I have also leveraged ANNs to predict protein contacts. Decision trees were used first as they require much less time to train and have fewer parameters to optimize. However, I believed that ANNs could outperform decision trees and bolster my hypothesis that machine learning can predict contacts more accurately than using direct information alone. The ANN used contained a single hidden layer of eight hidden nodes with a single input and a single output node. I determined some reasonable learning rate (eta) and momentum (alpha) parameters by using a grid search across a broad range of values. Figure 25 contains a heatmap of the resulting AUC values calculated for each pairing of the eta and alpha values examined. I determined each AUC by averaging the predictions across all 20 models generated during a fivefold cross validation with the given parameters. Each ANN trained on a balanced dataset of contacts and noncontacts with RMSD as the enrichment function, as it is more stable than enrichment, and a single hidden layer with eight nodes. An eta of 0.000017 delivered peak AUCs and alpha made little difference. In retrospect this was due to the step size of one, which negates the effect of alpha. I set alpha to 0.2 initially but changed it to 0.0, as the step size used was one.

		Alpha					
	Top 300 Features	**0.002**	**0.02**	**0.1**	**0.2**	**0.3**	**0.5**
Eta	**0.00000017**	0.589676	0.589676	0.589676	0.589676	0.589676	0.589676
	0.0000017	0.932091	0.932117	0.931931	0.93269	0.932742	0.931962
	0.000017	0.9301	0.932308	0.930813	0.932449	0.927344	0.924182
	0.00017	0.846374	0.844601	0.882601	0.864614	0.872312	0.880432
	0.0017	0.714922	0.674375	0.595411	0.636499	0.624858	0.589676
	0.017	0.589676	0.589676	0.606709	0.589676	0.589676	0.589676
	Top 500 Features	**0.002**	**0.02**	**0.1**	**0.2**	**0.3**	**0.5**
Eta	**0.00000017**	0.537884	0.535901	0.537075	0.546484	0.55679	0.580362
	0.0000017	0.929841	0.929821	0.930257	0.930451	0.93052	0.930616
	0.000017	0.924989	0.923166	0.929511	0.931659	0.919764	0.919941
	0.00017	0.793821	0.779113	0.81613	0.825275	0.844451	0.80905
	0.0017	0.5904	0.679239	0.636572	0.617231	0.585773	0.4917
	0.017	0.503165	0.504108	0.493002	0.553126	0.504553	0.4917

Figure 25: Initial ANN Alpha and Eta Grid Search Heatmap (AUC)

I evaluated various eta and alpha values using a grid search across pairings, to determine reasonable initial parameters for descriptor optimization and training. The results, average AUC values for the fivefold cross validation run for each pair, are depicted above. Step size was only 1 - explaining the minimal effects of varying alpha. For final training I set alpha to 0.0 and alpha to 0.000017.

Using the parameters selected above, I used a similar input sensitivity based iteration method to determine the optimal set of descriptors for ANNs but the best results were achieved using an iterative method that examines the weights between nodes with ANNs trained on the descriptors to be optimized. Jeffrey Mendenhall from the Meiler Lab developed the method as part of the BCL. At its core, the method utilizes the weights between the nodes of an ANN to calculate an approximate derivative for each descriptor. For the weight matrix between layer x and y (M_{xy}), this method computes the product of the transpose of M_{xy} for each model given. The result of that product of matrices is the approximate partial derivative of the result dependent on the feature in question. This method then scores the descriptors using two previously implemented statistical measures. The first evaluates the consistency, whether the descriptor

tends to increase or decrease the likelihood of a contact across all models given. The second is the average (pseudo) derivative squared. Each is then rescaled between 0 and 0.5, summed, and squared. Descriptors that are not generally useful should show little consistency and have small weights, so the outcome of this measure will range between 0 for non-general descriptors, and 1 for descriptors that are broadly useful. I have calculated the AUC and the integral of the precision from the previously specified range for each set of models and have graphed the results across all 29 iterations in Figure 26. The AUC slowly trends upwards across the entire set of runs with a peak at the 27th iteration, which uses 94 descriptors. However, using the integral results, performance stays range-bound around 0.4 until the 23rd iteration at which point there is a jump up to approximately 0.569 followed by a slow decline over the next two runs before returning to similar performance as the initial iterations interspersed with a few nearly equivalent runs. For final ANN training I selected this 23rd round as it had both the highest value as well as several subsequent iterations with promising results. Thus, I selected these 146 descriptors for final ANN training.

I trained decision trees and ANNs with a very similar protocol once I determined the optimal parameters and descriptor sets. Training, monitoring, and independent (prediction) data sets were created by randomizing proteins and then selecting without replacement 15, 5 and 5 for each set respectively. Once the independent set was selected, I performed five iterations where at each one the proteins within only the training and monitoring datasets were shuffled. I averaged the results from the five models produced to create the final contact predictions for the five proteins within the independent dataset. At no point was data from the independent proteins shown to the models during training. Datasets were unbalanced between contacts and noncontacts for decision trees but were balanced for ANNs. All predictions were combined to

generate the graphs of AUC and the integral for precision representing the final model performance depicted in Figure 27 and Figure 28.

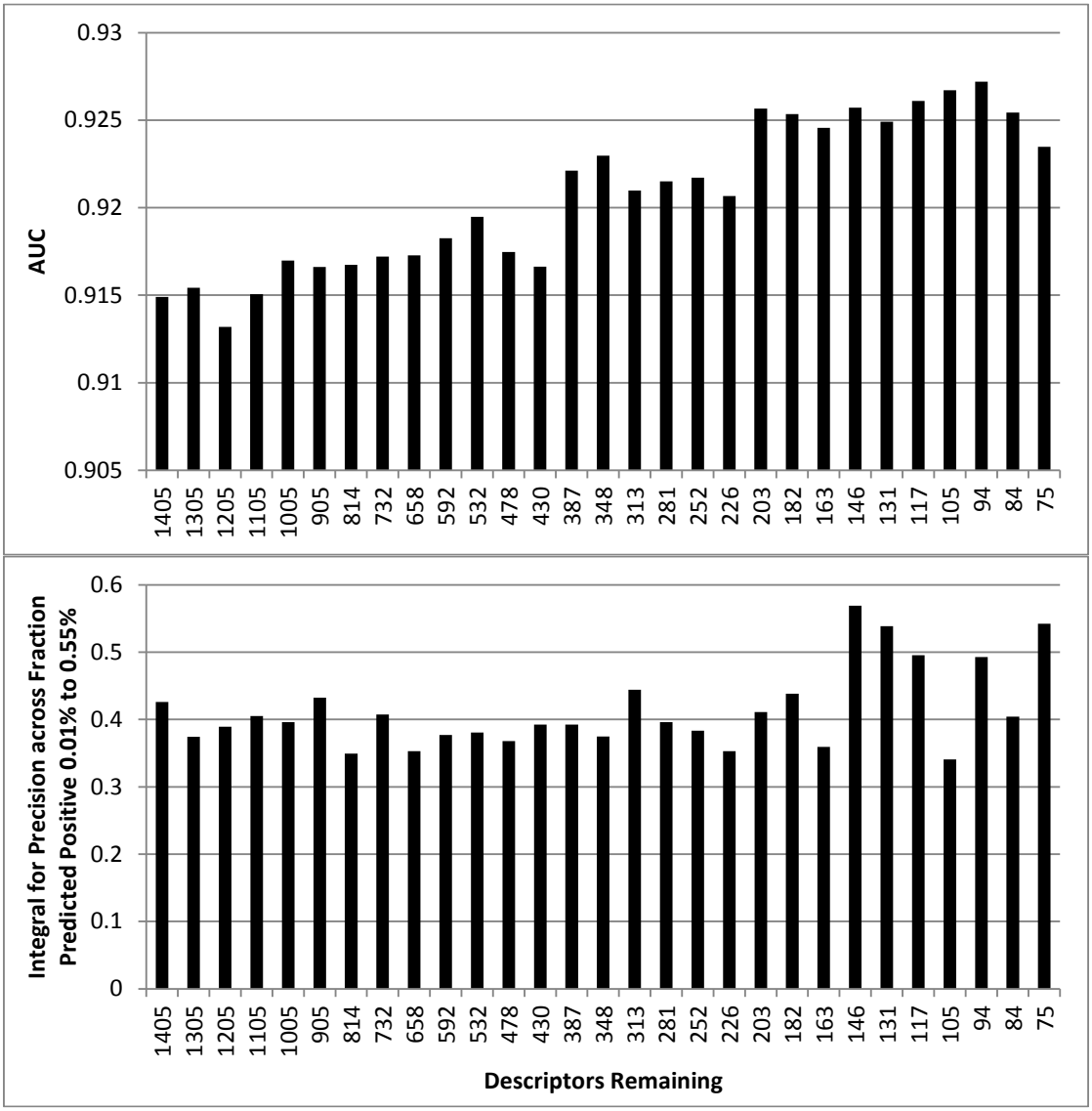


Figure 26: AUC and Integral for Precision across Fraction Predicted Positive After All 29 Rounds of ANN Weights-Based Optimization

In addition to the input sensitivity iteration method used with decision trees I also attempted a descriptor optimization that determines which descriptors are most useful by analyzing the weights of the ANN models trained. The results of a 29 round optimization are graphed above for both the AUC and the integral of the positive predictive value across the fraction predicted positive from 0.01% to 0.55%. The AUC trends upwards slowly as descriptors are removed. There is a slight plateau once one reaches 203 descriptors. However, the positive predictive value integral is largely steady within a range until one reaches 146 descriptors. This is the highest point, followed by several higher but declining values as one approaches the final round of optimization. Given that the positive predictive value integral is more representative of the top L contact predictions desired for protein fold prediction, I used the top 146 descriptors for final contact prediction (round 23).

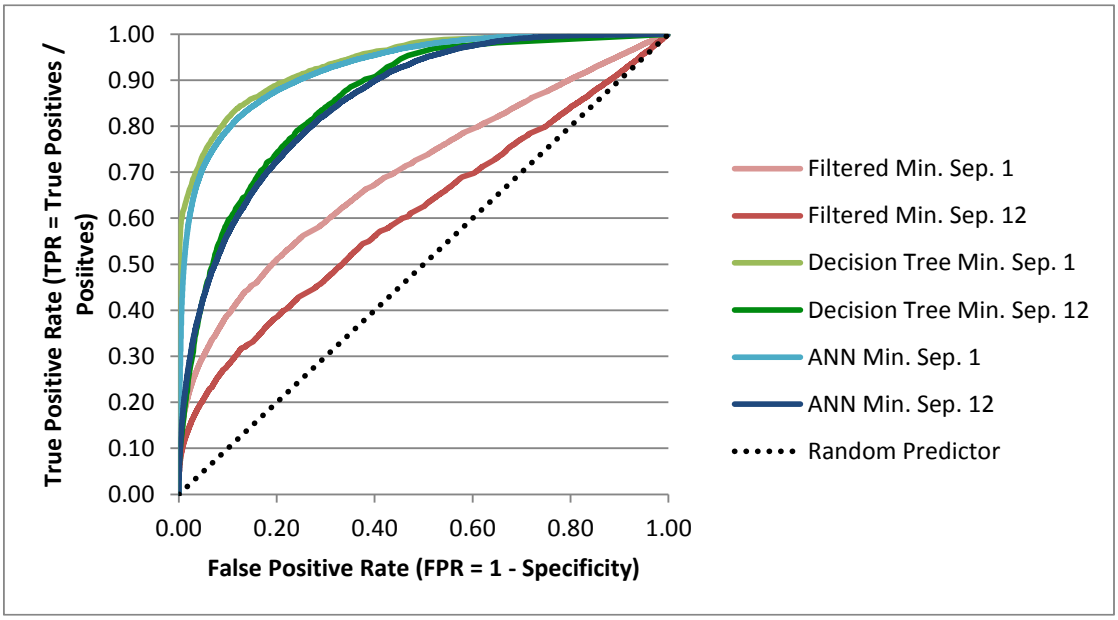


Figure 27: Best Decision Tree and ANN Contact Prediction ROC Curve Compared to Naïve Direct Information for All Pairs with a Minimum Separation of 1 and 12

The above contains the ROC curves of the merged predictions averaged from five different training iterations for each protein. Above I have also included results from contact prediction based solely on naïve direct information (using the optimal filtered MSA) for comparison. The training, monitoring, and independents set included data from 15, 5, and 5 proteins respectively. The independent predictions are the ones presented above. AUC is approximately 0.7000.700, 0.938, and 0.928 at a minimum separation of 1 for the filtered direct information, decision tree, and ANN methods respectively. For a minimum separation of 12 the AUC is approximately 0.611, 0.862, and 0.855 for the filtered DI, decision tree, and ANN methods respectively. A minimums separation of 6 performs very similarly to a minimum separation of 12 (not shown). Both methods significantly outperform naïve DI with a slight edge for decision trees. Performance is especially impressive for cases with a minimum separation of 12 given the difficulty of such predictions.

Figure 27 displays the ROC curves generated for decision trees (light and dark green) and ANNs (light and dark blue) for their predictions across the entire benchmark set in comparison to the results from naïve DI using filtered MSA (light and dark red) at minimum separations of 1 and 12. The lighter colors indicate the more trivial minimum separation of 1 and the darker colors distinguish the task of long-range contact prediction (minimum separation of 12). Most notably, the AUC is drastically higher for both machine learning methods in comparison to naïve DI.

Performance for either decision trees or ANNs on the harder long-range contact prediction task is still significantly higher than naïve DI predicting for contacts using a minimum separation of 1. AUC shows overall prediction, however the goal is to predict approximately L contacts for each protein with especially high accuracy. To evaluate these predictions across the dataset in a manner that more closely reflects the final prediction task, I have also graphed positive predictive value on a logarithmic scale focusing in on the most confident predictions (Figure 28). Once again, both decision trees and ANNs outperform naïve DI for the key range of 0.01% to 0.55%. Decision trees perform best for a minimum separation of 1 and ANNs perform best at the more important minimum separation of 12. The especially high performance for decision trees on closer range contacts may not easily translate to useful results for protein folding but may be of use for closer range protein structure prediction problems, such as secondary structure prediction.

Finally, I evaluated the accuracy for the entire benchmark set and averaged the final accuracies across all top L fractions used previously for this work. I have included a comparison for average benchmark set accuracy at a minimum separation of 6 between naïve DI using filtered and unfiltered MSA, processed filtered DI, and the best decision tree and ANN sets (Figure 29). Decision trees outperform all DI-only methods for all L-fractions L/5 and higher. ANNs outperform all DI-only methods for L-fractions of L/2 and higher. ANNs actually outperform all methods, including decision trees, by a significant margin for L-fractions of 1L and higher. L-fractions of L-2 and higher result in the best protein structure prediction performance using the BCL as is discussed in the section on “Protein Structure Prediction Using BCL::Fold and Contact Restraints”. Thus, machine learning methods achieve the best average benchmark accuracies for L-fractions most suitable for protein fold prediction using BCL::Fold. This suggests that machine learning

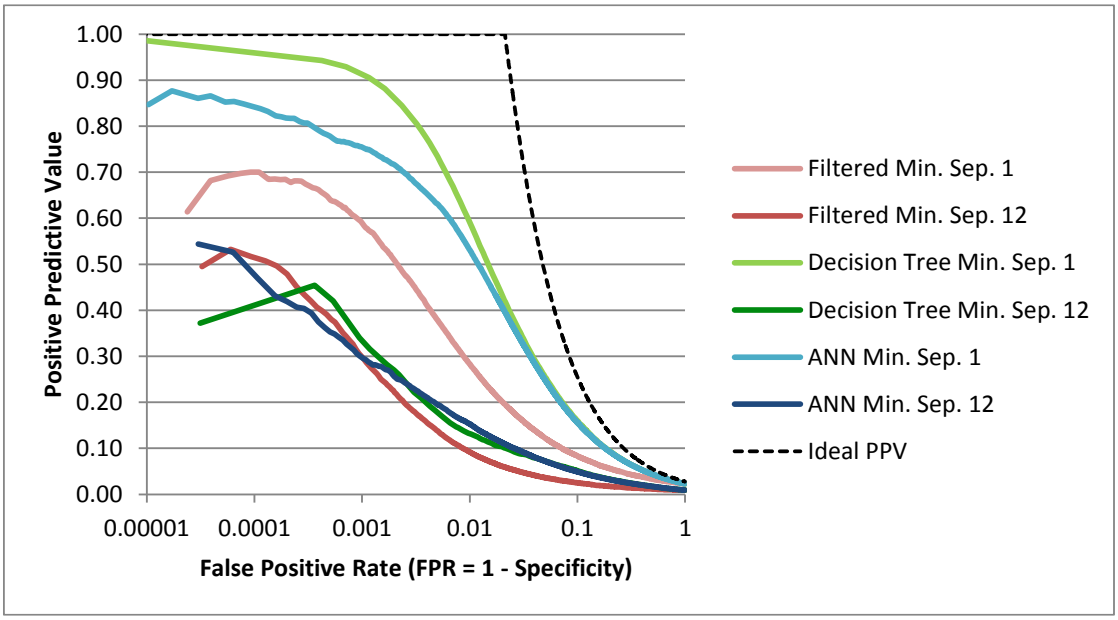


Figure 28: Best Decision Tree and ANN Prediction Logarithmic Precision vs. Fraction Positive Predicted (FPP) Values Compared to Naïve Direct Information for All Site-Site Pairs with a Minimum Separation of 1 and 12

The above contains a graph showing precision as the fraction predicted positive increases. Each curve includes the aggregated predicted contacts from five training iterations using decision trees or ANNs. Models were trained using all contacts with a minimum separation of 1 and were tested on pairs with a minimum separation of 12. Each iteration uses 15, 5, and 5 proteins for the training, monitoring, and hold-out sets respectively. The integral of the precision from 0.01% to 0.55% is approximately 0.656, 1.921, and 0.865 at a minimum separation of 1 for direct information, decision tree, and ANN based methods respectively. At a minimum separation of 12 the integral is approximately 0.537, 0.469, and 0.549 for direct information, decision tree, and ANN based methods respectively. A minimums separation of 6 performs very similarly to a minimum separation of 12 (not shown). Greater precision initially and continuing out as FPP increases is better. The black line depicts ideal performance.

methods that incorporate direct information in addition to other global position as well as sequence descriptors are a more accurate method for predicting contacts. Table 4 contains aggregated benchmark set contact prediction accuracies across the best results for naïve DI, processed and filtered DI, the best decision tree, and the best ANN model. The best accuracy for each PDBID and L-fraction is bolded. The table clearly shows that decision trees outperform other methods at lower L-fractions () and ANNs are able to predict larger numbers of contacts more

accurately than any other method across the entire benchmark set. ANNs have the highest accuracy at 3L and a minimum separation of 12 for 19 of the 25 benchmark proteins.

Table 4: Table of Aggregated Contact Prediction Accuracies from Best Methods Across Categories and Optimal L-fractions for Each Included Method

Part I

	DI Filtered MSA Stats				DI Naïve Accuracy (%)				DI Processed Filt Accuracy (%)			
PDBID	**L**	**TM_{helix}**	**M_{eff}**	**Cov**	**L/10 ms 12**	**L/2 ms 6**	**1L ms 12**	**3L ms 12**	**L/10 ms 12**	**L/2 ms 6**	**1L ms 12**	**3L ms 12**
3NCYA	422	12	1205	0.891	42.2	25.1	18.7	9.3	**44.4**	26.5	**20.9**	10.0
2RH1A	442	8	451	0.652	18.9	9.7	6.7	3.6	**21.6**	14.0	8.9	5.1
1OKCA	292	6	1043	0.89	**43.3**	**34.2**	19.5	10.2	36.7	**34.2**	**22.2**	10.9
1XQFA	362	11	1021	0.961	28.6	23.9	15.3	7.3	52.4	28.2	17.7	8.3
3GD8A	223	7	1062	0.964	90.9	**59.8**	39.5	18.5	90.9	53.6	36.3	18.4
1L7VA	324	10	1045	0.914	0.0	1.8	1.2	1.4	0.0	3.1	3.1	2.3
3MKTA	460	12	1068	0.917	**52.2**	32.2	18.7	8.7	50.0	32.6	20.9	10.7
3RKON	473	14	1722	0.831	44.9	18.9	13.4	7.2	49.0	22.2	15.9	9.4
1OCCA	514	12	754	0.979	70.8	42.7	31.1	14.7	**77.1**	**52.7**	34.4	17.6
1OCCC	261	6	521	0.958	65.4	29.0	19.2	8.7	61.5	29.8	18.8	7.9
1PP9C	379	8	581	0.921	73.7	29.0	17.4	7.9	**79.0**	32.1	21.1	9.5
3H90A	283	6	1039	0.968	45.0	31.7	19.9	10.0	50.0	29.7	16.4	9.1
3B45A	180	6	1073	0.867	66.7	41.1	28.3	12.4	**77.8**	44.4	27.2	13.2
1PW4A	434	12	1604	0.878	8.9	11.1	7.8	4.5	13.3	10.2	9.5	5.3
3DHWA	203	5	1065	0.877	**54.6**	30.3	23.0	10.6	50.0	31.2	24.4	11.4
1YMGA	233	7	1010	0.901	15.4	12.1	9.9	5.3	15.4	12.9	9.1	5.6
3B60A	572	6	1568	0.881	**42.4**	20.9	16.0	8.3	**42.4**	**27.6**	**16.9**	**8.7**
2A65A	510	12	1043	0.825	59.6	31.9	22.2	11.2	55.8	35.0	25.8	13.4
1HZXA	340	7	1151	0.803	31.3	23.3	17.7	8.1	43.8	**29.6**	21.1	9.5
3PJZA	468	12	923	0.793	6.1	6.1	3.4	2.0	**10.2**	**7.3**	3.2	2.2
2XUTA	456	14	1040	0.706	32.7	14.1	8.6	5.0	38.5	17.6	11.6	6.4
3ZUXA	308	10	1005	0.899	66.7	34.9	23.2	12.2	**69.7**	38.0	26.2	14.1
2XQ2A	538	15	1380	0.82	0.0	1.7	1.0	0.9	1.7	2.7	1.2	1.4
3M71A	306	10	646	0.971	6.5	10.8	8.0	6.2	9.7	16.6	13.4	8.1
3QE7A	407	14	1355	0.818	60.5	32.1	22.8	11.9	**69.8**	31.2	23.5	12.4
AVERAGE	376	9.68	1055	0.875	41.1	24.3	16.5	8.2	44.4	26.5	18.0	9.2

Part II

	DI Filtered MSA Stats				Best Dtree Accuracy (%)				Best ANN Accuracy (%)			
PDBID	**L**	**TM_{helix}**	**M_{eff}**	**Cov**	**L/10 ms 12**	**L/2 ms 6**	**1L ms 12**	**3L ms 12**	**L/10 ms 12**	**L/2 ms 6**	**1L ms 12**	**3L ms 12**
3NCYA	422	12	1205	0.891	**42.2**	**32.7**	**20.9**	**13.8**	8.9	15.7	15.1	13.5
2RH1A	442	8	451	0.652	37.8	19.9	13.4	8.7	**56.8**	**28.0**	**23.7**	**13.4**
1OKCA	292	6	1043	0.89	30.0	32.9	19.5	11.0	33.3	25.5	18.5	**15.2**
1XQFA	362	11	1021	0.961	**57.1**	**31.1**	**21.5**	12.7	35.7	25.4	19.6	**14.0**
3GD8A	223	7	1062	0.964	63.6	49.1	36.3	20.9	**95.5**	58.9	**48.4**	**27.7**
1L7VA	324	10	1045	0.914	**15.2**	12.3	11.4	8.8	12.1	**15.3**	**13.5**	**11.2**
3MKTA	460	12	1068	0.917	39.1	35.7	**29.1**	15.7	39.1	**36.1**	**29.1**	**22.0**
3RKON	473	14	1722	0.831	**63.3**	**38.3**	24.5	13.1	51.0	32.1	**31.6**	**19.2**
1OCCA	514	12	754	0.979	**77.1**	**57.7**	**36.1**	18.1	72.9	36.9	30.5	**20.2**
1OCCC	261	6	521	0.958	61.5	35.9	22.2	12.3	**80.8**	**41.2**	**27.2**	**13.7**
1PP9C	379	8	581	0.921	76.3	**36.3**	**23.2**	11.3	34.2	30.0	21.9	12.3
3H90A	283	6	1039	0.968	45.0	**37.6**	24.4	**16.9**	**55.0**	35.6	**25.9**	16.6
3B45A	180	6	1073	0.867	**77.8**	**46.7**	28.3	16.5	50.0	43.3	**35.6**	**21.7**
1PW4A	434	12	1604	0.878	8.9	12.4	15.5	11.5	**33.3**	**20.8**	**17.1**	**12.8**
3DHWA	203	5	1065	0.877	40.9	**37.6**	28.1	14.4	45.5	32.1	**28.6**	**17.1**
1YMGA	233	7	1010	0.901	15.4	14.4	11.0	10.0	**38.5**	**31.1**	**26.2**	**19.0**
3B60A	572	6	1568	0.881	15.2	4.3	4.0	3.2	15.2	4.9	7.7	6.3
2A65A	510	12	1043	0.825	**67.3**	**37.3**	22.5	12.8	63.5	31.5	**26.8**	**15.1**
1HZXA	340	7	1151	0.803	**65.6**	28.3	24.0	14.3	56.3	**29.6**	**31.6**	**19.7**
3PJZA	468	12	923	0.793	**10.2**	6.9	4.5	2.6	8.2	6.1	**7.7**	**6.0**
2XUTA	456	14	1040	0.706	**51.9**	**32.4**	**22.0**	**12.2**	42.3	27.5	21.8	11.6
3ZUXA	308	10	1005	0.899	66.7	**45.8**	32.5	18.5	66.7	44.0	**36.1**	**19.8**
2XQ2A	538	15	1380	0.82	0.0	4.7	1.7	2.1	**6.8**	**7.1**	**6.6**	**4.8**
3M71A	306	10	646	0.971	29.0	12.7	12.7	10.9	**38.7**	**29.3**	**21.7**	**14.4**
3QE7A	407	14	1355	0.818	67.4	32.6	23.8	**16.5**	39.5	**33.0**	**27.0**	15.1
AVERAGE	376	9.68	1055	0.875	**45.0**	**29.4**	20.5	12.3	43.2	28.8	**24.0**	**15.3**

Above is a two part table listing the final set of accuracies for the best model from each method category (naïve DI, processed filtered DI, decision trees, ANNs) with all optimal L-fractions for the given methods as well as L/10 at a minimum separation of 12 to show some of the highest precision prediction sets. All are given with the length (*L*), number of transmembrane helices (TM_{helix}), the effective alignment depth (M_{eff}), and target sequence coverage (*Cov*) matched to each PDBID for comparison. Additionally, the best accuracy for each PDBID and L-fraction is bolded. Decision trees outperform all other methods at L/10 and a minimum separation of 12 (45.0%) and also at L/2 and a minimum separation of 6 (29.4%). ANNs outperform all other methods at 1L and a minimum separation 12 (24.0%) and also at 3L and a minimum separation of 12 (15.3%).

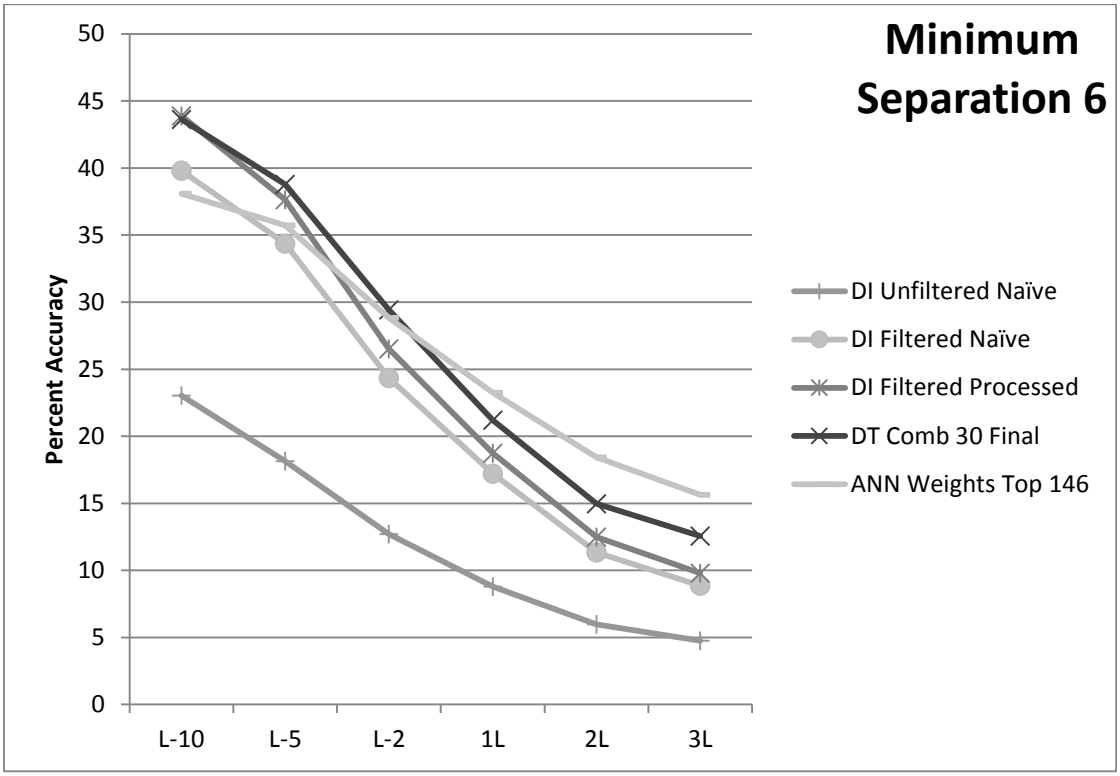


Figure 29: Accuracy Comparison across Best Decision Tree, ANN, Naïve Direct Information, and Processed Direct Information Contact Prediction for a Minimum Separation of 6

The graph above depicts the average accuracies of each method across the entire benchmark set for each of the top L fractions examined. Accuracy is significantly higher for DI contact predictions from filtered MSA in comparison to unfiltered and is further improved by processing (filtering based on predicted transmembrane topology). The processed method is best for the top L/10 predictions (43.89%) only slightly in comparison to the best decision trees (43.61%). For the top L/5 and L/2 decision trees are best (at 38.78% and 29.42%). For 1L, 2L and 3L ANNs optimized using an analysis of weights between nodes produces the best results (23.25%, 18.43%, and 15.64% respectively). Thus, for all L-fractions above L/10, one or more commonly both machine learning methods have a higher average accuracy. This is shown here for a minimum separation of 6 as that has been determined to be more useful for protein folding within BCL::Fold, however the effect is even more pronounced for a minimum separation of 12.

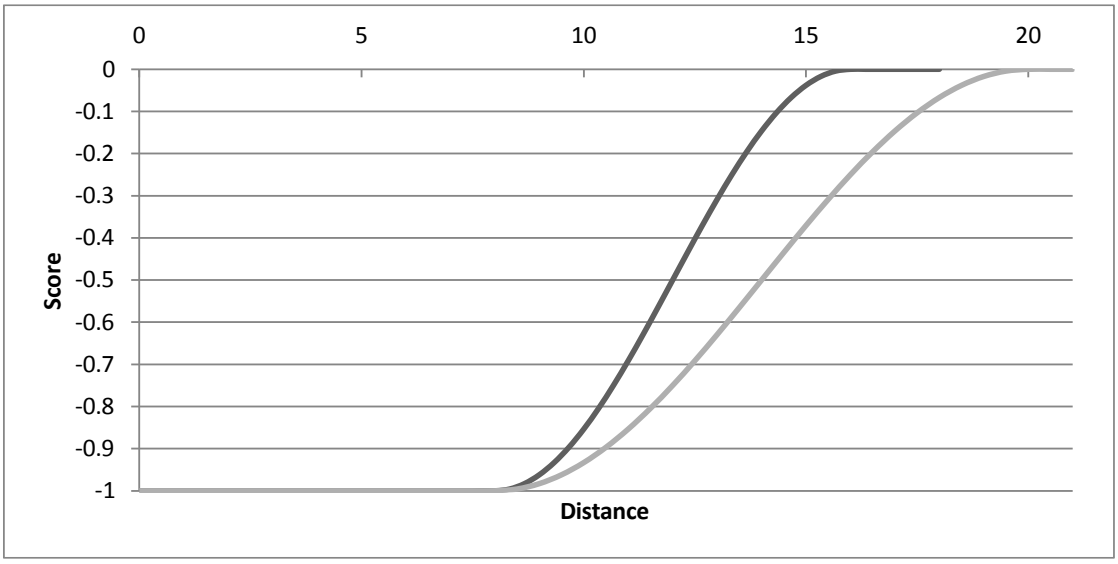


Figure 30: Original Scoring Function vs. Optimal Direct Information Scoring Function

The scoring function assigns an ideal score of -1.0 to all contacts within the contact distance threshold of 8 Å. The original parameters employed by BCL::Fold assign a score of 0.0, or no benefit nor penalty to anything outside of 16 Å, and assign anything between the two thresholds a value based on the sinusoidal transition function that increases from the ideal of -1.0 to 0.0. This transition region is designed to funnel near contacts towards the 8Å threshold during fold prediction. However, benchmarking determined that a wider transition of 12 Å (from 8 Å to 20 Å) both enriches for native–like model identification within a set of decoys as well as decreases RMSD100 if used during folding runs. As such, I used this parameter set for all predicted contact fold prediction.

Contact Score Function Optimization

The BCL::Fold method relies on a series of knowledge-based scores to evaluate models generated during Monte Carlo simulations and determine whether any given step has improved the protein fold prediction. Of note for this work is the contact scoring function, which evaluates how well a protein model satisfies a set of contact predictions or "contact restraints". This function, depicted in Figure 30, consists of a sinusoidal transition function and two threshold values. The first value represents the upper bound of the contact range. Anything below this value receives a -1.0, which is the best score possible for a contact restraint. The second parameter

represents the width of the transition region (indirectly setting the maximum of the transition function). Ideally, the contact threshold should represent the boundary for where two sites are near enough that the evolutionary forces from their interaction are equivalent. Similarly, the threshold for where the transition function ends should represent the distance at which interaction and thus direct evolutionary entanglement is negligible. Further, this threshold must also capture the maximum distance to extend to "pull" contact restraint pairs towards one another during the folding process. During the search across the energy landscape for low energy (and thus likely native-like) topologies, creating wider energy wells around minima simplifies discovery of minima by traveling down these wider gradually sloping gradients. Increasing the width of the transition function can widen these energy wells as long as it does not proceed to the point of obscuring the minima.

Initially, to evaluate these parameters for the BCL::Fold scoring function I used models previously created with the assistance of direct information based constraints as decoy models. I took ten random samplings for each protein and parameter set comprised of 10% "good" models – models below an 8Å RMSD100 threshold. The 8Å threshold for determining "good" models was used as this is approximately the RMSD100 of a native fold with a single transmembrane helix inverted (and thus a very close fold but an incorrect topology). In other words, this threshold roughly captures whether one has found the correct topology. I scored and ranked each set, then averaged the enrichment for "good" models in the top 10% by rank across all ten runs. The enrichment here is the fold difference between the percent of good models in this top 10% compared to that expected by random selection in the top 10%. The results are compared across parameter sets in Figure 31. The maximum enrichment of 3.76 occurred with a contact threshold

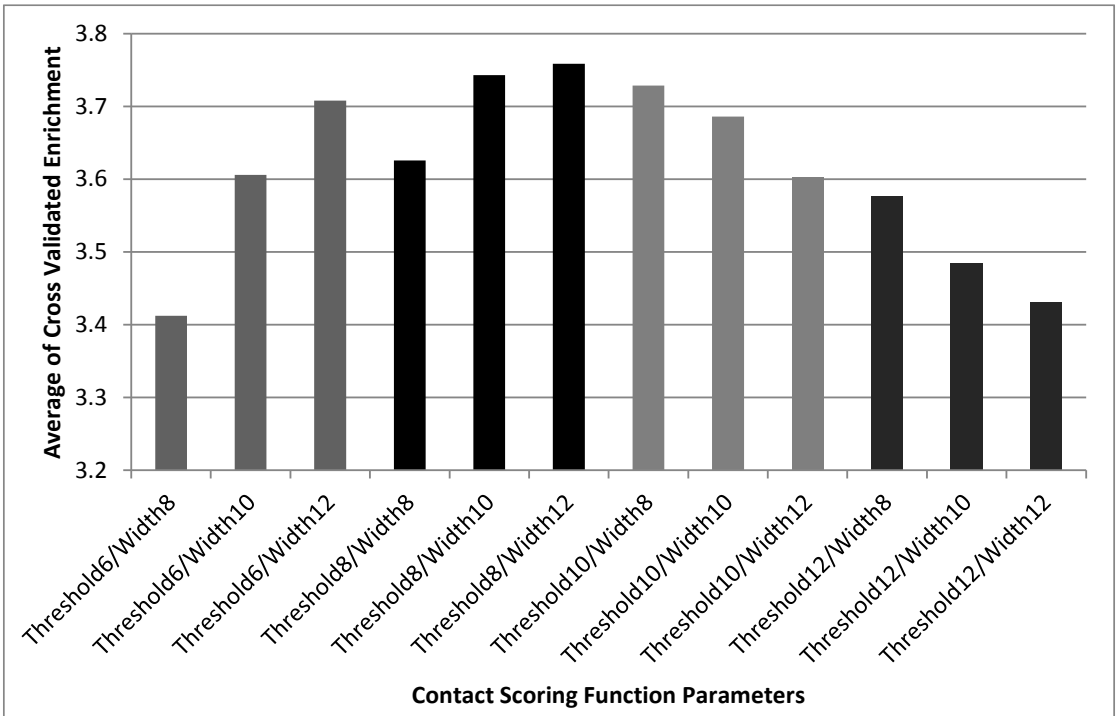


Figure 31: Comparison of Enrichment for Native-Like Models across Contact Scoring Function Parameter Sets Using the Entire Benchmark Set

The bar graph above depicts the enrichment for native-like models obtained by the BCL::Fold scoring function utilizing different contact threshold parameters (shading is used to group sets with equal initial thresholds). Enrichment was calculated by taking random subsets from models folded using direct information based contact predictions such that 10% were models within 8Å RMSD100 of the native fold (this threshold reflects whether a predicted fold likely captures the correct topology). Models within this threshold are labeled "good" and then all models in the subset are scored and ranked using the scoring function with the given parameters. The enrichment is then determined by calculating how much the top 10% ranked by the scoring function deviates from random or enriches for "good" folds. The maximum enrichment of is 3.76, which is seen for a contact threshold of 8Å with a transition function width of 12Å.

of 8Å and a transition function width of 12Å. There is also a trend for the lower thresholds of 6Å and 8Å that enrichment increases with wider transition functions. A lower initial threshold, up to a point, seems to enrich for models approaching the native topology. The trend is reversed for initial contact thresholds of 10Å and higher.

Detecting a native-like model by score may not accurately reflect that scoring function's ability to guide predicted structures towards a native-like fold. In other words, the scoring

function may have a very narrow well around the native structure – discriminating correct models very accurately but unable to discern between models further from the native fold. A wide sloping energy well drives the folding process. To verify the parameter set's capacity for model folding as well as enriching a decoy set of models with varying accuracy for native-like folds, a subset of the benchmark set was actively folded using a scoring function with the more promising parameter sets by enrichment. I then averaged the best ten models by RMSD100 and compared each set's results across parameter sets. The greatest average RMSD100 improvement is 1.991Å, using a contact threshold of 8Å and a transition function width of 12Å (Figure 32). However, a contact threshold of 6Å and a transition function width of 12Å also performs very similarly - 1.989Å. The trend for improved results as width increases when using a lower contact threshold is relatively preserved for folding a subset of proteins in comparison to the enrichment analysis (Figure 31).

For the remaining contact-assisted structure prediction results a contact threshold of 8Å and transition function of width of 12Å are used as this set of parameters performed best for both the enrichment as well as protein folding analyses. Interestingly, while similar to the original parameters (8Å cutoff threshold and 8Å transition width) there is an improvement from using a more generous width. This is likely related to the fact that direct information is both imperfect and the evolutionary influence captured by direct information does not necessarily cease at such a short distance as 8Å. As can be seen in the previous section on direct information based contact predictions, "incorrect" contacts often include near-contacts that are still informative for protein structure prediction. A more lenient set of scoring function parameters seems to coincide well with this point as the results are improved.

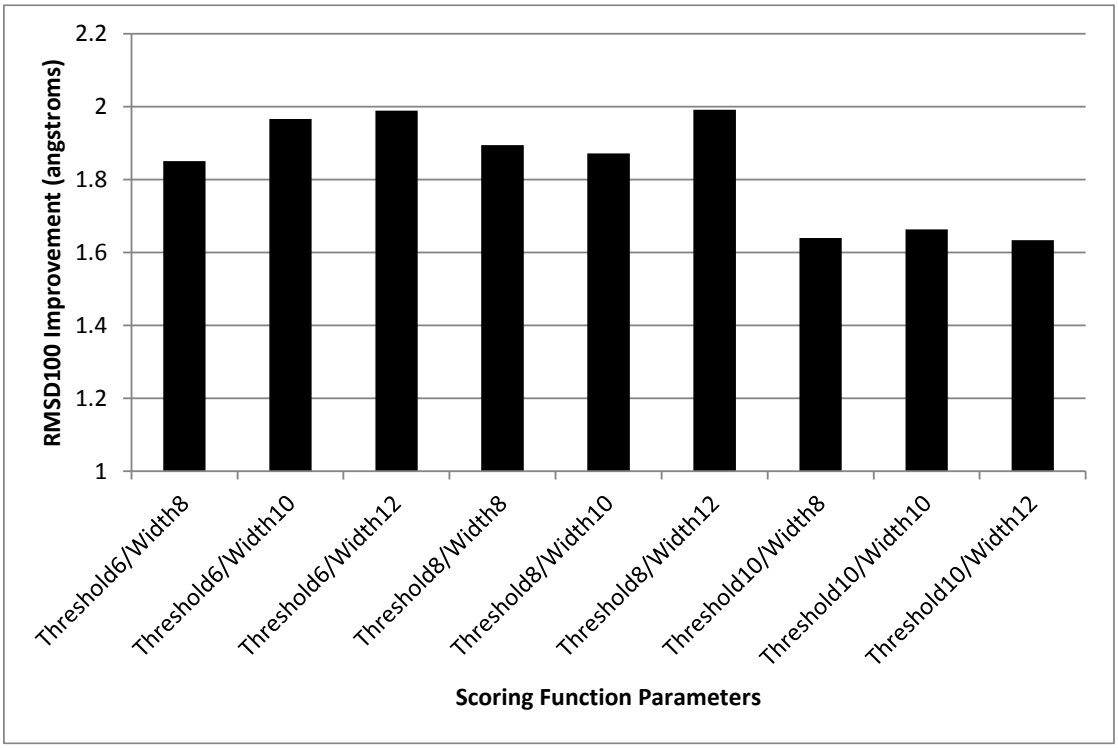


Figure 32: Average RMSD100 Improvement for Top 10 Models Across Various Contact Score Function Parameters

To verify that the best parameters for enriching for native-like folds also improved folding, I folded a representative subset of the benchmark set with contact scoring functions utilizing several parameter set combinations. I included all combinations of contact thresholds at 6Å, 8Å, and 10Å with transition function widths of 8Å, 10Å, and 12Å. I excluded thresholds above 10Å due to the exceptionally poor performance seen in Figure 31. The graph above shows the improvement in angstroms for the average RMSD100 of the top 10 models across a subset of the benchmark set. The maximum improvement for folding of 1.991Å is also achieved with a contact threshold of 8Å and a transition function width of 12Å. However, a cutoff threshold of 6Å and width of 12Å performs nearly as well with an average RMSD100 improvement of 1.989Å.

Protein Structure Prediction Using BCL::Fold and Contact Restraints

Accurate contact restraints limit the potential protein-fold search space thereby enabling more efficient conformational sampling. For methods such as BCL::Fold, the decrease in search complexity increases the likelihood of sampling native-like topologies. To first analyze the effect of including accurate contact restraints on BCL::Fold predictions I created sets of known contact restraints – having accuracies of 100% – for varying minimum separations and fractions of L. To determine how sensitive BCL::Fold is to variation in the distribution of contacts provided I used ten different randomly created sets of known contacts for the proteins shown in Figure 33. The

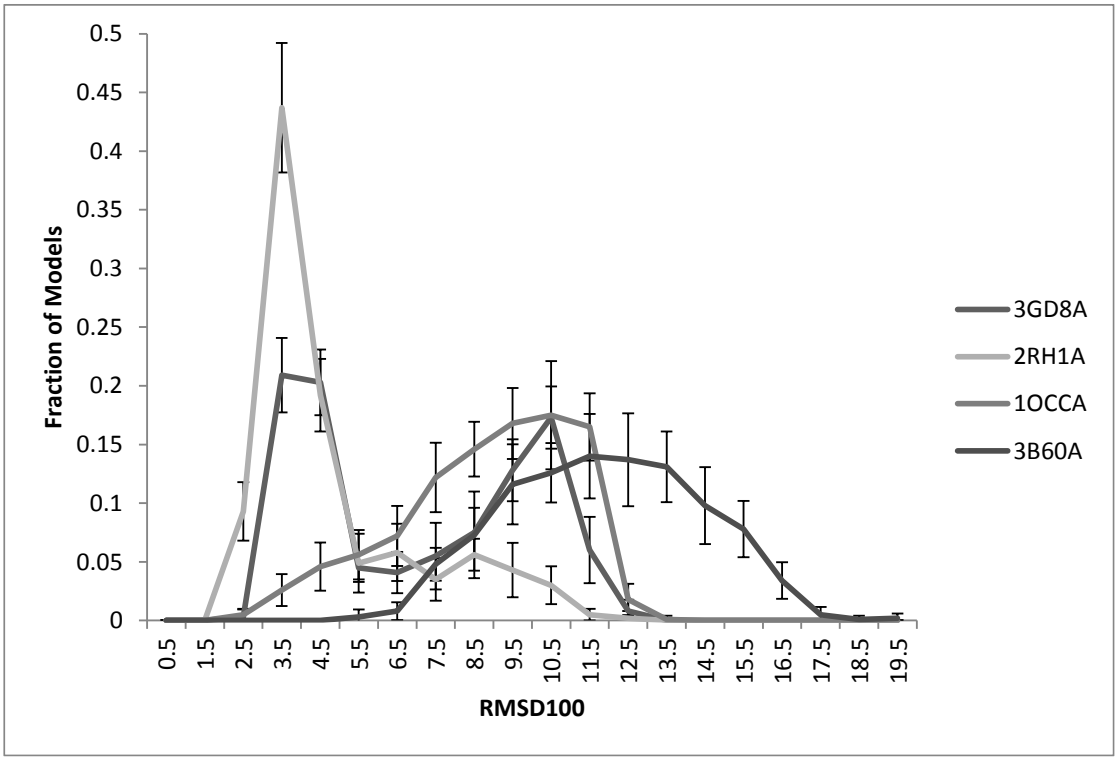


Figure 33: Comparison of RMSD100 Predicted Model Distributions across Runs with Different Sets of Known Constraints with Standard Deviation

Above are the resulting RMSD100 distributions for four proteins (3GD8A, 2RH1A, 1OCCA, and 3B6OA) aggregating the results from ten different simulations each generating 100 models (1,000 total with 2L contacts and a minimum separation of 12). The proteins above represent the diverse distribution morphologies present and how each morphology is relatively stable across different sets of known restraints.

distributions displayed represent the best performing parameter set for known contacts – an L-fraction of 2L and a minimum separation of 12 amino acids. For each restraint set, I generated 100 models with BCL::Fold and then took the average of the distributions for all 10 runs with error bars representing the standard deviation at each RMSD100. As one can see from the resulting merged distribution, the morphologies are relatively constant despite the variation between distributions for each run. Peaks range from near the 11.5Å down to the 3.5Å range. Cases of bimodal distributions are also included in Figure 33. The four proteins displayed also represent a diverse set of the morphologies seen across the entire benchmark set. This is true for both runs with and without contact restraints. Other distributions in this work are the result of 1,000 generated models using a single set of constraints determined using either DI or the machine learning models. For comparison, the positive control will be the aggregated result of all 10 sets of 100 models, or in other words the distribution resulting from all 1,000 models from the 10 randomly generated known constraint sets. This decreases the effect of the variation from a single randomly generated restraint set.

I also examined the effects of using minimum separations of 1, 6, and 12. Low minimum separations artificially decrease the number of informative contacts by including contacts with trivial sequence separations. In other words, positions separated by one or two amino acids are very frequently in contact due to their connection via peptide bonds. A minimum separation of approximately six ensures that contacts are on separate SSE and therefore contain useful information regarding the orientation of SSEs in three-dimensional space. As the minimum separation increases, the distribution of predicted models shifts towards lower RMSD100 models. However, once one reaches a minimum separation of six there appears to be an insignificant

benefit to increasing separation further. The shift from including contact restraints is significant. Thus, for known contacts, there is little difference between the results from sets using a minimum separation of 6 or 12 and the latter includes more information. However, as one can observe in Figure 36, predicted contact sets occasionally benefit from the use of a minimum separation of 6, although the results are still relatively similar. This is likely because contacts with a minimum separation between 6 and 12 may still be useful and more easily predicted than those above the more stringent cutoff of 12. Using a more conservative cutoff, occasionally decreases accuracy to a point that is more detrimental than the benefit of focusing on more informative long-range contacts. As such, I also optimized between the use of a minimum separation of 6 and 12 while examining the improvements from various L-fractions of contacts.

One can observe the potential improvement from using increasingly large L-fractions of known contacts in Figure 34. The resulting distributions for 2RH1A, 3GD8A, and 1OCCA are depicted in darker colors as the number of contacts increases. Concordantly, the distributions shift towards lower RMSD100. I have ordered them from most to least dramatic. The first, 2RH1A, results in a dramatic shift of the peak from 9.5Å to 3.5Å. However, it is more useful to examine the shift in the tail nearest the origin. This represents the best models sampled. This tail also experiences a significant shift (from 4.5Å to 2.5Å). The large increase in the fraction at these and nearby points also shows that there is a much higher sampling frequency for more native-like models. This enables faster structure determination and also simplifies the selection of models with correct topology. The increase in sampling frequency is less intense for 3GD8A and even less so for 1OCCA. However, a shift is still present in the tail of the distribution towards lower RMDS100 models. This trend of increasingly small RMSD100 across models exists across the

benchmark set and represents the upper limit on performance from using predicted contacts with BCL::Fold. In addition, these results also verify that including constraints of 100% accuracy significantly improve protein structure prediction accuracy.

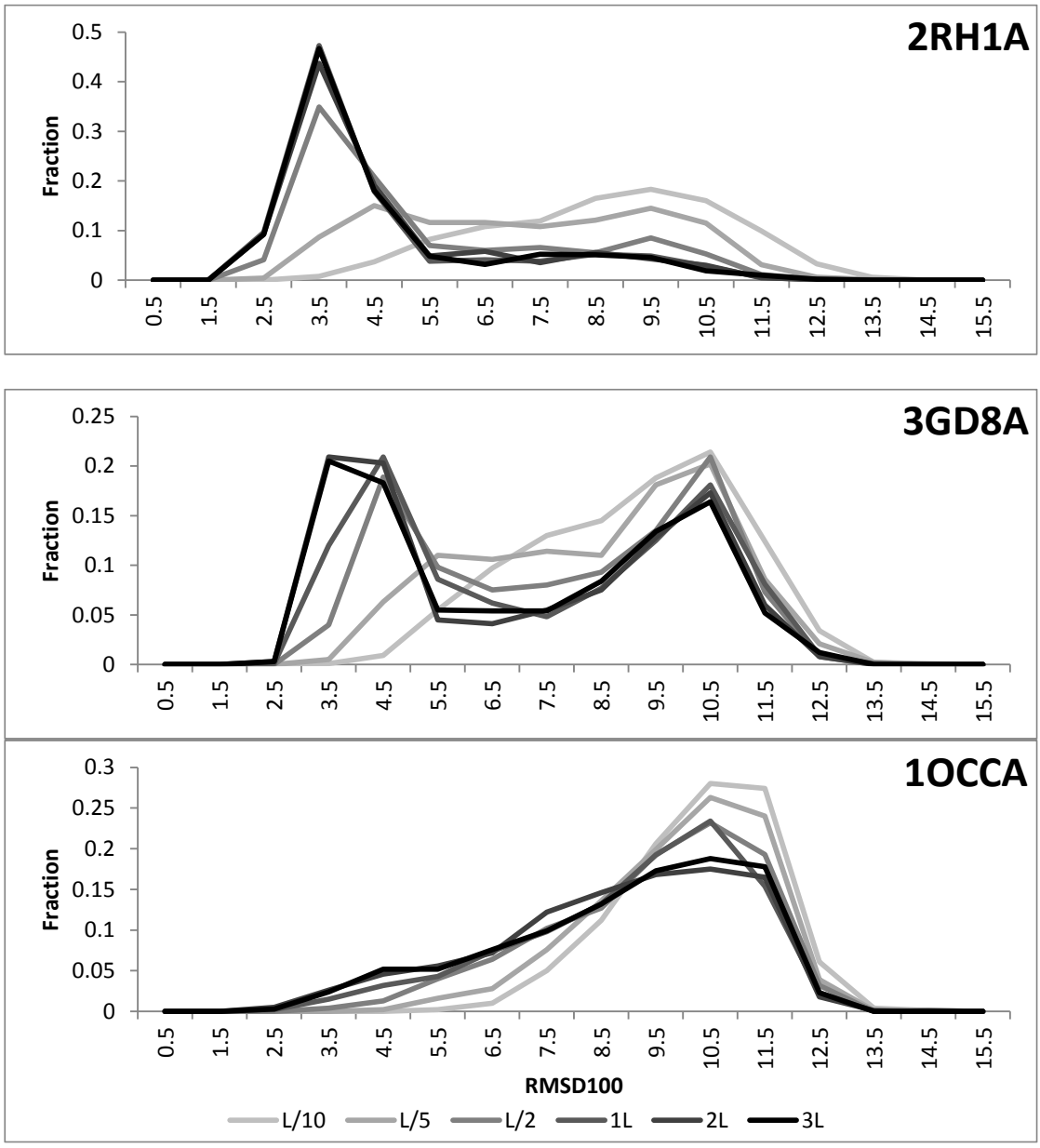


Figure 34: RMSD100 Distribution of Predicted Models as Increasing L-Fractions of Known Contacts are Used

The above set of proteins represents the range of improvement in model distributions from most to least drastic (2RH1A, 3GD8A, and 1OCCA). 2RH1A has a drastic shift from a peak at 9.5Å to 3.5Å. 3GD8A represents an intermediate improvement where the distribution becomes bimodal, with the original peak diminished at 10.5Å but not to the extent of 2RH1A. There is also a new peak at 3.5Å. Finally, 1OCCA has a peak which does not shift substantially but the tail is shifted towards lower RMSD100 scores, indicating some improved sampling with increasing numbers of contact restraints.

Having determined the upper limit of improvement, I compared the results using the above positive control restraint sets to the imperfect predicted contacts. I have included the comparison across negative control, naïve DI, processed DI, the best decision trees, the best ANNs, and the positive control in Figure 35. The negative control (black) shows the performance of BCL::Fold without any contact information. The predicted contacts all shift this distribution towards lower RMSD models although none of which are on par with the improvement achieved using known contacts (red). However, the shift seen with 1OCCA is very close to the performance of known contacts and is likely due to 1OCCA's high contact prediction accuracy. Both machine learning methods perform noticeably better than the DI based methods for 2RH1A. This is very encouraging and likely due to the many false positives that exist between opposite sides of the membrane which are removed by the use of machine learning and additional descriptors. Prediction accuracy is much better for 2RH1A when one the machine learning methods, even in comparison to the processed DI 12.37% the best ANNs achieve an accuracy of 30.65% for the top L/2 contacts at a minimum separation of 12. This is a nearly 2.5 fold increase in accuracy. Thus, contacts derived from machine learning methods result in similar or improved distributions as compared to DI based methods with and without further processing. Substantial improvement is still possible by further increasing the accuracy of contact prediction methods as evidenced by the significant discrepancy between distributions from known contacts to imperfectly predicted contacts. One may attain further improvement by leveraging the confidence associated with each contact prediction or dynamically adjusting the contact restraints used based on scoring and confidence values. Nevertheless, it is still encouraging to see a significant improvement using these imperfect contact sets over the performance seen from using no contact information.

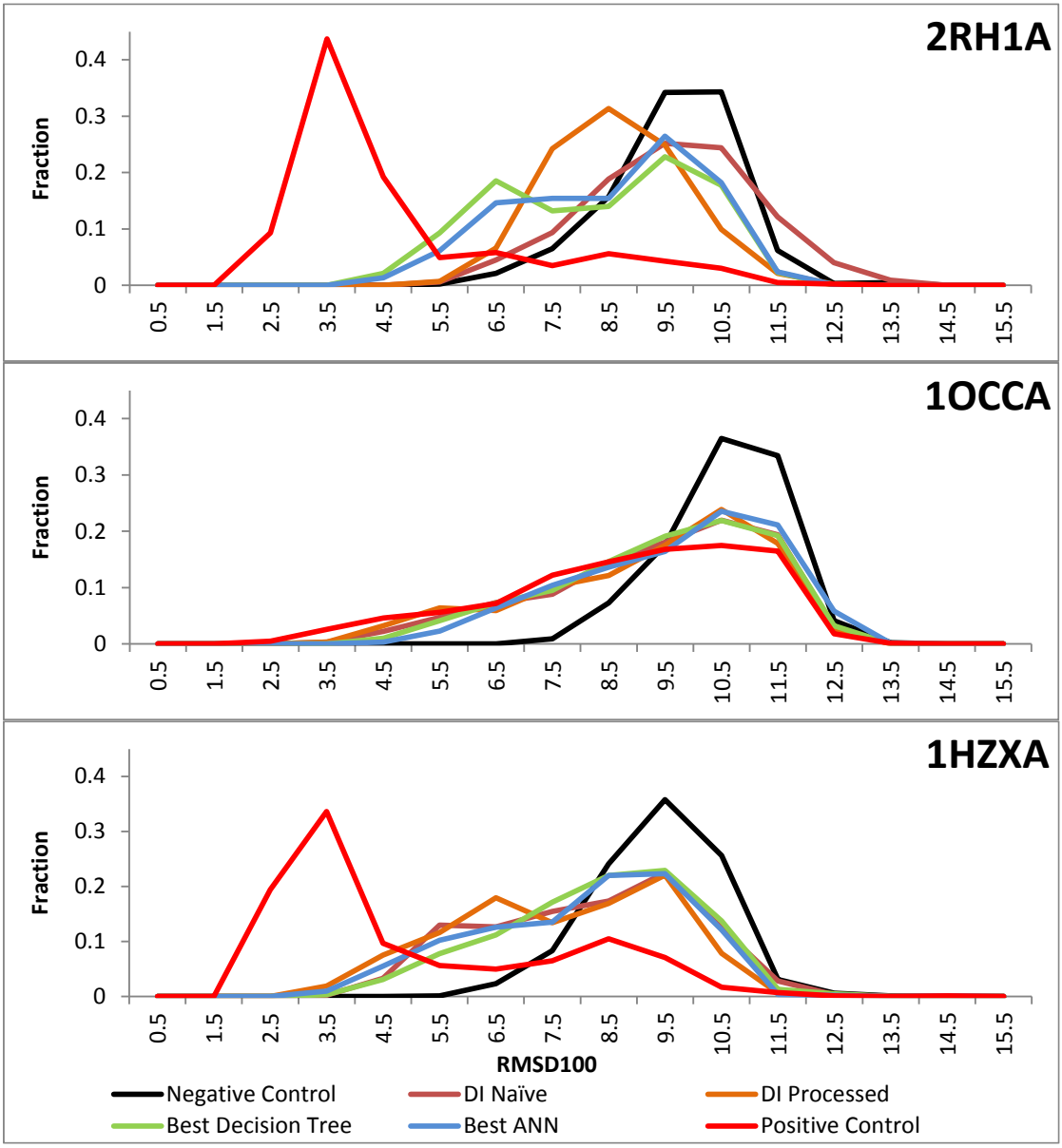


Figure 35: Comparison of Protein Model Distribution across Methods for 2RH1A, 1OCCA, and 1HZXA

The distributions across naïve DI (brown), processed DI (orange), the best decision trees (green), and the best ANNs (blue) above are book-ended by the distributions of the positive and negative controls (red and black respectively). The addition of contact restraints consistently shifts the distributions towards lower RMSD100 models. There is little difference between methods for 1OCCA and 1HZXA. However, both machine learning methods shift more substantially towards lower RMSD100 values in the case of 2RH1A. One should also note that the distributions for all experimental methods approach that of the positive control for 1OCCA.

Before progressing to a final comparison across methods, I first determined the best minimum separation and L-fraction (on average) for each method. I have included the results of this separation and L-optimization in Figure 36. For each method, I generated 1000 models for the same subset of 9 of the 25 benchmark proteins across all L-fractions examined for a minimum separation of 6 and 12. I then determined the 10 best models by RMSD100 for each set of conditions and calculated the difference between the average for this top set compared to the 10 best models for the models generated without any contact information. I determined the average RMSD100 improvement across all nine proteins and linked these values across L-fractions in the line graph in Figure 36.

The positive control, with a minimum separation of 6 and 12 (gray and black), plateaus at approximately 2L contacts. The maximum occurs at 2L with a minimum separation of 12. There is very little difference between the different separation parameters for the control, but the predicted contacts have much more variation between the two separations – especially for contact fractions greater than L/2. Naïve DI is especially similar between the two for contact fractions less than or equal to L/2. The difference in performance also seems anti-correlated with the increasing accuracy from naïve DI, to the best decision trees, and finally to the best ANNs. In addition, there also appears to be a correlation between accuracy and the L-fraction that results in the best BCL::Fold performance. As accuracy increases across these three examples the best L-fraction also increases – L/2 to 1L and finally to 3L. Naïve DI is also the only method with a maximum resulting from a set on contacts with a minimum separation of six. This may indicate that the method's accuracy drops off more rapidly as one attempts to predict increasing numbers

of contacts. Machine learning based methods appear to reduce this trend, which may be because they do not rely solely on DI to rank potential contact pairs. I also examined processed DI and it performed nearly identically to naïve DI, as such, I did not include it. Finally, it is also interesting that the results from the best ANN models do not peak but rather continue to improve across the entire set of L-fractions examined. This is similar to the pattern seen with known contacts, as a larger fraction of false positive does not accompany the additional information that can confound predictions. Thus, the similarity of the ANN based method to the L-optimization trend seen with known constraints further suggests that ANNs are most accurate as well as beneficial for protein structure prediction.

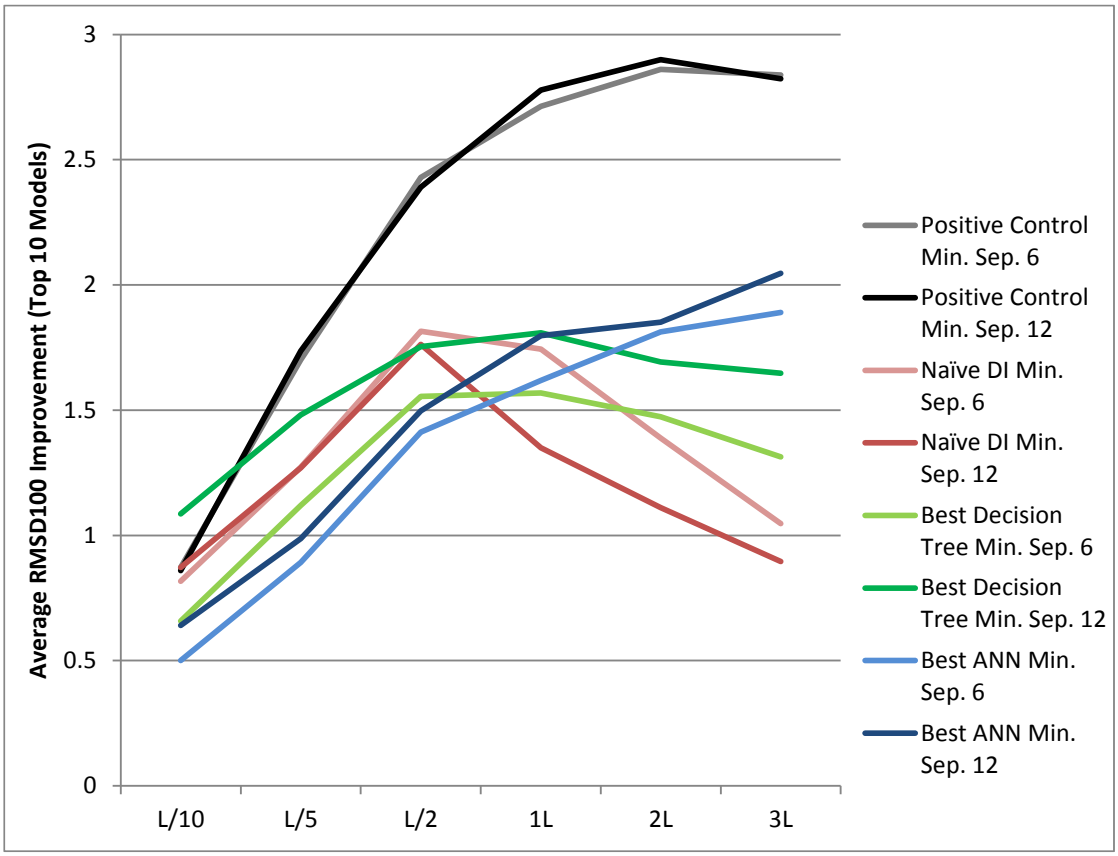


Figure 36: L-Fraction Optimization for Structure Prediction Using Contacts from the Positive Control, Naïve Direct Information, Best Decision Tree, and Best ANN

Above is the average RMSD100 improvement for the top 10 models across the L-fractions examined with a minimum separation of 6 and 12 (light and dark colors respectively) for the positive control (black), naïve DI (red), best decision tree (green), and the best ANN (blue). Using known contacts leads to greater improvement, which plateaus at 2L and a maximum average improvement with a minimum separation of 12 of 2.90Å. Naïve DI, and the best decision tree peak for L-fractions of L/2 and 1L at a minimum separation of 6 and 12 respectively (1.81Å for both). Finally, the best ANN peaks for the maximum L-fraction of 3L and a minimum separation of 12 with a maximum average improvement of 2.05Å.

I applied the optimal L-fractions and minimum separations determined using the nine protein subset of the benchmark set to predict structures for the entire benchmark set. I have depicted the distribution of the average angstrom improvement for the top 10 models from each protein across the top DI and machine learning methods in Figure 37. The improvement of each set is in comparison to the top 10 models from the negative control set, which was generated

without contact information. Known contacts result in the greatest average improvement - 4.14Å. The other experimental methods all significantly improve over the negative control but are not statistically significantly different from one another as determined by the Wilcoxon signed-rank test. However, the distributions as indicated by the box plots do suggest that the higher accuracy from ANNs does lead to more consistent improvement as compared to the DI only methods and the best decision trees. In addition, the average improvement is highest for results from structures predicted using the contacts from the best ANNs (1.70Å) compared to the second best, processed DI (1.65Å). Thus, predicted contacts are beneficial for protein structure prediction and results suggest that increased prediction accuracy in this range improve BCL::Fold performance. Other modifications to BCL::Fold that take better advantage of contact predictions may serve to highlight the ANN's higher accuracy within the realm of protein structure prediction.

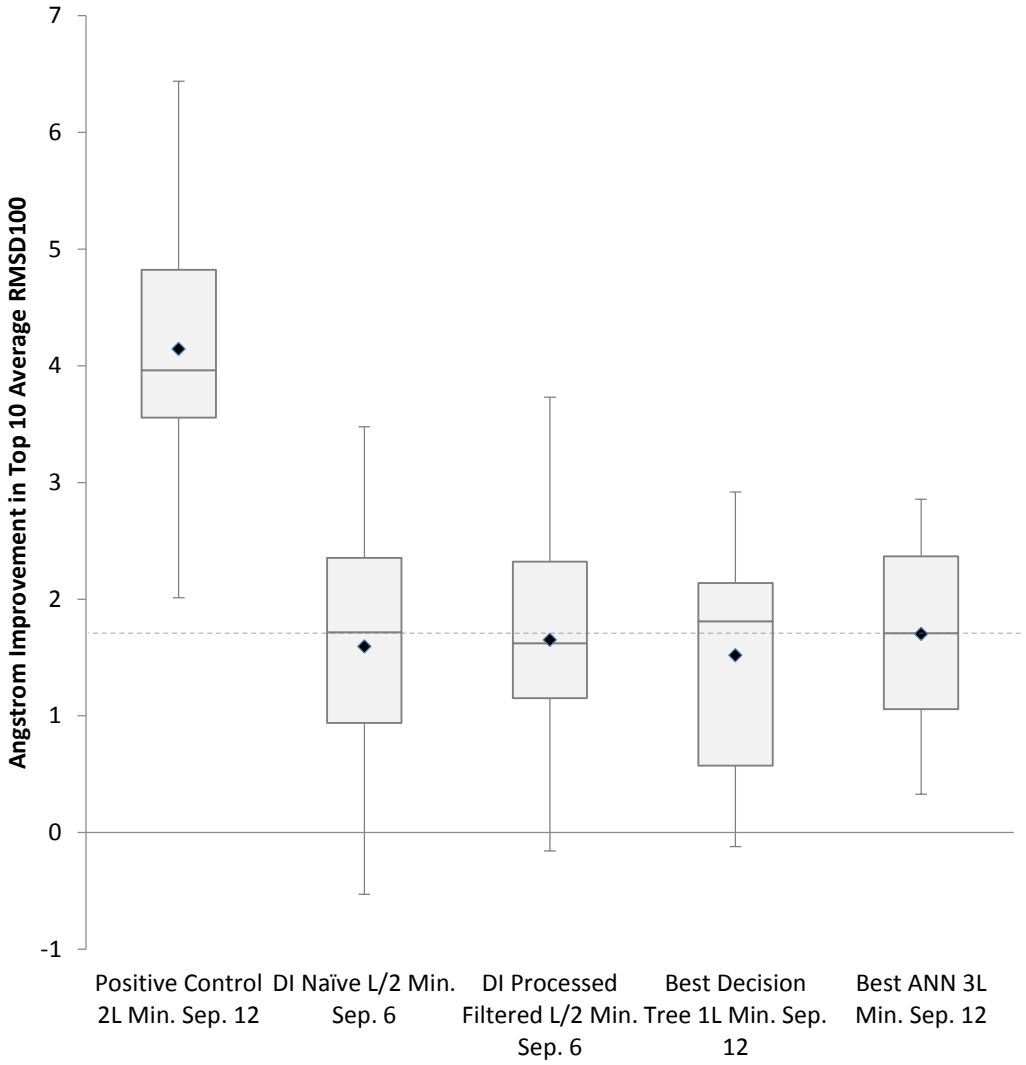


Figure 37: Box Plot Comparing Top 10 Models by Average Angstrom Improvement in RMSD100 across Benchmark Set for Best Direct Information, Decision Tree, and ANNs

The box plot depicted above shows the distribution of the average improvement in RMSD100 across the entire benchmark set for the best L-fraction and minimum separation across naïve DI, processed DI, the best decision trees, and the best ANNs in comparison to the positive control set (4.14Å). Among the experimental set the best ANN has the highest average improvement (1.70Å) as opposed to the average for naïve DI, processed DI, and the best decision trees (1.60Å, 1.65Å, and 1.52Å). Calculating the statistical significance of the minor average improvement using ANNs using the Wilcoxon signed-rank test found no statistically significant difference between that and the DI-based or decision tree sets. However, the distribution as gauged by the quartiles does appear to be consistently higher for ANNs. Lastly, there is still further room for improvement before these methods are able to match the performance achieved with known contacts.

In addition, Figure 40 shows how the performance is improved individually across the entire benchmark set of 25 proteins. All points are below the diagonal indicating that the average

RMSD100 of the top 10 predicted models is improved by the inclusion of contact restraints (in this case from the best model – the optimized ANN with the top 3L using a minimum amino acid separation of 12). Two proteins show relatively little improvement and have larger RMSD100 values regardless of method – 2XQ2A and 3B6OA at (9.28Å, 8.85Å) and (10.01Å, 9.60Å) respectively.

Table 5 and Table 6 display the aggregated results for the best runs across the entire benchmark set. The former contains the baselines provided by both the negative and positive

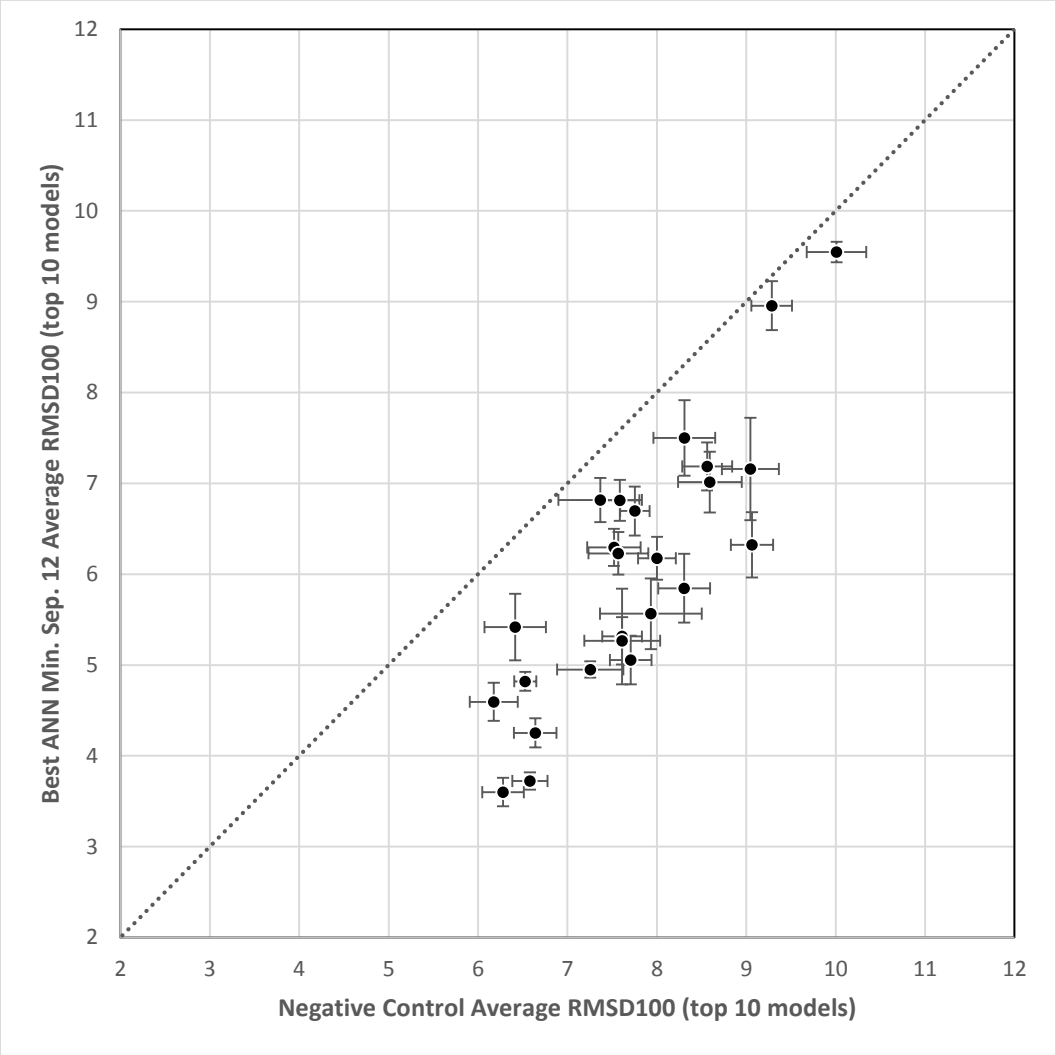


Figure 38: RMSD-RMSD Comparison of the Top 10 Models from Runs with Contact Restraints from the Best Model (ANN) and without Any Contact Restraints

The plot shows the relative benefit of using predictions from the best model, the optimized ANN at 3L minimum separation of 12, in comparison to folding without any contact restraints. Average RMSD100 for the top 10 models in each run are plotted such that the result from the negative control set is given on the x-axis and the set using the best predictions is on the y-axis. Equal performance is depicted via the dotted x=y line and any point below this diagonal is improved by inclusion of our predicted restraints. All 25 points are below the diagonal showing consistent improvement across the benchmark set.

control sets. The inclusion of known contacts significantly reduces the RMSD100 across the entire set. These results indicate the potential upper bound for BCL::Fold using contact predictions with perfect accuracy. Table 6 contains the folding results for runs that included predicted contact constraints from the best naïve DI, processed filtered DI, decision tree, and ANN methods. Of the predicted methods, our best ANN method's predicted contact restraints result in the lowest average best single model (5.50Å) and best average top 10 models (6.05Å) by RMSD100. The best result for each PDBID is bolded in the table. The ANN is best for the most benchmark models for both metrics, with 10 of the 25 best single models and 8 of the 25 best average top 10 models. The next best method for best model is processed and filtered DI, with 7 of the 25 best single models. The next best method for the top 10 is naïve DI, with 8 of the 25 best top 10 average.

Table 5: Table of Full Benchmark Positive and Negative Control Folding Results (Top 10 Average RMSD100)

	DI Filtered MSA Stats				Negative Best RMSD100			Pos. Control 2L ms 12		
PDBID	***L***	TM_{helix}	M_{eff}	***Cov***	**Best**	**Top 10 Avg**	**S.D.**	**Best**	**Top 10 Avg**	**S.D.**
3NCYA	**422**	12	1205	0.89	7.90	8.56	0.28	2.53	2.78	0.11
2RH1A	**442**	8	451	0.65	5.52	6.17	0.27	2.22	2.44	0.09
1OKCA	**292**	6	1043	0.89	6.45	7.61	0.42	3.56	3.89	0.17
1XQFA	**362**	11	1021	0.96	7.46	7.75	0.16	2.34	2.45	0.07
3GD8A	**223**	7	1062	0.96	6.25	6.58	0.20	2.88	3.08	0.13
1L7VA	**324**	10	1045	0.91	6.53	7.37	0.47	2.90	3.12	0.11
3MKTA	**460**	12	1068	0.92	6.33	7.93	0.57	3.86	4.17	0.20
3RKON	**473**	14	1722	0.83	7.80	8.30	0.29	2.86	3.36	0.21
1OCCA	**514**	12	754	0.98	7.33	7.71	0.23	2.32	2.85	0.32
1OCCC	**261**	6	521	0.96	7.44	8.00	0.21	5.36	5.59	0.08
1PP9C	**379**	8	581	0.92	7.15	7.59	0.22	3.56	4.04	0.27
3H90A	**283**	6	1039	0.97	6.62	7.26	0.37	3.17	3.84	0.24
3B45A	**180**	6	1073	0.87	6.30	6.53	0.12	3.46	3.66	0.10
1PW4A	**434**	12	1604	0.88	6.98	7.57	0.33	2.46	2.82	0.21
3DHWA	**203**	5	1065	0.88	6.94	7.52	0.30	5.33	5.51	0.11
1YMGA	**233**	7	1010	0.9	6.42	6.64	0.24	2.84	3.08	0.12
3B60A	**572**	6	1568	0.88	9.37	10.01	0.33	5.56	6.27	0.40

2A65A	**510**	12	1043	0.83	8.64	9.06	0.24	2.20	2.63	0.15
1HZXA	**340**	7	1151	0.8	5.77	6.28	0.23	2.08	2.16	0.06
3PJZA	**468**	12	923	0.79	7.69	8.59	0.36	4.14	4.82	0.35
2XUTA	**456**	14	1040	0.71	7.74	8.31	0.34	3.18	3.74	0.26
3ZUXA	**308**	10	1005	0.9	7.11	7.61	0.22	2.66	2.79	0.08
2XQ2A	**538**	15	1380	0.82	8.75	9.28	0.23	3.40	3.88	0.39
3M71A	**306**	10	646	0.97	5.88	6.41	0.34	2.34	2.45	0.08
3QE7A	**407**	14	1355	0.82	8.43	9.05	0.32	4.37	4.64	0.18
AVERAGE	**376**	9.68	1055	0.875	7.15	7.75	0.29	3.26	3.60	0.18

The table above displays the average, best, and standard deviation with respect to the RMSD100 across the entire benchmark set for both the positive and negative controls. The positive control shown was the best performing run analyzed and utilized 2L contacts at a minimum separation of 12. In addition, each result row has the length (*L*), number of transmembrane helices (TM_{helix}), the effective alignment depth (M_{eff}), and target sequence coverage (*Cov*) matched to each PDBID.

Table 6: Table of Full Benchmark Folding Results with Predicted Contact Restraints from the Best Methods Across Categories (Top 10 Average RMSD100 Å)

	DI Naïve L/2 ms 6			DI Processed Filt L/2 ms 6			Best Dtree 1L ms 12			Best ANN 3L ms 12		
PDBID	**Best**	**Top 10 Avg**	**S.D.**	**Best**	**Top 10 Avg**	**S.D.**	**Best**	**Top 10 Avg**	**S.D.**	**Best**	**Top 10 Avg**	**S.D.**
3NCYA	**6.15**	**6.84**	0.39	6.17	6.89	0.47	6.26	6.85	0.38	6.66	7.19	0.26
2RH1A	5.84	5.98	0.08	5.49	5.86	0.23	4.26	**4.47**	0.10	**4.24**	4.60	0.21
1OKCA	5.12	5.90	0.32	5.49	5.99	0.30	5.28	5.47	0.13	**4.56**	**5.27**	0.26
1XQFA	5.87	6.81	0.51	**5.76**	6.53	0.38	5.81	**6.46**	0.31	6.01	6.70	0.27
3GD8A	3.58	**3.68**	0.06	4.08	4.26	0.11	3.62	3.82	0.13	**3.54**	3.72	0.09
1L7VA	**5.19**	6.34	0.54	5.71	**6.33**	0.33	6.09	6.91	0.41	6.31	6.82	0.24
3MKTA	6.27	6.67	0.14	5.65	5.85	0.12	6.11	6.50	0.24	**4.73**	**5.57**	0.39
3RKON	6.11	6.98	0.48	5.69	6.34	0.31	7.17	7.86	0.30	**4.98**	**5.85**	0.38
1OCCA	3.94	4.23	0.22	**3.84**	**4.16**	0.20	4.45	4.79	0.17	4.46	5.06	0.27
1OCCC	5.63	**6.06**	0.22	6.20	6.39	0.16	6.02	6.19	0.13	**5.52**	6.18	0.24
1PP9C	5.26	5.63	0.21	5.30	5.62	0.20	**5.04**	**5.51**	0.23	6.41	6.81	0.22
3H90A	**4.78**	**4.90**	0.09	5.71	5.80	0.05	5.05	5.33	0.14	4.79	4.95	0.09
3B45A	**4.29**	**4.54**	0.12	4.64	4.93	0.17	4.43	4.63	0.15	4.70	4.82	0.10
1PW4A	5.20	5.47	0.17	**4.50**	**5.11**	0.34	5.35	5.67	0.23	5.78	6.23	0.23
3DHWA	6.41	6.63	0.11	**5.81**	6.30	0.21	6.88	7.19	0.13	5.91	**6.30**	0.21
1YMGA	**3.89**	**4.22**	0.18	4.06	4.23	0.12	4.29	4.43	0.06	3.99	4.25	0.16
3B60A	8.69	9.36	0.32	**8.57**	**8.86**	0.18	9.84	10.13	0.23	9.25	9.55	0.11
2A65A	4.78	5.59	0.37	**4.67**	**5.33**	0.32	6.02	6.42	0.29	5.72	6.32	0.36
1HZXA	3.95	4.14	0.16	3.31	3.61	0.17	3.64	4.12	0.20	**3.28**	**3.60**	0.16
3PJZA	8.01	8.44	0.18	7.95	8.53	0.25	7.41	8.02	0.32	**6.42**	**7.02**	0.33

2XUTA	8.34	8.84	0.28	8.19	8.47	0.16	8.00	8.27	0.12	**6.73**	**7.50**	0.42
3ZUXA	4.88	5.14	0.14	4.50	**4.84**	0.21	4.87	5.17	0.21	**4.33**	5.31	0.53
2XQ2A	**8.27**	9.51	0.48	9.12	9.32	0.10	8.44	9.08	0.35	8.38	**8.96**	0.27
3M71A	4.69	**5.19**	0.28	**4.56**	5.53	0.35	4.88	5.31	0.18	4.57	5.42	0.37
3QE7A	**5.98**	**6.69**	0.40	6.23	7.30	0.43	6.81	7.12	0.21	6.13	7.16	0.56
AVERAGE	5.64	6.15	0.26	5.65	6.09	0.24	5.84	6.23	0.21	**5.50**	**6.05**	0.27

The table above displays the average, best, and standard deviation with respect to the RMSD100 Å across the entire benchmark set with contacts predicted from one of the following methods: naïve DI at L/2 and minimum separation of 6, processed and filtered DI at L/2 and minimum separation of 6, best decision tree at 1L and minimum separation of 12, or best ANN at 3L and a minimum separation of 12. The L-fraction and minimum separation for each method was independently optimized by sampling across L-fractions of L/10, L/5, L/2, 1L, 2L, and 3L as well as minimum separations of 6 or 12. Results are all matched to each PDBID and further alignment details can be found in Table 5. Additionally, the best single model and top 10 average results from all shown contact prediction methods for each PDBID is bolded. Our method using ANNs has both the lowest average RMSD100 for the best model (5.50Å) as well as the lowest top 10 average across the benchmark (6.05Å).

Finally, Figure 39 shows how close the best model by RMSD100 replicates the native fold for 1HZXA. Topology is correct and there is substantial superimposition of model helices with those of the native structure. One expects some deviation as BCL::Fold uses idealized helices without bends or kinks. These models still require the addition of side chains as well as other refinement but the similarity between the predicted model and the native greatly simplifies refinement and final all-atom predictions.

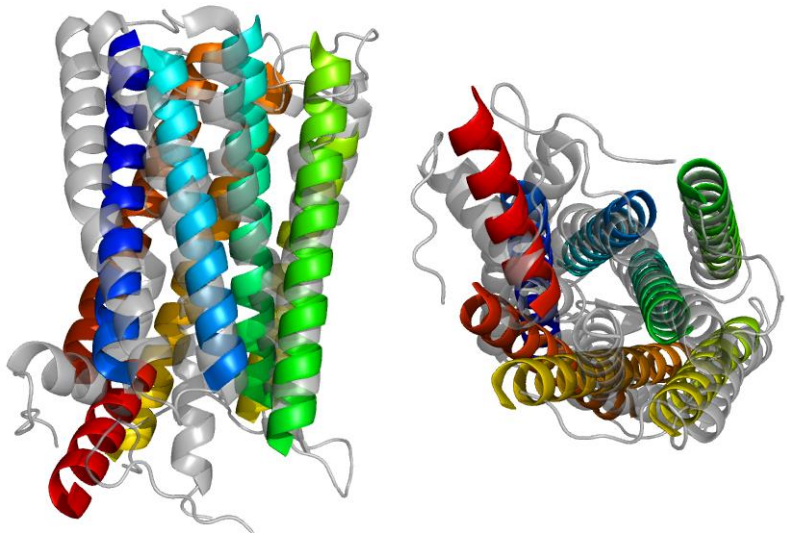


Figure 39: Visualization of Best Protein Model by RMSD100 Aligned to Native from Contact Predictions Made by the Best ANN (3L and Minimum Separation 12 for 1HZXA)

The best model by RMSD100 (3.3Å) is aligned above to the native structure. The model was produced as part of a folding run of 1000 proteins using the top 3L contacts as predicted by the best ANN with a minimum sequence separation of 12. Helices line up well, as can be seen from both an in-membrane and above-membrane view. Deviation from the native structure is due in part to the use of idealized SSEs that do not bend.

BCL::Fold Structure Prediction with Confidence-Based Scoring

The decrease in accuracy as one uses larger numbers of contact restraints highlights an opportunity for improving protein structure prediction within BCL::Fold. I have explored one potential adjustment to the algorithm, which differentiates between restraints at different positions within the ranking by weighting their respective scores differently. Equation 3 describes this modified weighting; each model's contact score is the sum of the scores as determined by the optimized scoring function multiplied by the normalized confidence of each prediction. The normalized weight is the contact prediction's confidence divided by the average confidence for the given restraint set. As such, this method weights scores of contact predictions with confidence values above the mean more heavily and vice versa. Thus, the higher accuracy contacts at the top of the ranking have more influence over the contact score while the false positives, which are more numerous among the lower ranked predictions, have less influence. In addition, true positives ranked lower within the given contact restraint set also have less influence on the model's contact score.

$$CS(p) = \sum_{i=1}^{f} S(c_i) \frac{D(c_i)}{\bar{c}} \qquad (3.)$$

Equation 3: Simple Confidence-Based Scoring Algorithm

The algorithm for confidence-based scoring (CS) for a given protein *p* is the sum of the fraction of *L (f)* restraint scores. Each restraint score (*S*) is multiplied by the confidence (direct information value) for the restraint divided by the mean confidence across all *f* restraints.

I evaluated the promise of such confidence based scoring by folding 1,000 models using contact predictions from direct information rankings both with and without confidence weighting.

I have presented the average RMSD100 for the top 10 best models for each protein from both conditions in Figure 40 (error bars are given displaying the standard deviation for each ten protein set). Average RMSD100 for models produced using confidence weighting is displayed along the y-axis and results from predictions without confidence weighting are on the x-axis. Points that lie along or very near the black dashed line have roughly equivalent average RMSD100 values regardless of confidence weighting. The points in Figure 40 suggest improvement from the addition of confidence weighting and vice versa. Of the eight proteins examined, five lie very near the diagonal – showing essentially no difference, two (2RH1A and 3MKTA) lie slightly below it – suggesting some possible improvement, and one (1OCCA) is far above the diagonal and is

significantly and negatively impacted by confidence weighting. The large variation in performance is likely related to the wide range of accuracies seen in this benchmark set. The distribution of direct information values, and consequently the distribution of confidence based weights, are much more similar across proteins than their accuracy. Thus, in cases where the confidence weights appropriately match the accuracy of the predicted contacts, the final models are closer to the native-like fold.

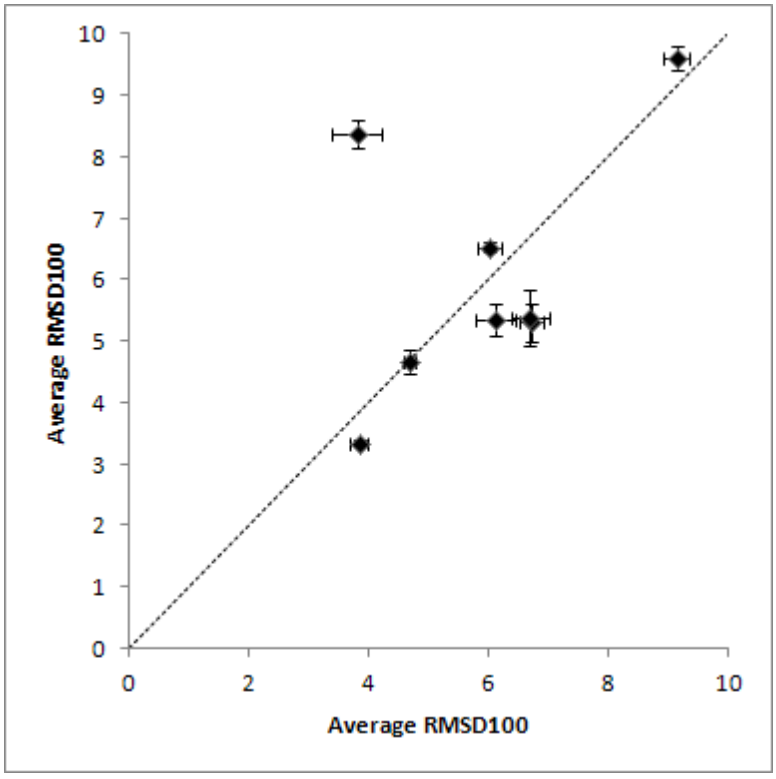


Figure 40: RMSD-RMSD Comparison of DI with and without Confidence-Based Scoring at 1L and Minimum Separation 6 for a Diverse Subset of 8 Benchmark Proteins

I calculated the mean of the top 10 models by RMSD100 for each protein for both folding runs performed with and without confidence-based scoring. There is little difference from adding confidence-based scoring across seven of eight proteins (points are very close to the diagonal dashed line, which indicates no change). One protein is significantly worse after the addition of confidence based scoring – 10CCA. This is likely due to its relatively poor accuracy using an early direct information based contact prediction set. The negative effects of the false positives included appears to be amplified by confidence scoring.

In most cases, the decrease in accuracy for DI predictions, as one uses contact restraints with lower DI values, does not properly match the simple formula of Equation 3. Cases with especially high accuracy lose information, as the formula will reduces the influence of still useful contact restraints below the mean. This results in an effectively smaller number of usable contact restraints. While confidence based scoring can reduce the effect of false positives, it is more appropriate for larger numbers of predicted contacts and must match the accuracy across the contact restraint set.

In Figure 41, I have also depicted the accuracy, direct information, and confidence functions for 3MKTA and 1OCCA, which improve and worsen respectively. The increased weighting for the highest ranked contact predictions for 3MKTA does increase the effect of restraints that are slightly more accurate, although accuracy remains relatively high across the entire set. The point at which the confidence weights cross the 1.0 threshold also coincides roughly with the beginning of a steady decline in accuracy for 3MKTA. On the other hand, the confidence weighting formula in Equation 3 does not capture the sharper decrease in accuracy in 1OCCA, which has a much smaller range of accurate contacts. 3MKTA's DI values do decrease more gradually than 1OCCA despite its smaller number of contacts. Further analysis of the distribution and magnitude of direct information values may elucidate methods to better predict the contact prediction accuracy for a given protein.

Confidence based scoring prediction is a promising potential avenue of research for further improvement of protein structure prediction accuracy, but results from this simple implementation suggest that significant work is necessary to achieve reliable improvement. Results would benefit from a better understanding of prediction accuracy for a given protein

based on the distribution of DI values. In addition, improved methods will likely be less reliant on information regarding the number of top contacts given. It would be ideal to determine an expected accuracy across a set of predicted contacts based on their direct information values alone – adjust weights to maximize the effects of true positives while minimizing the effects of false positives within the restraint set. In this way, one would enable the prediction of more native-like models despite the imperfect nature of DI-based restraints.

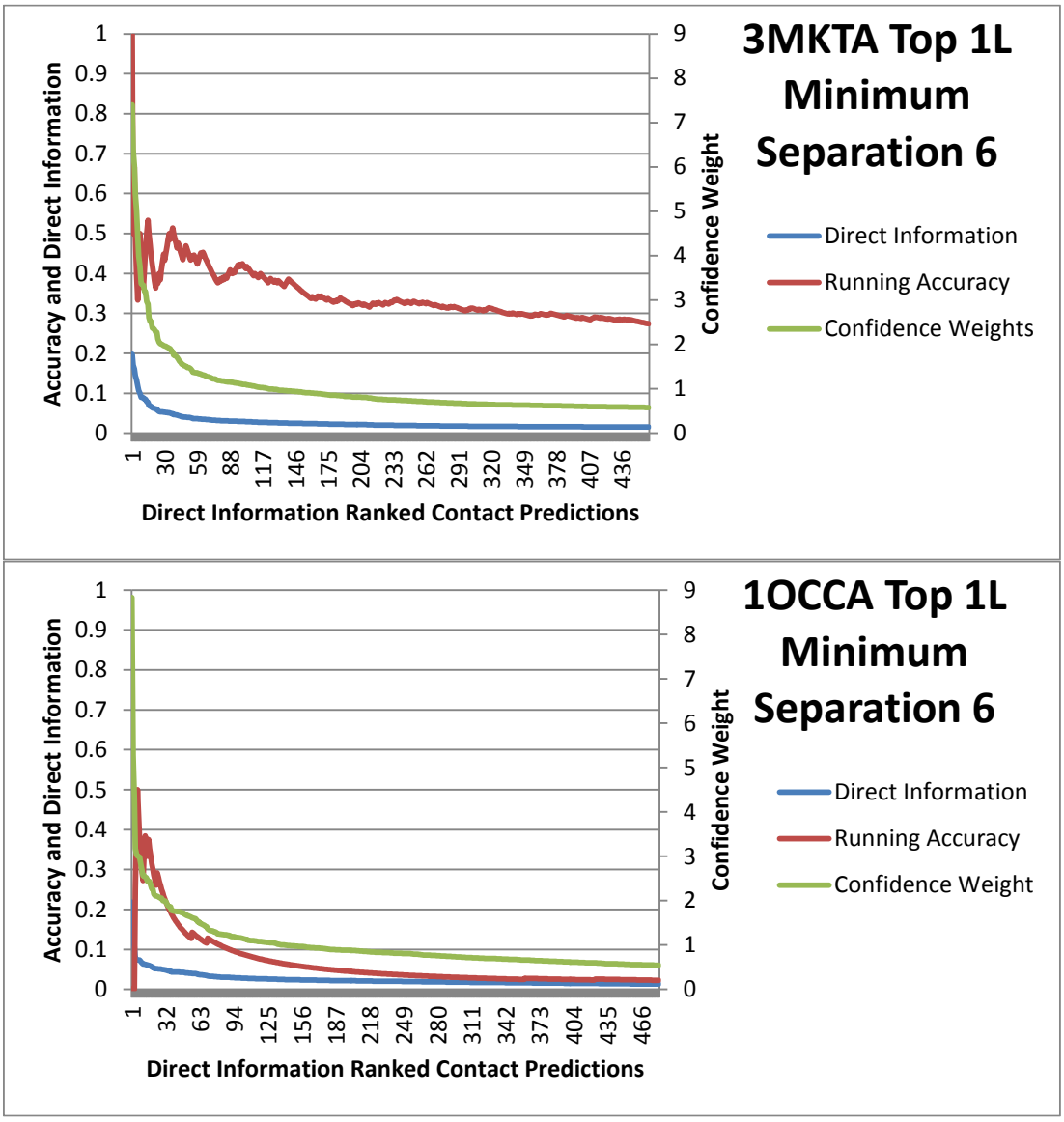


Figure 41: Comparison of Direct Information, Running Accuracy, and Confidence Weights across the Top 1L Contacts for 3MKTA and 1OCCA

Above I have included a comparison of the direct information values, running accuracy, and confidence weighting for 3MKTA, an average example of good contact prediction using direct information, and 1OCCA, a poorly performing example from an early direct information based method. The distribution and magnitude of the confidence weighting is relatively similar, while the direct information values are much larger and decrease more slowly for 3MKTA. The point where confidence weighting crosses the 1.0 threshold – distinguishes between the set of contacts weighted more and less heavily by confidence scoring. The accuracy for 3MKTA is much higher initially and stays around 30% across the entire set of the top L contact predictions. 1OCCA's accuracy drops off much more precipitously and quickly approaches 0%. Thus, the confidence based score increases the weighting for many contacts that are in a range at or below 10% accuracy.

BCL::Fold Structure Prediction Using Fractions of the Given Contact Restraints

The top L-fractions of contact predictions using solely direct information or combinations of such correlation descriptors with other sequence data via machine learning include some number of false positives. Such incorrect contact restraints negatively affect protein structure prediction. One potential approach to reduce their impact includes only attempting to satisfy fractions of the given contact restraints. I evaluate each contact restraint across a given model based on the scoring function discussed in the section "Contact Score Function Optimization". I then sort all restraints by their individual values and only the top indicated fraction is included in the model's score. If the fraction used is close to the actual positive predictive value and sampling is not biased against satisfying correct contact restraints, then some structure prediction runs may eventually satisfy most if not all correct restraints while excluding incorrect restraints. Such cases may more closely replicate the performance of the positive control sets, which consistently outperform current imperfect prediction methods.

Figure 42 displays the average improvement in RMSD100 for the top ten models for across a nine protein subset of the benchmark. I employed fraction thresholds of 0.25, 0.5, 0.75, 0.95, and 1.0, which are also compared to results from the positive and negative controls. Contact restraints are derived from the naïve filtered DI-based set. Including a fraction of contact restraints always improves accuracy for eight of the nine proteins. For 2RH1A, which does not consistently benefit from contact restraints, there is some improvement when one uses the top 2L restraints with a 0.25 fraction threshold. It is clear that inclusion of contact restraints improves

results and that the optimum fraction threshold varies with little difference between most thresholds. This is especially true when one uses the top 1L constraints. There is a slight shift towards lower contact fractions when one uses the top 2L constraints, which are usually far less accurate. Thus, enabling the scoring function to ignore results from a subset of the provided restraints seems to provide a slight but unreliable benefit for protein structure prediction when used with lower accuracy contact restraint sets. One of the most significant issues with this approach is the inability to determine the ideal fraction before knowing the accuracy of one's contact predictions. The fraction thresholds used in Figure 42 cover a very large range and as a result, most are very distant from the ideal. Using too low a fraction may underutilize correct contact predictions, while too large a fraction forces one to include false positives. The slight shift between the two different sized sets of contact predictions suggests that performance may benefit from a more nuanced application of contact fractions. It is also possible that the benefits of fractional scoring require one to more rigorously sample the search space to detect a significant benefit. Furthermore, while sampling all fractions may be too computationally demanding, one may be able to dynamically explore fraction thresholds during folding. Given a contact prediction method with reasonable accuracy, the smaller set of false positives should be included in the high scoring set less often; this is as the overall BCL::Fold scoring function guides each model towards a native-like structure. As such, one may explicitly eliminate contacts that score poorly more often. One may also gradually decrease the fraction threshold by monitoring the score across restraints over time. For example, beginning with all given constraints and finding that models score 90% of the contacts very highly, one could adjust the fraction threshold to 0.90 to see if overall scoring improved further.

Regardless, scoring based on fractions was not further evaluated, as the benefit appears to be relatively minor and unreliable given the large variation in contact prediction accuracy across proteins. In addition, these results also confirm that simply including relatively accurate contact prediction already significantly improves BCL::Fold performance.

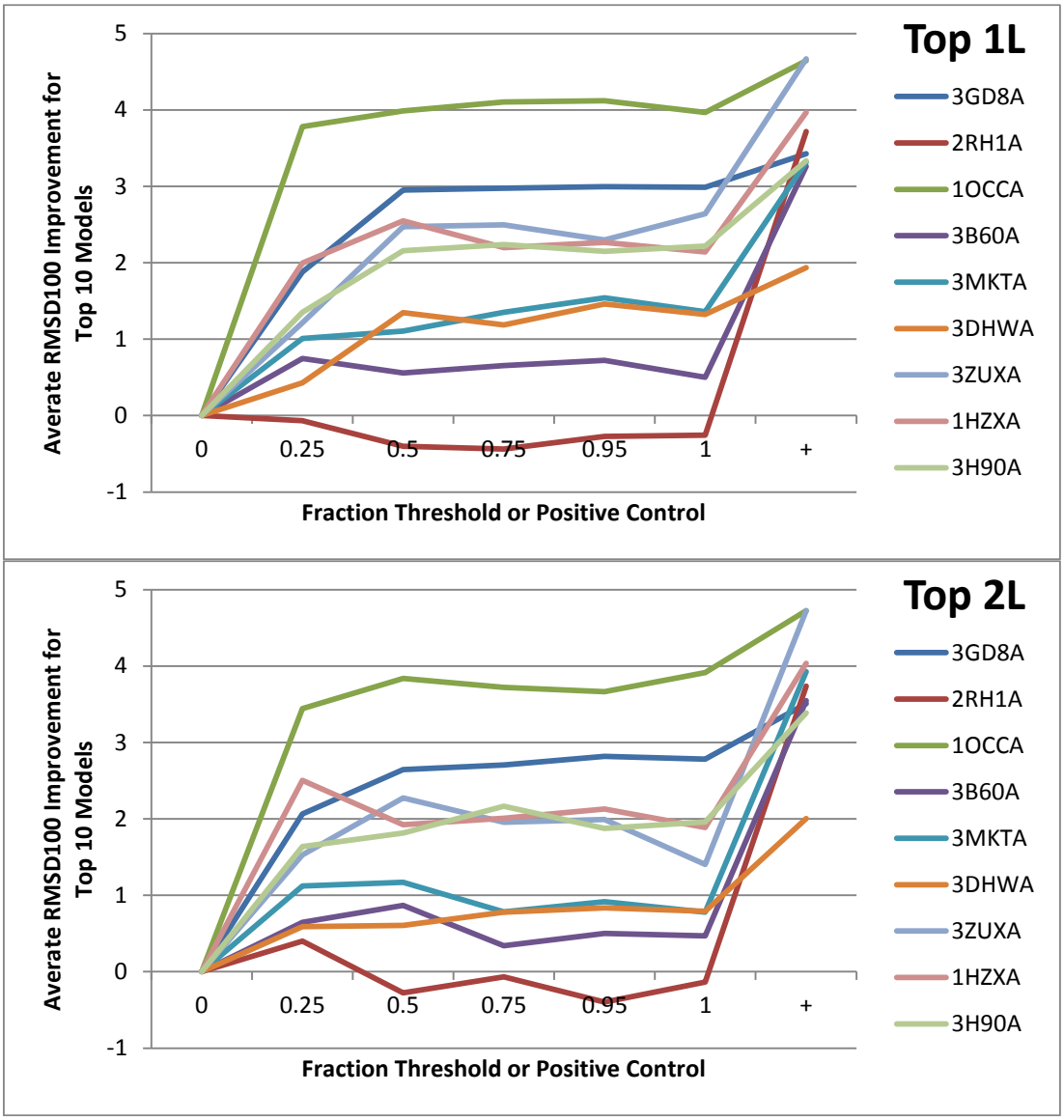


Figure 42: Comparison of RMSD Improvement with Different Sample Fraction Sizes

I modified BCL::Fold to use a threshold value to determine a fraction of contacts to satisfy - including them in scoring. Above are the average RMSD100 improvements for a subset of nine benchmark proteins. Results range from the negative (using zero of the predicted constraints), to the full restraint set, and finally the improvement seen using randomly selected known contacts. I evaluated initial sets of the top 1L and 2L constraints with a minimum separation of six. There is little difference between the different fractions used. Even satisfying 25% of the top 1L contacts already significantly improves folding, although not as much as higher fractions nor as much as the positive control set of restraints. The top 2L constraints are far less accurate and thus contain several examples where smaller fractions of contact restraints perform better than all larger fractions. The difference in performance is minimal and unreliable. All nine proteins potentially benefit from including some fraction of the predicted contact restraints and eight of the nine improve with any fraction of the predicted contact restraints.

Conclusion

Predicted contacts which leverage global position, sequence, and correlation information significantly outperforms DI-only prediction. The best ANNs outperform the best decision trees and the increased accuracy of these methods carries through to protein structure prediction. The average RMSD100 improvement is highest for models generated using contact predictions from the best ANNs as compared to the best decision trees or DI-based methods.

DI from filtered MSA with manually optimized e-values yield much more accurate contact-predictions than traditional methods and most pairs outside the 8Å cutoff are still relatively close to that threshold. However, the highest accuracies require topology-based filtering. Regardless, DI-based constraints without such filtering can still significantly improve BCL::Fold performance. Furthermore, filtering does not necessarily always yield the best BCL::Fold performance. Increased performance with such constraints is substantial but not equivalent to using known contacts. As such, further increases in contact prediction accuracy will further improve fold predictions.

Finally, initial results from the addition of confidence-based scoring suggest that such scoring may improve results and specifically increase the sampling frequency of native-like topologies in those cases. However, the confidence scoring method must appropriately match the accuracy distribution across the L-fraction used. This distribution is highly variable across the benchmark set and requires new methods that are able to predict the accuracy distribution to maximize the benefits of confidence-based scoring.

Methods

Project Overview

The process for all contact predictions begins with a target protein sequence and the aggregation of sufficient homologous protein sequences with sufficient coverage (Lunt et al., 2010a). I then used HHBlits to align these sequences and columns with greater than 30% gaps are

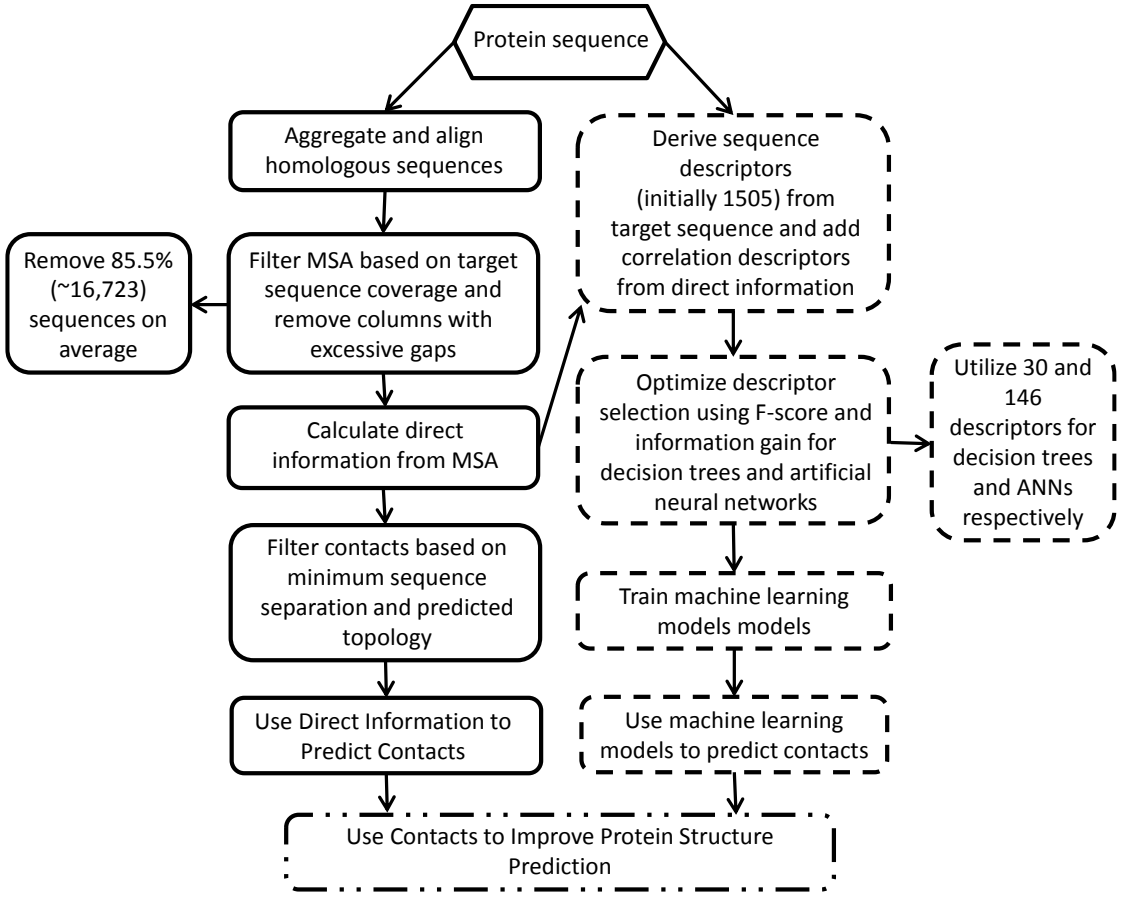


Figure 43: Contact Prediction Flowchart for Both Direct Information and Machine Learning Based Methods

The process overview for producing contacts is above divided into the DI is divided above into the DI based (solid black outline) and machine learning (dashed black outline). Select steps have been quantified in the adjacent sections. Both branches result in sets of predicted contacts, which were then used in combination with BCL::Fold to predict the structures of the 25 membrane proteins in the benchmark set.

removed. One may then calculate DI between the remaining columns. For the prediction of membrane proteins, I also removed soluble domains focusing solely on the transmembrane domain for all proteins in my benchmark set. DI can be used to rank and select the top L-fractions of contacts or one can provide DI for each contact pair position as a descriptor to a machine learning method. In this case I also created aggregate statistics of the aggregated mean, max, standard deviation, and normalized mean for each e-values used to create MSA. These correlation descriptors are combined with sequence information and global sequence position descriptors to create a set of 1,505 descriptors for each contact pair. I then selected from these descriptors an optimal set of 30 descriptors for use with decision trees and 146 for ANNs. For predictions, I used the average prediction across 5 models from each method to rank contact pairs and then used the top L-fractions of those rankings as contact predictions. In the case of a non-zero minimum separation, filtering is performed before the top L-fraction was selected. Contact prediction restraint sets are then provided to BCL::Fold for inclusion during generation of protein structure predictions. These contacts, if sufficiently accurate, guide structure prediction by influencing the scoring of each model produced during the Monte Carlo simulation performed within BCL::Fold.

Multiple Sequence Alignments

I generated MSA using the HHBlits software suite. HHBlits is a hidden markov-based iterative homologous sequence identifier and multiple sequence aligner. The sheer size of the alignments necessary for accurate DI calculation necessitate such an exceptionally fast alignment method (Remmert, Biegert, Hauser, & Söding, 2012). The package also contains several utility

scripts that reformat and filter MSA, as well as renumber PDB files such that they correspond to the generated alignments. I used the following parameters: maximum identity 100, sequence coverage of 70%, two iterations, and e-values at thresholds of 1E-3, 1E-5, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40. I filtered MSA based on these parameters where indicated. I also removed MSA columns with greater than 30% gaps before DI calculation. All sequence alignments were done using the nr20 Uniprot dataset last updated on August 11th, 2012 (Eddy, 2010; Lunt et al., 2010b; Remmert et al., 2012).

Calculating Direct Information

DI calculation begins with accumulating a significant number of evolutionarily diverse but related sequences. Morcos *et al.* use a minimum of 1000 non-redundant sequences after filtering with higher E-value thresholds than default for the HMMer software package. The E-values used are - 1E-3, 1E-5, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40. After alignment, the DI algorithm re-weights element frequencies to reduce the impact of overly similar sequences. One can then compute mutual information from the re-weighted frequency counts (for columns and column pairings). One then applies the maximum entropy principle to compute the direct coupling between sites. Finally, one determines direct information - separating direct from indirect correlations. The direct information is essentially the mutual information between columns that is solely a result of the direct coupling in the prior step. The output is a symmetric matrix of significant pairwise correlation C_{ij}^{ab}. HHBlits scripts re-align contacts by mapping the MSA to the original target structure.

Selection of Restraints for Positive Control

The positive control benchmark runs use random samplings of known contacts based on the solved protein structures documented in each protein's respective PDB file. Cβ to Cβ distances (Cα in the case of glycine) which are within 8Å of one another determine all contacts. After randomization, I create each contact file based on the necessary fraction of L and minimum separation. L is based on the length of the sequence file trimmed to solely the transmembrane region of the protein at multiples of L/10, L/5, L/2, L, and 2L. Minimum separations used include 0, 3, 6, and 12.

Selection of Direct Information Restraints

All pairwise positions included within the final MSA after gaps are removed for columns with greater than 30% gaps are assigned their value from the DI correlation matrix. Pairs are ranked according to the magnitude of the accompanying DI value. Then, depending on the desired L-based cutoff or minimum separation filter, a restraint file is generated beginning with the highest ranked pairs and excluding pairs that violate these parameters. The length of the sequences determined the L values, after trimming down to encompass only the portion of the protein that includes the transmembrane helices. Soluble domains and experimentally added domains to aid in crystallization were removed.

Machine Learning Model (ANN and Decision Tree) Training

Once I determined optimal parameters and descriptors, I trained five models for each set of five proteins using datasets with 15, 5, and 5 proteins forming the training, monitoring, and independent sets respectively. The five models were created by randomizing proteins between the training or monitoring test sets. Data points never appears in more than one set and no data from proteins in the independent set was ever included in either the training or monitoring data sets. The objective function used for training during initial descriptor selection was RMSD and then average enrichment for final training. For decision trees, I attempted to reduce the effect of overtraining by setting the minimum split size to 20. To determine the optimal alpha and eta parameters for the ANNs I performed a grid search settling on an eta of 0.000017 and an alpha of 0.0 as my step size during training was always set to 1. For all correlation descriptors, any undefined values were replaced by the average correlation across the protein sequence. Finally, the average of all five predictions across each set of five models are used to rank all possible pairwise amino acid pairings and the top ranked L-fractions are selected as the contacts for the given protein.

Machine Learning Descriptor Optimization

Decision Trees – Descriptors were optimized by iteratively scoring all descriptors using input sensitivity (with a delta of 1.0) across the 20 models from a five fold cross validation for each set of ranked descriptors. The first scoring indicated that only the top 210 had non-trivial scores. As such I iteratively rescored the top descriptors (starting from this set of 210) with input sensitivity using the best run from the previous stage. The best threshold is set as the new top

descriptor threshold. To decrease the likelihood of removing useful descriptors, I removed no more than half of the descriptors before rescoring at each stage. Thus, I subsequently examined the top 160, 130, and 70 descriptors. For the top 70 the range around the top 30 of this set resulted in the best performance. I used the enrichment average as the objective function and evaluated each set of models generated by calculating the integral of the precision over the range 0.01% to 0.55% of the fraction predicted positive. This range closely captures the contacts predicted when taking the top 1L predictions across all proteins while decreasing the noise present below 0.01%. The modest number of data points results in drastic changes from small perturbations in overall predictions below 0.01%.

ANNs - Descriptors were optimized with a descriptor selection method that was previously introduced to the BCL. At its core, utilizes the weights that comprise the neural network to calculate an approximate derivative for each feature column on the result, specifically, terming the weight matrices between layer *i* and layer *j Mij*, it computes *product(Transpose(Mij))* for each model in the cross validation. Two statistical measures previously implemented in the BCL were used score descriptors: consistency of effect (e.g. does descriptor *i* tend to increase (or decrease) the likelihood of a contact across all models in the cross validation ensemble, or not), and average (pseudo) derivative squared. Each measure was rescaled between 0-0.5, summed, and squared. The rationale is that features that are just noise will tend to have small overall weights and that they should be centered about zero. It is possible for non-noise columns that have approximately 0 distribution about 0 though, so that's why the derivatives squared also helps.

Predicted Contact Restraint Generation

Once an optimal set of machine learning training parameters were determined for DTs and ANNs, I trained a set of 5 models that did not include any information from the proteins to be predicted with each set of models. All possible pairwise contacts for each model are then ranked by the contact predictions averaged from all 5 models and the top L-fraction of the ranking is used as the contact restraints for folding simulations. In the case that a minimum separation is used, all pairs are removed from the ranking before the top L-fraction is selected.

BCL::Fold Membrane Protein Structure Prediction

BCL::Fold creates each model through a Monte Carlo optimization with two stages. It begins with the assembly of models by placing and performing large SSE-based moves. The second stage is focused on the refinement of the generated models and utilizes small amplitude SSE translations and rotations to arrive at a final model. After each SSE move the models are scored using knowledge-based potentials that examine membrane protein topology, environment prediction accuracy, SSE alignment, as well as radius of gyration, amino acid environment, contact order, amino acid clashes, and a loop score among others(Karakaş et al., 2012; B. Weiner, Woetzel, & Karakaş, 2013; Woetzel et al., 2012). If given, contact information is also used to score models. Distributions are generated by repeating this process in parallel 1,000 times and comparisons between runs are done using the average of the top 10 models by RMSD100.

RMSD-RMSD Comparison

RMSD-RMSD points are the average of the top 10 models by RMSD100 from model folding runs of 1,000 models. BCL::Score also filters models before ranking, such that incomplete models which may receive arbitrarily low RMSD100 scores (due to the fact that atoms with largely deviating positions which would elevate the RMSD100 score) are excluded in cases where some SSE are not included in the final model. Error bars are the standard deviations of the average for each 10 model set.

Overview

This protocol capture contains the steps necessary to obtain the results presented in the master's thesis titled "Using Evolutionarily-Based Correlation measures and Machine Learning to Improve Protein Structure Prediction in BCL::Fold" by Pedro Teixeira. While the actual protocol was carried out on every protein within Table 1 of the thesis, this protocol capture only uses 1HZXA as an example for simplification. The BCL software suite is publically available and the license is free for non-commercial users at http://www.meilerlab.org/index.php/bclcommons/show/b_apps_id/1.

Background

SUMMARY

De novo protein structure prediction is a challenge due to the sheer size of the potential search space. One can limit the set of possibilities with long-range contact restraints (positions distant in the primary sequence but known to be in close proximity within the tertiary structure). Most available contact prediction methods achieve accuracies insufficient for de novo protein folding. Direct Information (DI) is a notable exception. DI has been used to determine the structures of some membrane and soluble proteins with large numbers of homologous sequences compiled into deep alignments. However, DI has many limitations. This work documents the usage of machine learning methods to predict contacts more accurately by combining DI with sequence information. In addition, we used predicted contacts to improve the accuracy of protein structure within the Biochemical Library (BCL). This innovative resource will augment the elucidation of traditionally challenging membrane protein structures – specifically larger proteins, which are customarily computationally difficult to address. This protocol capture covers the following:

1. Generating multiple sequence alignments (MSA) and determine DI between all pairwise amino acid sites in a given protein
2. Training machine learning models to predict more accurately long-range protein contacts using DI and other sequence information
3. Predicting long-range contacts using the aforementioned machine learning models
4. Visualizing the contacts generated
5. Leveraging the predicted contacts to enrich for native-like models during de novo prediction using BCL::Fold
6. Initiating iterative folding runs to further refine protein fold models

Protocol

Environment and Directory Setup (Required Before All Other Steps)

```
setup and source cshrc
cd <installation directory>
```

Step	Commands	Comment

1A. Prepare directory and file set	**Create Project Directory with Similar Structure as Example:**
	
	`mkdir data/1HZXA/`
	`mkdir pbs/`
	`mkdir -p training/COMBINED/`
	`mkdir folding/`
	
	Download the BCL: Place the executable in the top level directory of this project as bcl.exe (at the same level as the folding/ training/ data/ pbs/ directories)
	Obtain PDB files:
	Download GPCR crystal structures from the Protein Data Bank at http://www.rcsb.org. Visualize to assist in trimming long non-membrane loops using pymol (and here's a handy script for turning on cartoon and rainbow spectrum):
	`PyMOL>run scripts/pymol.py`
	
	(Set of all trimmed .pdb files used in manuscript is included in trimmed_pdbs/)
	Copy over .pdb and .fasta files with PDBID.<pdb>|<fasta> and include the chain IDDownload the BCL and place the executable in the top level directory of this project (at the same level as the folding/ training/ data/ directories). Files should be named as 1HZXA.pdb, 1HZXA_trim.pdb, and 1HZXA.fasta within data/1HZXA/
	Get secondary structure outputs (within data/PDBID/ directory):
	`runss 1HZXA.fasta`
	`run_octopus.pl --fasta 1HZXA.fasta`
	`runBCLjufo9D 1HZXA.fasta`
		Input:
		Crystal structure PDB files from the Protein Data Bank at http://www.rcsb.org. Subdirectory created to house 1HZXA specific data, improves organization as remaining data files are generated.
		
		All proteins in the paper were trimmed, only the transmembrane domain and short internal loops are used for contact prediction and folding.
		
		*.fasta|.pdb|_trim.pdb
		
		Output:
		1HZXA.ascii, 1HZXA.fasta, 1HZXA.jufo9d_ss, 1HZXA.jufo9d_tmh, 1HZXA.jufo9d_topo, 1HZXA.nnprf, 1HZXA.pdb, 1HZXA.psipred_blast, 1HZXA.psipred_ss, 1HZXA.rdb6Prof, 1HZXA.topo, 1HZXA.ascii6,

		1HZXA.jufo9d, 1HZXA.jufo9d_tm, 1HZXA.jufo9d_tms, 1HZXA.jufo_ss, 1HZXA.octo_topo@, 1HZXA.png, 1HZXA.psipred_horiz, 1HZXA.psipred_ss2, 1HZXA.rdbProf
1B. Create a pdbs.ls file that lists all PDBIDs on which scripts will run	Make a pdbs.ls file for other scripts to use so they can run through all PDBIDs that you are using (format is PDBID with chain id separated by newlines: 1HZXA 1GZMA etc..	**Output (located in top level project directory):** pdbs.ls
2A. Download and prepare the BCL	1. cd into the protocol capture directory 2. install the bcl directly into this directory (the extracted protocol-capture directory) 3. When following the protocol capture directions, replace bcl.exe some:Application with ./bcl/bcl-apps-release-static.exe some:Application	**Output:** Unpacked BCL directory in protocol home
2B. Download and prepare the HHSuite	Download and unpack HHBlits such that the directory path for scripts is as such link to software: `hhsuite/hhsuite-2.0.15/scripts/renumberpdb.pl`	**Output:** Unpacked hhsuite/ directory in project root

GENERATE MSA AND DI FOR ALL AMINO ACID SITE PAIRS

Identify Sequences and Create MSA

I generated MSA using the HHBlits software suite. HHBlits is a hidden markov-based iterative homologous sequence identifier and multiple sequence aligner. The sheer size of the alignments necessary for accurate DI calculation necessitate such an exceptionally fast alignment method (Remmert, Biegert, Hauser, & Söding, 2012). The package also contains several utility scripts that reformat and filter MSA, as well as renumber PDB files such that they correspond to the generated alignments. I used the following parameters: maximum identity 100, sequence coverage of 70%, two iterations, and e-values at thresholds of 1E-3, 1E-5, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40. I filtered MSA based on these parameters where indicated. I also removed MSA columns with greater than 30% gaps before DI calculation. All sequence alignments were done using the nr20 Uniprot dataset last updated on August 11th, 2012 (Eddy, 2010; Lunt, Szurmant, Procaccini, Hoch, Hwa, & Weigt, 2010b; Remmert et al., 2012). Calls across many proteins are simplified using python scripts as shown below.

Step	Text	Commands	Comment
1A. Run script to get all alignments for your fasta with filtering (true)	The 25 membrane proteins listed in Table 1 are a diverse set of non-trivial (having more than four transmembrane helices) α-helical transmembrane proteins with more than 1000 homologous sequences of	`python scripts/make_pbs_to_run_hhblits_for_all_E.py 1HZXA.fasta 1HZXA_RUN1 data/1HZXA/ pbs/ true 70`	**Input:** 1HZXA.fasta **Output (in pbs/):** 1HZXA_RUN1_cov_70_FILTERED_01.pbs

	sufficient coverage.		
1B. Run script to get all alignments for your fasta without filtering (false)	I generated MSA using the HHBlits software suite. HHBlits is a hidden markov-based iterative homologous sequence identifier and multiple sequence aligner. The sheer size of the alignments necessary for accurate DI calculation necessitate such an exceptionally fast alignment method (Remmert, Biegert, Hauser, & Söding, 2012). The package also contains several utility scripts that reformat and filter MSA, as well as renumber PDB files such that they correspond to the generated alignments.	`python scripts/make_pbs_to_run_hhblits_for_all_E.py 1HZXA.fasta 1HZXA_RUN1 data/1HZXA/ pbs/ false 70`	**Input:** 1HZXA.fasta **Output:** 1HZXA_RUN1_cov_70_UNFILTERED_01.pbs

1B. Run pbs scripts	I used the following parameters: maximum identity 100, sequence coverage of 70%, two iterations, and e-values at thresholds of 1E-3, 1E-5, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40.	**Run pbs scripts to generate MSA outputs:**
		Due to large resource requirements it is often best to run MSA generation and DI calculation on a cluster using the generated pbs files above.
		`ssh vmplogin <or> ssh piranha`
		`cd pbs/`
		`qsub 1HZXA_RUN1_cov_70_FILTERED_01.pbs`
		`qsub 1HZXA_RUN1_cov_70_UNFILTERED`
			Input: n/a
			
			Output: (The following set of files will be produced for each e-value cutoff: 1E-03, 1E-05, 1E-10, 1E-15, 1E-20, 1E-30, 1E-40) 1HZXA_RUN1_e_1E-05_cov_70_FILTERED.a3m

		_01.pbs	1HZXA_RUN1_e_1E-05_cov_70_FILTERED.log 1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED.a3m 1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED.log
2A. Convert PDBI.pdb to BCL type pdb file with PDBID.pdb name that includes chain ID (if you didn't start with a BCL PDB or for intermediately created files)	n/a	**Run within appropriate data/SUBDIRECTORY/ :** `bcl.exe PDBConvert 1HZX.pdb -fasta -convert_to_natural_aa_type -split_ensemble -output_prefix 1HZX -bcl_pdb`	**Input:** 1HZX.pdb **Output:** 1HZX_0bcl.pdb
2B. Move over output .pdb file to better naming	n/a	`mv 1HZX_0bcl.pdb 1HZXA_trim.pdb`	**Input:** 1HZX_0bcl.pdb **Output:** 1HZXA_trim.pdb
3. Fix top line of the .a3m files for all the alignments such that the naming is correct which is necessary for running renumber pdb perl script	n/a	`python scripts/fixtopline.py --dir=data/`	**Input:** **Output:**

Calculate DI for all Amino Acid Site Pairings

DI calculation begins with accumulating a significant number of evolutionarily diverse but related sequences. Morcos et al. use a minimum of 1000 non-redundant sequences after filtering with higher E-value thresholds than default for the HMMer software package. The E-values used are - 1E-3, 1E-5, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40. After alignment, the DI algorithm re-weights element frequencies to reduce the impact of overly similar sequences. One can then compute mutual information from the re-weighted frequency counts (for columns and column pairings). One then applies the maximum entropy principle to compute the direct coupling between sites. Finally, one determines direct information - separating direct from indirect correlations. The direct information is essentially the mutual information between columns that is solely a result of the direct coupling in the prior step. The output is a symmetric matrix of significant pairwise correlation Cijab. HHBlits scripts re-align contacts by mapping the MSA to the original target structure.

4A. Obtain and prepare matlab Direct Coupling Analysis (DCA) script/binary	n/a	To obtain a copy of the script one has to contact via the excerpt provided below and abide by the following
		`%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%`
		`%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%`
		`% Copyright for this implementation:`
		`%              2011/12 - Andrea Pagnani and Martin Weigt`
		`%`
		`andrea.pagnani@gmail.com`
		`%`
		`martin.weigt@upmc.fr`
		`%`
		`% Permission is granted for anyone to copy, use, or modify this`
		`% software and accompanying documents for any uncommercial`
			Input:
			DCA.m
			run_DCA_50.sh
			
			Output:
			DCA_50.m
			DCA_50

```
% purposes, provided this copyright notice is retained, and note is
% made of any changes that have been made. This software and
% documents are distributed without any warranty, express or
% implied. All use is entirely at the user's own risk.
%
% Any publication resulting from applications of DCA should cite:
%
%     F Morcos, A Pagnani, B Lunt, A Bertolino, DS Marks, C Sander,
%     R Zecchina, JN Onuchic, T Hwa, M Weigt (2011), Direct-coupling
%     analysis of residue co-evolution captures native contacts across
%     many protein families, Proc. Natl. Acad. Sci. 108:E1293-1301.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
In addition, I have set the pseudocount weight to 0.5 and renamed the function and scriptname to DCA_50 and DCA_50.m. I have also added a stat output to be used later as a descriptor by including the following code after one computes true frequencies:
% Added stat output - Pedro Teixeira
    fprintf('### N = %d M = %d Meff = %.2f q = %d, L = %d\n', N,M,Meff,q,L);
    statstring = sprintf('### N = %d M = %d Meff = %.2f q = %d, L = %d', N,M,Meff,q,L);


    % Output N, M, Meff, and q to file for future processing
    statfilename = strtok( outputfile, '.');
    statfilename = [ statfilename, '.statlog'];
```

		statfp = fopen(statfilename, 'w');
		tempout = sprintf('%d,%d,%.2f,%d,%d', N,M,Meff,q,L);
		fprintf(statfp, '%s\n', statstring);
		fprintf(statfp, '%s\n', tempout);
		fclose(statfp);Lastly, compile the script into a matlab binary named DCA_50 - which is then used by the provided script script/run_DCA_50.sh (additional details found in scripts/run_DCA_script_readme.txt). The compilation step is necessary to run on a cluster regardless of provided MATLAB libraries/licenses.	

4B. Generate and Run PBS files to generate DI and scoring files	I calculated DI for all 25 membrane proteins listed in Table 1 for both filtered and unfiltered MSA. The filtering process removes sequences that individually do not align to cover at least 70% percent of the original target sequence used to create the alignment. This is the same cutoff used by Hopf et al. to determine membrane protein structure for the same set of membrane proteins (Hopf, Colwell, Sheridan, Rost, Sander, & Marks, 2012a). In both cases, I used e-value cutoffs for	python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-03_cov_70_FILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-05_cov_70_FILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-10_cov_70_FILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-15_cov_70_FILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-20_cov_70_FILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-30_cov_70_FILTERED.a3m	Bump up time/memory for DI_01_1HZXA_RUN1_e_1E-03_cov_70_UNFILTERED_GAPSREMOVED_01.pbs to 22GB and 9hrs due to extra size outside of approximation algorithm's range and cluster constraints **Input:** 1HZXA.pdb 1HZXA_RUN1_e_1E-XX_cov_70_[UN]FILTERED.a3m **Output:** DI_01_1HZXA_RUN1_e_1E-03_cov_70_FIL

	sequence aggregation of 1E-03, 1E-05, 1E-10, 1E-15, 1E-20, 1E-30, and 1E-40. In addition, Hopf et al. also provided a set of “optimal e-values” for prediction within the work.	1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-40_cov_70_FILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-03_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-10_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-15_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-20_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py 1HZXA_RUN1_e_1E-30_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01 python scripts/make_pbs_for_contact_and_scoring_from_alignment_and_pdb.py	TERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-15_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-40_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-20_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-10_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-30_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-03_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-

		1HZXA_RUN1_e_1E-40_cov_70_UNFILTERED.a3m 1HZXA.pdb data/1HZXA/ pbs/ DI_01	15_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-40_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-20_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-10_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-30_cov_70_UNFILTERED_GAPSREMOVED_01.pbs
5. Run generated DI and scoring calculation pbs scripts on cluster	After alignment, the DI algorithm re-weights element frequencies to reduce the impact of overly similar sequences. One can then compute mutual information from the re-weighted	ls DI_01*.pbs | awk '{system("qsub "$1)}'	Some sequences may require more time or RAM than the above script's formula allots, in which case you may have to increase one or both to ensure the process

	frequency counts (for columns and column pairings). One then applies the maximum entropy principle to compute the direct coupling between sites. Finally, one determines direct information - separating direct from indirect correlations.		runs to completion **Input:** DI_01_1HZXA_RUN1_e_1E-03_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-15_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-40_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-20_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-10_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-30_cov_70_FILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-

			03_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-15_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-40_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-20_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-10_cov_70_UNFILTERED_GAPSREMOVED_01.pbs DI_01_1HZXA_RUN1_e_1E-30_cov_70_UNFILTERED_GAPSREMOVED_01.pbs **Output:** DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED_sep_4_.RR_MI_DCA

			DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.statlog DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.a3m DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.corr_mat_bcl DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.dca DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED_renumbered_1HZXA_bcl.pdb DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED_renumbered_1HZXA.pdb DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.RR DI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.RR_MI_DCA DI_01_1HZXA_

			RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED.scoreDI_01_1HZXA_RUN1_e_1E-05_cov_70_FILTERED_GAPSREMOVED_sep_4_.RR DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.a3m DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.corr_mat_bcl DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.dca DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED_renumbered_1HZXA_bcl.pdb DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED_renumbered_1HZXA.pdb DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.RR

			DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.RR_MI_DCA DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.score DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED_sep_4_.RR DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED_sep_4_.RR_MI_DCA DI_01_1HZXA_RUN1_e_1E-05_cov_70_UNFILTERED_GAPSREMOVED.statlog
6. Create BCL correlation files	(see step 5)	`python scripts/dca_to_corrmat_converter_GENERAL.py data/1HZXA/`	**Input:** All .dca files in the given subdirectory **Output:** Matching .corr_mat_bcl files for each given .dca file
7. Run filename preparation	n/a	When all MSA have been created run the following script to create appropriately named versions for downstream use (include the PDBID and E-value to be used as the	**Input:** **Output:** Within

script		optimal threshold)
		Note:
		Run from within training/COMBINED/
		`Usage general:python prep_and_ss_link_msa_and_corrmat_filenames.py file_prefix PDBID 1E-opt_value /data/directory/ /target/directory/`
		`Usage specific (run from within training/COMBINED/):`
		`python ../../scripts/prep_and_ss_link_msa_and_corrmat_filenames.py DI_01 1HZXA 1E-20 data/ training/COMBINED/`
			training/COMBINED/1HZXA/1HZXA_1E-03_FILT.corr_mat_bcl@1HZXA_1E-10_FILT.mfasta@ 1HZXA_1E-20_UNFILT.corr_mat_bcl@1HZXA_1E-40_UNFILT.mfasta@1HZXA.jufo9d_tmh@1HZXA_OPT_FILT.mfasta@1HZXA.rdb6Prof@ 1HZXA_1E-03_FILT.mfasta@ 1HZXA_1E-10_UNFILT.corr_mat_bcl@1HZXA_1E-20_UNFILT.mfasta@1HZXA.ascii@1HZXA.jufo9d_tms@1HZXA_OPT_UNFILT.corr_mat_bcl@1HZXA.rdbProf@ 1HZXA_1E-03_UNFILT.corr_mat_bcl@1HZXA_1E-10_UNFILT.mfasta@1HZXA_1E-30_FILT.corr_mat_bcl@1HZXA.ascii6@1HZXA.jufo9d_topo@1HZXA_OPT_UNFILT.mfasta@1HZXA.topo@

			1HZXA_1E-03_UNFILT.mfasta@ 1HZXA_1E-15_FILT.corr_mat_bcl@ 1HZXA_1E-30_FILT.mfasta@ 1HZXA_contact.bin 1HZXA.jufo_ss@ 1HZXA.pdb@ 1HZXA_1E-05_FILT.corr_mat_bcl@ 1HZXA_1E-15_FILT.mfasta@ 1HZXA_1E-30_UNFILT.corr_mat_bcl@ 1HZXA.fasta@ 1HZXA.mfasta@ 1HZXA.png@ 1HZXA_1E-05_FILT.mfasta@ 1HZXA_1E-15_UNFILT.corr_mat_bcl@ 1HZXA_1E-30_UNFILT.mfasta@ 1HZXA_full_dataset.bin 1HZXA.nnprf@ 1HZXA.psipred_blast@ 1HZXA_1E-05_UNFILT.corr_mat_bcl@ 1HZXA_1E-15_UNFILT.mfasta@ 1HZXA_1E-40_FILT.corr_mat_bcl@

			1HZXA.jufo9d@ 1HZXA_noncontact.bin 1HZXA.psipred_horiz@ 1HZXA_1E-05_UNFILT.mfasta@ 1HZXA_1E-20_FILT.corr_mat_bcl@ 1HZXA_1E-40_FILT.mfasta@ 1HZXA.jufo9d_ss@ 1HZXA.octo_topo@ 1HZXA.psipred_ss@ 1HZXA_1E-10_FILT.corr_mat_bcl@ 1HZXA_1E-20_FILT.mfasta@ 1HZXA_1E-40_UNFILT.corr_mat_bcl@ 1HZXA.jufo9d_tm@ 1HZXA_OPT_FILT.corr_mat_bcl@ 1HZXA.psipred_ss2@

Train Contact Prediction Models and Score Descriptors (F-Score, Information Gain, and Input Sensitivity)

Once I determined optimal parameters and descriptors, I trained five models for each set of five proteins using datasets with 15, 5, and 5 proteins forming the training, monitoring, and independent sets respectively. The five models were created by randomizing proteins between the training or monitoring test sets. Data points never appears in more than one set and no data from proteins in the independent set was ever included in either the training or monitoring data sets. The objective function used for training during initial descriptor selection was RMSD and then average enrichment for final training. For decision trees, I attempted to reduce the effect of overtraining by setting the minimum split size to 20. To determine the optimal alpha and eta parameters for the ANNs I performed a grid search settling on an eta of 0.000017 and an alpha of 0.0 as my step size during training was always set to 1. For all correlation descriptors, any undefined values were replaced by the average correlation across the protein sequence. Finally, the average of all five predictions across each set of five models are used to rank all possible pairwise amino acid pairings and the top ranked L-fractions are selected as the contacts for the given protein.

Step	Text	Commands	Comment
1. Prepare necessary files	I trained five models for each set of five proteins using datasets with 15, 5, and 5 proteins forming the training, monitoring,	Create object files - input, output, and ID (.obj files) with the descriptors you want to use (examples given at -training/) Create a msa_stats.csv file if you want to include that information/descriptor, data given below for an example file Format summary (data below) = Meff, M, Coverage, L For each of the above the sub-order is Filtered at all	**Input:** **Output:** initial_code_inputCOMBINED.obj initial_code_output_extensive.obj initial_code_idTEST.obj

	and independent sets respectively.	E-values (decreasing), Unfiltered at all E-values (decreasing) - example at: training/msa_stats.csv	msa_stats.csv

2. Generate binaries of data for each protein	(see step 1)	`Usage general: python generate_datasets_COMBINED.py /run/directory/ PDBID1,PDBID2,PDBID3,PDBID4,... /obj/file/directory/ bcl_filename.exe num_threads input.obj output.obj id.obj"`
		
		`Usage specific (run from within training/COMBINED/):python ../../scripts/generate_datasets_COMBINED.py ../training/ 1HZXA ../training/ ../../bcl.exe 2 initial_code_inputCOMBINED.obj initial_code_output_extensive.obj initial_code_idTEST.obj`
			Input:
			Data files and links within training/COMBINED/SUBDIR/ and initial_code_inputCOMBINED.obj initial_code_output_extensive.obj initial_code_idTEST.obj
			
			Output:
			1HZXA_full_dataset.bin

3A. Split binaries into contact and noncontact	(see step 1)	Create the separated binaries for each protein, one contacts, one non-contacts (useful for balancing and separating by protein for different training and test datasets)
		`Usage general:python split_into_contact_non_contact.py /run/directory/ PDBID1,PDBID2,PDBID3,PDBID4,... bcl_filename.exe num_threads`
		
		`Usage specific (run from within training/COMBINED/):`
		`python ../../scripts/split_into_contact_non_contact.py training/ 1HZXA ../bcl.exe 6`
			Input:
			1HZXA_full_dataset.bin
			
			Output:
			1HZXA_contact.bin
			1HZXA_noncontact.bin

3B. Make sure you have at least 3 proteins to generate training,	I trained five models for each set of five proteins using datasets with 15, 5, and 5 proteins	Training expects at least 3 proteins, one for each of training, monitoring, and test. There are two dummy proteins based on 1HZXA for the sake of example training dividing up by protein (one training, one monitoring, one test, make sure these are non-overlapping sets for actual model training) `python (run from within`	**Input:** **Output:**

monitoring, and test data	forming the training, monitoring, and independent sets respectively.	training/COMBINED/) ../../scripts/make_fake_PDBID.py 1HZXA 1HZXB training/COMBINED/python ../../scripts/make_fake_PDBID.py 1HZXA 1HZXC training/COMBINED/	
4. Create parameter input files based on examples given	(see step 3B)	Input example files are given in training/COMBINED/aggregate_noncontacts.input and training/COMBINED/aggregate_contacts.input	**Input:** **Output:** aggregate_noncontacts.input aggregate_contacts.input
4B. Aggregate binaries into one set of all contact and noncontacts respectively	(see step 3B)	Aggregate the binaries for each protein, into only on all protein contacts and one all protein non-contacts (useful for comparing and analyzing datasets, also simplifies some test training since multiple files don't have to be manipulated/combined via the command line): Usage general: bcl.exe descriptor:GenerateDataset @aggregate_noncontacts.input > compare_noncontact_DATE.log bcl.exe descriptor:GenerateDataset @aggregate_contacts.input > compare_contact_DATE.log Usage specific (run from within training/COMBINED/): bcl.exe descriptor:GenerateDataset @aggregate_noncontacts.input > compare_noncontact_2013-11-30.log bcl.exe descriptor:GenerateDataset @aggregate_contacts.input > compare_contact_2013-11-30.log	**Input:** aggregate_noncontacts.input aggregate_contacts.input Given .bin files as listed in *.input files **Output:** all_noncontact_dataset.bin all_contact_dataset.bin
5. Determine F-score/Information Gain (can use these	We scored descriptors using information gain and F-score to	Make sure you have an appropriate object file e.g. initial_code_outputCONTACT.obj in training/ Update paths and filenames in score_combined.sh if necessary bash score_combined.sh	**Input:** all_noncontact_dataset.bin all_contact_dataset.bin initial_code_outputCO

metrics to filter and optimize which descriptors are used for training)	determine their individual potential for contact prediction.		NTACT.obj **Output:** all_comb_fs_ig.score all_comb_fs.score all_comb_ig.score

6. Train ANNs separating by proteins and making the final prediction on each protein using only other proteins for training/monitoring (This is obviously just a test case as the other proteins are just dummy copies of the original)	Once an optimal set of machine learning training parameters were determined for DTs and ANNs, I trained a set of 5 models that did not include any information from the proteins to be predicted with each set of models.	`Usage general:`
		`python scriptname directory eta alpha #node window_size min_separation_train min_separation_test train_num monitor_num independent_num iteration_num ARCHIVE_DIRNAME Mode PDBID_PDBID_PDBID /path/input_code.obj /path/result_code.obj`
		
		`Usage specific (run from within training/):`
		`python2.7 ../scripts/automate_focused_membrane_training_separate_independent_selected_balance_ann.py COMBINED/ 0.000017 0.0 8 9 1 12 1 1 1 2 ANN_trainsep1_testsep12_tmi_1-1-1_iter2_ALLDESC_contact ANN 1HZXA initial_code_inputWEIGHTS_OPT_DESC_146.obj initial_code_outputCONTACT.obj`
			Input:
			initial_code_inputWEIGHTS_OPT_DESC_146.obj
			initial_code_outputCONTACT.obj
			Output:
			commandline_0.input
			monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot_txt
			commandline_1.input
			monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot_txt.png
			indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl
			monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.table
			indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot
			monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl
			indep_1000_w9_eta_0.000017_alpha_0.0_8n

			_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot_txt monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot_txt.png monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot_txt indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.table monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot_txt.png indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.table indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot run1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.log indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_

			ANN_1HZXA_01.bcl.gnuplot_txt run1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.log indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot_txt.png stats1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_.log indep_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.table train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl model000000.descriptor train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot model000000.info train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot_txt model000000.model train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot_txt.png model000001.descriptor train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.table

			model000001.info train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl model000001.model train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot model.result train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot_txt monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.bcl.gnuplot_txt.png monitor_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_00.bcl.gnuplot train_1000_w9_eta_0.000017_alpha_0.0_8n_tr_1_mon_1_test_1_ANN_1HZXA_01.table
7. Determine input sensitivity	Descriptors were optimized by iteratively scoring all descriptors using input sensitivity (with a delta of 1.0) across the 20 models from a five	Script code is included in training/scoring_combined.sh similar to F-Score and Information Gain to handle many parameters and models. Uncomment/comment code as desired to calculate descriptor scores with each method. `(run from within training/)bcl.exe descriptor:ScoreDataset -feature_labels ./initial_code_inputWEIGHTS_OPT_DESC_146.obj -result_labels ./initial_code_outputCONTACT.obj -source 'Chunks( number chunks=1,`	**Input:** initial_code_inputWEIGHTS_OPT_DESC_146.obj initial_code_outputCONTACT.obj all_noncontact_dataset.bin all_contact_dataset.bin model* in given directory

	fold cross validation for each set of ranked descriptors. The first scoring indicated that only the top 210 had non-trivial scores. As such I iteratively rescored the top descriptors (starting from this set of 210) with input sensitivity using the best run from the previous stage. The best threshold is set as the new top descriptor threshold. To decrease the likelihood of removing useful descriptors, I removed no more than half of the descriptors before rescoring at each stage. Thus, I subsequently examined the top 160, 130, and 70 descriptors.	`chunks="[0, 10000)", dataset=Combined(Subset(filename = './COMBINED/all_contact_dataset.bin'), Subset(filename = './COMBINED/all_noncontact_dataset.bin')))' -output all_comb_is_ann_10000_top146.score -score 'InputSensitivity(delta=0.1,storage=File(directory=./COMBINED/ARCHIVE/ANN_trainsep1_testsep12_tmi_1-1-1_iter2_ALLDESC_contact_1HZXA/,prefix=model),weights=ScoreDerivativeEnsemble(consistency=0,consistencybest=0,square=0,absolute=0,utility=0,average=0,balance=1,categorical=1))' -message_level Verbose` Scores were used to manually cut down total descriptors used based on the criteria described in the manuscript text. Descriptor files obtained after each round of scoring are included in training/intermediate_descriptors/	COMBINED/ARCHIVE/ANN_trainsep1_testsep12_tmi_1-1-1_iter2_ALLDESC_contact_1HZXA/ **Output:** all_comb_is_ann_10000_top146.score

	For the top 70 the range around the top 30 of this set resulted in the best performance.		

PREDICTING CONTACTS AND GENERATING CONTACT RESTRAINT FILES

Once an optimal set of machine learning training parameters were determined for DTs and ANNs, I trained a set of 5 models that did not include any information from the proteins to be predicted with each set of models. All possible pairwise contacts for each model are then ranked by the contact predictions averaged from all 5 models and the top L-fraction of the ranking is used as the contact restraints for folding simulations. In the case that a minimum separation is used, all pairs are removed from the ranking before the top L-fraction is selected.

Step	Text	Commands	Comment
1. Prepare directory by creating links for all mfasta files, corr_mat, and SS prediction files in folding/ directory	All possible pairwise contacts for each model are then ranked by the contact predictions averaged from all 5 models and the top L-fraction of the ranking is used as the contact restraints for folding simulations.	`(run from within folding/)` `python ../scripts/prep_and_ss_link_msa_and_corrmat_filenames.py DI_01 1HZXA 1E-20 ../data/ ../folding/`	**Input:** **Output:**

Predict using machine learning models (step 2A)

Predictions may be done in two ways. The first uses the models generated previously, which take into account DI as well as sequence information. The second method only uses ranked DI to create contact restraint files.

Note - Multiple proteins can be addressed with an awk line as written below (modify for model-predicted contacts):

```
(run from within folding/)
cat ../pdbs.ls | awk '{ system("python
../scripts/convert_simple_DI_csv_to_PROCESSED_modified_RR.py --
input_path "$1"/"$1"_DI_UNFILT_SIMPLE.csv --output
"$1"_UNFILT_SIMPLE_UNprocessed.RR")}'
```

2A.I. Create or update prediction .obj files	(see step 1)	Object files should be set such that desired outputs and paths to generated models are correct. Examples given at training/initial_code_pred.obj Model path in this case is: training/COMBINED/ARCHIVE/ANN_trainsep1_testsep12_tmi_1-1-1_iter2_ALLDESC_contact_1HZXA/	Models used should not have been trained using any data from the contacts/noncontacts of proteins one is about to predict! *Input:* **Output:** initial_code_pred.obj
2A.II. Generate .csv files that will be parsed for model-generated contact restraint files	(see step 1)	`(run from within folding/)\\bcl.exe GenerateDataset -source 'ProteinDirectory("./1HZXA/")' -feature_labels ../training/initial_code_inputWEIGHTS_OPT_DESC_146.obj -result_labels initial_code_pred.obj -id_labels initial_code_idTEST.obj -output ./1HZXA/1HZXA_ANN_TOP146_pred.csv`	Models used should not have been trained using any data from the contacts/noncontacts of proteins one is about to predict! *Input:* initial_code_inputWEIGHTS_OPT_DESC_146.obj initial_code_pred.obj initial_code_idTEST.obj data within ./1HZXA/ **Output:** 1HZXA_ANN_TOP146_pred.csv
2A.III. Parse .csv files into modified restraint (.RR) files	(see step 1)	`(run from within folding/)python2.7 ../scripts/convert_predicted_csv_to_modified_RR.py 1HZXA_ANN_TOP146_pred.csv 1HZXA_FILT_pred.RR`	Models used should not have been trained using any data from the contacts/noncontacts of proteins one is about to predict! *Input:* 1HZXA_ANN_TOP146_pred.csv **Output:** 1HZXA_FILT_pred.RR

Predict using only ranked DI (step 2B)

2B.I. Create .csv files necessary for parsing out DI files	The top L-fraction of the ranking is used as the contact restraints for folding simulations. In the case that a minimum separation is used, all pairs are removed from the ranking before the top L-fraction is selected.	The following line uses the descriptor generation process with the model to create predictions for the contacts which are then used downstream to sort the contacts and take the top L-fraction as the positive cases for folding constraints `(run from within folding/)bcl.exe descriptor:GenerateDataset -source 'ProteinDirectory(./1HZXA)' -feature_labels input_simple.obj -result_labels output.obj -id_labels id.obj -output ./1HZXA/1HZXA_DI_UNFILT_SIMPLE.csv` Make sure you have the input, id, and output object files in folding/ they are needed to create the appropriate output .csv files for parsing.	**Input:** input_simple.obj id.obj output.obj **Output:** 1HZXA/1HZXA_DI_UNFILT_SIMPLE.csv
2B.II. Parse .csv files into modified restraint (.RR) files	(see step 2B.I.)	`(run from within folding/)python ../scripts/convert_simple_DI_csv_to_PROCESSED_modified_RR.py --input_path 1HZXA/1HZXA_DI_UNFILT_SIMPLE.csv --output 1HZXA_UNFILT_SIMPLE_UNprocessed.RR`	**Input:** 1HZXA_DI_UNFILT_SIMPLE.csv **Output:** 1HZXA_UNFILT_SIMPLE_UNprocessed.RR

Trim and score contact restraint files for analysis and use in protein folding

3. Trim contact files to L-based lengths	The top L-fraction of the ranking is used as the contact restraints for	This example is done with model-predicted contacts and no .RR file suffix
		`Usage general:`
		`python trim_contact_files_to_L.py /directory/with/trimmed/fasta/data/ /directory/to/process/with/PDBIDs/ input_RR_suffix <output_suffix_for_RR>`
		
		`Usage specific (predicted contacts example, run from within folding/):`
		`python ../scripts/trim_contact_files_to_L.py ../data/ ./ FILT_pred`
			Input:
			
			Output:

	folding simulations.		

4. Score contact files for analysis and visualization	n/a	Creates a copy with the _SCORE.RR suffix of all contact files scored (uses PDB files to calculate running accuracy, true positives/false positives, good for analysis of output results, also necessary for visualization). Requires directory to hold aggregated data across proteins
		`mkdir aggregated_data/`
		`Usage general:`
		`python score_contact_files.py /dir/with/trimmed/fastas/ /run/dir/with/PDBID/SUBDIRs/ /dir/aggregated_data/`
		
		`Usage specific (run from within folding/):`
		`python ../scripts/score_contact_files.py ../data/ ./ ./aggregated_data/`
			Input: All *.RR files in the given directory
			
			Output: Matching *_SCORED.RR file for each .RR file found

FOLDING PROTEINS USING CONTACT RESTRAINT FILES

BCL::Fold creates each model through a Monte Carlo optimization with two stages. It begins with the assembly of models by placing and performing large SSE-based moves. The second stage is focused on the refinement of the generated models and utilizes small amplitude SSE translations and rotations to arrive at a final model. After each SSE move the models are scored using knowledge-based potentials that examine membrane protein topology, environment prediction accuracy, SSE alignment, as well as radius of gyration, amino acid environment, contact order, amino acid clashes, and a loop score among others(Karakaş et al., 2012; Weiner, Woetzel, Karakaş, Alexander, & Meiler, 2013; Woetzel et al., 2012). If given, contact information is also used to score models. Distributions are generated by repeating this process in parallel 1,000 times and comparisons between runs are done using the average of the top 10 models by RMSD100.

1. Prepare files for protein folding		You will need the following files in your folding directory: assembly_01_contact.scoreweights assembly_02_contact.scoreweights assembly_03_contact.scoreweights assembly_04_contact.scoreweights assembly_05_contact.scoreweights refinement_contact.scoreweights pred_stages_with_rest.txt Update pred_stages_with_rest.txt file to point to the correct scoreweights files Change the script "create_fold_with_rest_pred.pl" such that bcl.exe	**Input:** assembly_01_contact.scoreweights assembly_02_contact.scoreweights assembly_03_contact.scoreweights assembly_04_contact.scoreweights assembly_05_contact.scoreweights refinement_contact.scoreweights pred_stages_with_rest.txt **Output:**

		some:Application becomes **./bcl/bcl-apps-release-static.exe some:Application -histogram_path ./bcl/histogram/ -models_path ./bcl/models/** Ensure correct data directory in the create_fold_with_rest_pred.pl script	
2. Generate .pbs script files to run the protein folding	BCL::Fold creates each model through a Monte Carlo optimization with two stages. It begins with the assembly of models by placing and performing large SSE-based moves. The second stage is focused on the refinement of the generated models and utilizes small amplitude SSE translations and rotations to arrive at a final model. After each SSE move the models are scored using knowledge-based potentials that examine membrane protein topology, environment prediction accuracy, SSE alignment, as well as radius of gyration, amino acid environment, contact order, amino acid clashes, and a loop score among others(Karakaş et al., 2012; Weiner, Woetzel, Karakaş, Alexander, & Meiler, 2013; Woetzel et al., 2012).	Run for L-10, L-5, L-2, 1L, 2L, and 3L fractions of L and minsep6 and minsep12 `(run from within folding/)\\cat ../pdbs.ls | awk '{system("perl ../scripts/create_fold_with_rest_pred.pl 1.0 20 50 "$1" ../folding/ L-10_minsep6 L-10_minsep6 8 12")}'`	**Input:** **Output (in 1HZXA/build_L-10_minsep6/pbs/):** pbs_1HZXA_L-10_minsep6_0.pbs pbs_1HZXA_L-10_minsep6_16.pbs pbs_1HZXA_L-10_minsep6_4.pbs pbs_1HZXA_L-10_minsep6_10.pbs pbs_1HZXA_L-10_minsep6_17.pbs pbs_1HZXA_L-10_minsep6_5.pbs pbs_1HZXA_L-10_minsep6_11.pbs pbs_1HZXA_L-10_minsep6_18.pbs pbs_1HZXA_L-10_minsep6_6.pbs pbs_1HZXA_L-10_minsep6_12.pbs pbs_1HZXA_L-10_minsep6_19.pbs pbs_1HZXA_L-10_minsep6_7.pbs pbs_1HZXA_L-10_minsep6_13.pbs pbs_1HZXA_L-10_minsep6_1.pbs pbs_1HZXA_L-10_minsep6_8.pbs pbs_1HZXA_L-10_minsep6_14.pbs pbs_1HZXA_L-10_minsep6_2.pbs pbs_1HZXA_L-10_minsep6_9.pbs pbs_1HZXA_L-10_minsep6_15.pbs pbs_1HZXA_L-10_minsep6_3.pbs
3. Submit .pbs	n/a	`(run from within folding/)find . –`	Check to make sure a single script command runs locally as well as

script files to the cluster		`maxdepth 4 -name "?????/build_L-10_minsep6/pbs/*.pbs" -type f -exec qsub {} \; > submit_all_build_L-10_minsep6.log`	on the given cluster to ensure the paths are all correct. Should at least get to the minimization step before stopping the test run.**Input:** *.pbs files in folding/SUBDIR/pbs/ .RR contact restraint file chosen **Output:** *.out files in folding/SUBDIR/pbs/ *.pdb fold model files

4. Analyze output models	(see step 3)	Create list of model filenames:
		`(run from within folding/)cat ../pdbs.ls | awk '{ system("ls "$1"/pdbs/"$1"build_L-10_minsep6_?_final_*.pdb > "$1"/"$1"_L-10_minsep6_filenames.ls")}'`
		Get top 10 by RMSD100
		`(run from within folding/)cat ../pdbs.ls | awk '{system("../bcl.exe FoldAnalysis -table_from_file ./"$1"/scores_L-10_minsep6.ls -sort RMSD100 -output_value RMSD100 10 "$1"_L-10_minsep6_top10_RMSD.out")}'`
		Run general analysis and visualization (make sure path is set correctly within analyze_folding_run_general.py)
		`python2.7 scripts/analyze_folding_run_general.py`
			Input:
			pdbs.ls
			*.pdb resulting models
			Output:
			pngs/1HZXA_L-10_minsep6_compare_memmodel_0_0.png
			pngs/1HZXA_L-10_minsep6_mem.pml
			pngs/1HZXA_L-10_minsep6_sum_dist.gnuplot.png
			pngs/1HZXA_scatter_L-10_minsep6.gnuplot
			pngs/1HZXA_L-10_minsep6_comparemodel_0_0.png
			pngs/1HZXA_L-10_minsep6.pml
			pngs/1HZXA_L-10_minsep6_sum_mem.pml
			pngs/1HZXA_scatter_L-10_minsep6_sum.gnuplot
			pngs/1HZXA_L-10_minsep6_dist.gnuplot
			pngs/1HZXA_L-10_minsep6_sum_dist.gnuplot
			pngs/1HZXA_L-10_minsep6_sum.pml
			pngs/1HZXA_scatter_L-10_minsep6_sum.gnuplot.png

		folding/ pdbs.ls	

ITERATIVE FOLDING OF PROTEINS USING CONTACT RESTRAINT FILES

Iterative folding was tried but did not substantially change results for the set examined. The protocol is listed below as some proteins did benefit slightly.

1. Analyze results from normal folding run by score	n/a	`python2.7 scripts/analyze_folding_run_general_by_score.py folding/ pdbs.ls`	**Input:** pdbs.ls *.pdb files in appropriate filenames.ls file **Output:** scores_L-10_minsep6_sum.ls OUTPUT.FILE
1. Copy over 20 best models by score	n/a	`cat pdbs.ls | awk '{system("scripts/fold_start_models.pl -table "$1"/scores_L-10_minsep6_sum.ls -d pdbs/ -p "$1"/iter1_ -n 20")}'`	**Input:** **Output:**
1. Create .pbs scripts for folding run	n/a	`cat pdbs.ls | awk '{for(i=0;i<20;i++){system("perl folding/create_fold_with_rest_pred_iteration.pl 1.0 1 50 "$1" folding/1L_minsep6_iter1_m"i" 1L_minsep6 8 12 iter1_"i".pdb")}}'`	**Input:** iter1_*.pdb files **Output:**
1. Check and submit .pbs scripts	n/a	Check submit size: `find ?????/*iter1_m*/pbs/ -maxdepth 1 -name "*pbs" -type f -exec echo {} \; | wc -l` Submit pbs files: `find ?????/*iter1_m*/pbs/ -maxdepth 1 -name "*pbs" -type f -exec qsub {} \; > & submit_all_iter1.logt`	**Input:** *.pbs scripts INPUT.FILE **Output:** same as for regular protein submission with different **iter** suffix
1. Analyze iteration output	n/a	`(run from within folding)cat ../pdbs.ls | awk '{ system("ls "$1"/pdbs/"$1"build_L-10_minsep6_iter1_m*_*_final_*.pdb >>`	**Input:** pdbs.ls *_L-10_minsep6_iter1_s

		"$1"/"$1"_L-10_minsep6_iter1_sum_filenames.ls")}' Usage general: python scriptname --directory folding/ --pdblist_filename pdbs.ls --prefix R --suffix tmdiff0-35 --l_frac 1L --minimum_separation 6 Usage specific: python scripts/analyze_folding_run_super_general_by_score.py --sub_directory folding/ --pdblist_filename pdbs.ls --l_frac 1L --suffix iter1 --minimum_separation 6	um_filenames.ls *.pdb files generated for this run **Output:** Same as for normal folding runs (types listed above)

VISUALIZING CONTACT RESTRAINT FILES

Below is a simple script for visualizing contact restraint files on the given PDB file to quickly see accuracy and distribution of errors. This script was used to generate many of the visuals in the manuscript as well as provide confirmation that predictions were successful.

1. Visualize contact restraints on PDB	n/a	Usage general: perl display_contacts_pymol.pl --pdb_file <pdb_file> --contact_file <contact_file> Usage specific (run from within folding/SUBDIR/): From directory folding/1HZXA/ perl ../../scripts/display_contacts_pymol.pl --pdb_file 1HZXA.pdb --contact_file 1HZXA_L-10_minsep12_SCORED.RR Display with: pymol 1HZXA_L-10_minsep12_SCORED.py	Displays output PDB in gray with correct contacts in blue (<=8) close (between 8 and 12) and wrong (further than 12 angstroms) *Input:* 1HZXA.pdb 1HZXA_L-10_minsep12_SCORED.RR **Output:** 1HZXA_L-10_minsep12_SCORED.py

REFERENCES

Ahram, M., Litou, Z. I., Fang, R., & Al-Tawallbeh, G. (2006). Estimation of membrane proteins in the human proteome. *In Silico Biology*, *6*(5), 379–86. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17274767

Alder, B. J., & Wainwright, T. E. (1959). Studies in Molecular Dynamics. I. General Method. *The Journal of Chemical Physics*, *31*(2), 459. doi:10.1063/1.1730376

Alexander, N., Bortolus, M., Al-Mestarihi, A., Mchaourab, H., & Meiler, J. (2008). De novo high-resolution protein structure determination from sparse spin-labeling EPR data. *Structure (London, England : 1993)*, *16*(2), 181–95. doi:10.1016/j.str.2007.11.015

Altschul, S. F., Madden, T. L., Schäffer, a a, Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, *25*(17), 3389–402. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=146917&tool=pmcentrez&rendertype=abstract

Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science (New York, N.Y.)*, *181*(4096), 223–30. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/4124164

Baker, D. (2000). A surprising simplicity to protein folding. *Nature*, *405*(6782), 39–42. doi:10.1038/35011000

Bakheet, T. M., & Doig, A. J. (2009). Properties and identification of human protein drug targets. *Bioinformatics (Oxford, England)*, *25*(4), 451–7. doi:10.1093/bioinformatics/btp002

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., … Zardecki, C. (2002). The Protein Data Bank. *Acta Crystallographica Section D Biological Crystallography*, *58*(6), 899–907. doi:10.1107/S0907444902003451

Bonneau, R., Ruczinski, I., Tsai, J., & Baker, D. (2002). Contact order and ab initio protein structure prediction, 1937–1944. doi:10.1110/ps.3790102.that

Bonneau, R., Strauss, C. E. ., Rohl, C. a, Chivian, D., Bradley, P., Malmström, L., … Baker, D. (2002). De Novo Prediction of Three-dimensional Structures for Major Protein Families. *Journal of Molecular Biology*, *322*(1), 65–78. doi:10.1016/S0022-2836(02)00698-8

Chen, P., & Li, J. (2010). Prediction of protein long-range contacts using an ensemble of genetic algorithm classifiers with sequence profile centers. *BMC Structural Biology*, *10 Suppl 1*(Suppl 1), S2. doi:10.1186/1472-6807-10-S1-S2

Di Lena, P., Nagata, K., & Baldi, P. (2012). Deep Architectures for Protein Contact Map Prediction. *Bioinformatics (Oxford, England)*, 1–9. doi:10.1093/bioinformatics/bts475

Ding, F., Tsao, D., Nie, H., & Dokholyan, N. V. (2008). Ab initio folding of proteins with all-atom discrete molecular dynamics. *Structure (London, England : 1993)*, *16*(7), 1010–8. doi:10.1016/j.str.2008.03.013

Eddy, S. R. (2010). HMMER User ' s Guide, (March), 0–93.

Ezkurdia, I., Graña, O., Izarzugaza, J. M. G., & Tress, M. L. (2009). Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. *Proteins*, *77 Suppl 9*, 196–209. doi:10.1002/prot.22554

Fagerberg, L., Jonasson, K., von Heijne, G., Uhlén, M., & Berglund, L. (2010). Prediction of the human membrane proteome. *Proteomics*, *10*(6), 1141–9. doi:10.1002/pmic.200900258

Fariselli, P., Olmea, O., Valencia, A., & Casadio, R. (2001). Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, *14*(11), 835–843.

Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., … Bateman, A. (2010). The Pfam protein families database. *Nucleic Acids Research*, *38*(Database issue), D211–22. doi:10.1093/nar/gkp985

Floudas, C. a., Fung, H. K., McAllister, S. R., Mönnigmann, M., & Rajgaria, R. (2006). Advances in protein structure prediction and de novo protein design: A review. *Chemical Engineering Science*, *61*(3), 966–988. doi:10.1016/j.ces.2005.04.009

Gloor, G. B., Martin, L. C., Wahl, L. M., & Dunn, S. D. (2005). Mutual Information in Protein Multiple Sequence Alignments Reveals Two Classes of Coevolving Positions †. *Methods*, 7156–7165.

Hopf, T. a, Colwell, L. J., Sheridan, R., Rost, B., Sander, C., & Marks, D. S. (2012a). Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, *149*(7), 1607–21. doi:10.1016/j.cell.2012.04.012

Hopf, T. a, Colwell, L. J., Sheridan, R., Rost, B., Sander, C., & Marks, D. S. (2012b). Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, *149*(7), 1607–21. doi:10.1016/j.cell.2012.04.012

Izarzugaza, M. G., Gran, O., Tress, M. L., Valencia, A., & Clarke, N. D. (2007). Assessment of intramolecular contact. *Cancer Research*, 152–158. doi:10.1002/prot

Jones, D. T., Buchan, D. W. a, Cozzetto, D., & Pontil, M. (2012). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics (Oxford, England)*, *28*(2), 184–90. doi:10.1093/bioinformatics/btr638

Karakaş, M., Woetzel, N., & Meiler, J. (2010). BCL::contact-low confidence fold recognition hits boost protein contact prediction and de novo structure determination. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, *17*(2), 153–68. doi:10.1089/cmb.2009.0030

Karakaş, M., Woetzel, N., Staritzbichler, R., Alexander, N., Weiner, B. E., & Meiler, J. (2012). BCL::Fold--de novo prediction of complex and large protein topologies by assembly of secondary structure elements. *PloS One*, *7*(11), e49240. doi:10.1371/journal.pone.0049240

Kim, D. E., Blum, B., Bradley, P., & Baker, D. (2009). Sampling bottlenecks in de novo protein structure prediction. *Journal of Molecular Biology*, *393*(1), 249–60. doi:10.1016/j.jmb.2009.07.063

Kim, D., Xu, D., Guo, J. -t., Ellrott, K., & Xu, Y. (2003). PROSPECT II: protein structure prediction program for genome-scale applications. *Protein Engineering Design and Selection*, *16*(9), 641–650. doi:10.1093/protein/gzg081

Kopp, J., & Schwede, T. (2004). Automated protein structure homology modeling: a progress report. *Pharmacogenomics*, *5*(4), 405–16. doi:10.1517/14622416.5.4.405

Levinthal, C. (1968). Are there pathways for protein folding. *J. Chim. Phys*, *65*(1), 44. Retrieved from http://csb.stanford.edu/class/public/readings/Molecular_Simulation_I_Lecture4/Levinthal-_J.Chemie_68_Protein_folding.pdf

Lindorff-Larsen, K., Piana, S., Dror, R. O., & Shaw, D. E. (2011). How fast-folding proteins fold. *Science (New York, N.Y.)*, *334*(6055), 517–20. doi:10.1126/science.1208351

Lunt, B., Szurmant, H., Procaccini, A., Hoch, J. A., Hwa, T., & Weigt, M. (2010a). *Inference of Direct Residue Contacts in Two-Component Signaling*. *Methods in Enzymology: Two-Component Signaling Systems, Part C* (1st ed., Vol. 471, pp. 17–41). Elsevier Inc. doi:10.1016/S0076-6879(10)71002-8

Lunt, B., Szurmant, H., Procaccini, A., Hoch, J. A., Hwa, T., & Weigt, M. (2010b). *Inference of direct residue contacts in two-component signaling. Methods in enzymology* (1st ed., Vol. 471, pp. 17–41). Elsevier Inc. doi:10.1016/S0076-6879(10)71002-8

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. a, Pagnani, A., Sander, C., & Zecchina, R. (2011). Protein 3D structure computed from evolutionary sequence variation. *PloS One*, *6*(12), e28766. doi:10.1371/journal.pone.0028766

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., & Sander, C. (2011). Protein 3D Structure Computed from Evolutionary Sequence Variation. *Structure*, *6*(12). doi:10.1371/journal.pone.0028766

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., … Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(49), E1293–301. doi:10.1073/pnas.1111471108

Olmea, O., & Valencia, a. (1997). Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Folding & Design*, *2*(3), S25–32.

Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9218963

Ortiz, a R., Kolinski, a, Rotkiewicz, P., Ilkowski, B., & Skolnick, J. (1999). Ab initio folding of proteins using restraints derived from evolutionary information. *Proteins*, *Suppl 3*(May), 177–85. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/10526366

Ortiz, a R., Kolinski, a, & Skolnick, J. (1998). Nativelike topology assembly of small proteins using predicted restraints in Monte Carlo folding simulations. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(3), 1020–5. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=18658&tool=pmcentrez&rendertype=abstract

Proctor, E. A., Ding, F., & Dokholyan, N. V. (2011). Discrete molecular dynamics. *Science*, *1*(February). doi:10.1002/wcms.4

Przybylski, D., & Rost, B. (2004). Improving fold recognition without folds. *Journal of Molecular Biology*, *341*(1), 255–69. doi:10.1016/j.jmb.2004.05.041

Punta, M., & Rost, B. (2005). Protein folding rates estimated from contact predictions. *Journal of Molecular Biology*, *348*(3), 507–12. doi:10.1016/j.jmb.2005.02.068

Raman, P., Cherezov, V., & Caffrey, M. (2006). The Membrane Protein Data Bank. *Cellular and Molecular Life Sciences : CMLS*, *63*(1), 36–51. doi:10.1007/s00018-005-5350-6

Remmert, M., Biegert, A., Hauser, A., & Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, *9*(2), 173–5. doi:10.1038/nmeth.1818

Rohl, C. a, Strauss, C. E. M., Chivian, D., & Baker, D. (2004). Modeling structurally variable regions in homologous proteins with rosetta. *Proteins*, *55*(3), 656–77. doi:10.1002/prot.10629

Sanders, C. R., & Sönnichsen, F. (2006). Solution NMR of membrane proteins: practice and challenges. *Magnetic Resonance in Chemistry : MRC*, *44 Spec No*, S24–40. doi:10.1002/mrc.1816

Schwede, T., Sali, A., & Eswar, N. (n.d.). Protein Structure Modeling. *World*, 3–35.

Scop Classification Statistics. (2009). Retrieved from http://scop.mrc-lmb.cam.ac.uk/scop/count.html#scop-1.75

Shackelford, G., & Karplus, K. (2007). Contact prediction using mutual information and neural nets. *Proteins: Structure, Function, and …*, *69 Suppl 8*, 159–64. doi:10.1002/prot.21791

Shaw, D., Deneroff, M., & Dror, R. (2008). Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the …*, 91–97. doi:10.1145/1364782

Shindyalov, I. N., Kolchanov, N. A., & Sander, C. (1994). Can three-dimensional contacts in protein

structures be predicted by analysis of correlated mutations? *Protein Engineering*, *7*(3), 349–358. doi:10.1093/protein/7.3.349

Skolnick, J., Kolinski, a, & Ortiz, a R. (1997). MONSSTER: a method for folding globular proteins with a small number of distance restraints. *Journal of Molecular Biology*, *265*(2), 217–41. doi:10.1006/jmbi.1996.0720

Tusnády, G. E., Dosztányi, Z., & Simon, I. (2004). Transmembrane proteins in the Protein Data Bank: identification and classification. *Bioinformatics (Oxford, England)*, *20*(17), 2964–72. doi:10.1093/bioinformatics/bth340

Viklund, H., Bernsel, A., Skwark, M., & Elofsson, A. (2008). SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology. *Bioinformatics (Oxford, England)*, *24*(24), 2928–9. doi:10.1093/bioinformatics/btn550

Weiner, B. E., Woetzel, N., Karakas, M., Alexander, N., & Meiler, J. (2013). BCL::MP-Fold – Folding membrane proteins through assembly of transmembrane helices.

Weiner, B., Woetzel, N., & Karakaş, M. (2013). BCL:: MP-fold: folding membrane proteins through assembly of transmembrane helices. *Structure*, 1–11. doi:10.1016/j.str.2013.04.022

Woetzel, N., Karakaş, M., Staritzbichler, R., Müller, R., Weiner, B. E., & Meiler, J. (2012). BCL::Score--knowledge based energy potentials for ranking protein models represented by idealized secondary structure elements. *PloS One*, *7*(11), e49242. doi:10.1371/journal.pone.0049242

Wu, S., Szilagyi, A., & Zhang, Y. (2011). Improving protein structure prediction using multiple sequence-based contact predictions. *Structure (London, England : 1993)*, *19*(8), 1182–91. doi:10.1016/j.str.2011.05.004

Xue, B., Faraggi, E., & Zhou, Y. (2009). Predicting residue-residue contact maps by a two-layer, integrated neural-network method. *Proteins*, *76*(1), 176–83. doi:10.1002/prot.22329

Yarov-Yarovoy, V., & Schonbrun, J. (2006). Multipass membrane protein structure prediction using Rosetta. *PROTEINS: Structure,*, *62*(4), 1010–1025. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/prot.20817/full

Zwanzig, R., Szabo, A., & Bagchi, B. (1992). Levinthal ' s paradox, *89*(January), 20–22.