Supplementary Materials for

Protein Structure Determination using Metagenome sequence data

Sergey Ovchinnikov, Hahnbeom Park, Neha Varghese, Po-Ssu Huang, Georgios A. Pavlopoulos, David E. Kim, Hetunandan Kamisetty, Nikos C. Kyrpides, and David Baker.

correspondence to: dabaker@u.washington.edu

This PDF file includes:

Materials and Methods

The flowchart of the whole process is shown in the Fig. S4.

Metagenome sequences

The Integrated Microbial genomes (IMG) database (*37*) is a publicly available, comprehensive resource consisting of assembled and annotated metagenomes, the genomic information of which has been integrated successfully with isolate genomes from all three domains of life. The metagenomes present in IMG are a combination of Joint Genome Institute (JGI)-sequenced projects and data deposited by users. An initial dataset of over 2 billion proteins, predicted on assembled contigs, from ~5000 metagenomes in IMG served as the initial dataset. Both partial and full length proteins were included in this set in order to maximize the search space.

The *hmmsearch* tool from the HMMER package version 3.1b1 (*41*) was used to search the dataset of proteins described above, with each PFAM HMM (*1*) as the query, using the trusted cutoffs for each HMM. The results were then filtered to only retain those protein hits, that covered at least 75% of the PFAM query. The protein sequence of the retained gene hits were then extracted from the IMG database and used in the further analysis. These we will refer to as "metagenomic PFAM sequences" in remainder of the methods section.

Nf (effective number of sequences) calculation

The Hamming distance is computed between all of the sequences in a given MSA (multiple sequence alignment). Each sequence is given a weight of 1/(number of sequences > 80% identity). These weights are used within GREMLIN (*4*, *12*) to downweight redundant sequences, and the sum of these weights divided by square-root of length of the MSA is used for the Nf calculation. 80% identity threshold was chosen because it results in both the best accuracy for predicted contacts and the best correlation to accuracy for the Nf calculation (Fig. S5).

To test the effects of re-weighting, the dataset from (ref *21*; Supplementary file 3) was used. The alignments were filtered to remove sequences that do not cover at least 75% of the query sequences. Positions that had more than 50% gaps from previous filtering (HHfilter version 2.0.15; -id 90 -cov 75)(*2*) were removed. For this test, only identical sequences were removed. Alignments with less than 100 unique sequences were excluded.

Contact prediction

In some cases, the domain boundaries defined in PFAM are not structurally realistic, this is evident when there are strong contacts that stretch beyond the PFAM boundaries, or when the N and C-terminal regions are split into two domains, yet there are extensive contacts between the two. In both cases, modeling a representative sequence alleviates this issue. For each PFAM with at least 64 Nf, *hmmsearch* (from HMMER version 3.1b1; default parameters) (*41*) was used to identify a representative sequence from either the SWISS-PROT or reference genome database.

Starting with this sequence, HHblits (from HHsuite version 2.0.15; -n 8 -e 1E-20 -maxfilt ∞ -neffmax 20 -nodiff -realign_max ∞) (*42*) was run against the clustered UniProt database from 2015_06 to generate an initial alignment. After HHfilter (-id 90 -cov 75) this initial alignment was used to construct a hidden Markov model (HMM) using *hmmbuild* (from HMMER version 3.1b1) (*41*). A conservative bit-score value of 27 (this is typically the cutoff used for PFAM definitions) was used to search against a master database containing both the UniRef100 and metagenomic PFAM sequences. Bit-score instead of E-value was used, because it is independent of database size. The UniRef100 contains all the UniProtKB records plus the UniPrac records not covered by the UniProtKB (*35*). The output was again filtered using HHfilter (-id 90 -cov 75). If more than 2 fold fewer sequences were recovered compared to the PFAM alignment, the representative sequence was trimmed to the PFAM boundaries and the alignment was recreated (for the 81 protein benchmark described below, such cases were discarded and hence there was no trimming to PFAM boundaries). Following the filtering, positions that have more than 50% gaps are removed. GREMLIN (v2.01) with default parameters was used for contact prediction (*12*). CCMPRED (v0.1), a parallel implementation of GREMLIN (*43*), was used for the subsampled alignments and reweighting test. For CCMPRED, the default maximum number of iterations was modified to 100 to ensure convergence.

Contact map alignment

For the purposes of identifying structural homologs and characterizing protein families, we developed a contact map alignment method called *map_align*. *map_align* uses an iterative double dynamic programming algorithm similar to the one first described by Taylor (*33*) for protein structure comparison, with some modifications for our purposes to produce an alignment that optimizes structural overlap.

The *map_align* algorithm comprises two dynamic programming steps. In the first step, a score is computed for each row (corresponding to a specific residue) of the first contact map with each row for the second contact map. This score is the sum of Gaussian functions: $\exp(-x^2/(2y^2))$, where x is the difference in sequence separation of aligned contacts and y (or standard deviation) is a function of the smaller of the two sequence separations, as described below. Dynamic programming is used to find the alignment of the contacts for the two rows being matched which maximizes the sum of these Gaussian functions. These optimized sums of scores are then entered in a second matrix, and the optimal contact alignment is then found by dynamic programming, using the Smith–Waterman algorithm (*44*). At this point, however, the scores for individual row-row comparisons are overestimates since in the first step the alignments for each pair are independent. We then update the second step similarity matrix based on the current alignment, and carry out the second step dynamic programming again (although the score is quite different, this updating strategy is similar to that used in Taylor's method (*33*) for comparing two structures based on distance matrices). This process is repeated 20 times (by

20th iteration, the alignment convergences and no more contacts are made). The initial estimate of the similarity matrix is critical in getting at least part of the alignment correct as this serves as a nucleation point for aligning the rest of the contacts. To maximize the chance of success, we try a number of variations in the first step. The standard deviation in the Gaussian function is made either a constant, linear and quadratic function of the lower of the two sequence separations, and a range of scaling parameters are tested. For each choice of functional form and scaling parameter, we carry out the full iterative optimization described above, and choose that alignment which best matches the two contact maps (maximizes the number of aligned contacts) assigning lower weight to low sequence separation contacts (weight of 0.50 for sequence separation ≤ 4, 0.75 for 5 and 1.00 for ≥ 6; the weight is based on the lower of the two sequence separations). The pseudocode of the aligorithm is provided at the end of this section.

Fig. S7 compares our approach against A_purva (*45*), GR-align (*46*), Al-Eigen (*47*) and MSVNS (*48*). A_purva is an "exact" branch and bound algorithm which if run long enough will find the maximum overlap of contacts. The problem is that it can take a long time to run for a protein pair, which is not practical for database search, it is useful to obtain the best possible alignment, once ranking has been achieved with approximate methods. We use the results of A_purva to establish the baseline for the Skolnick dataset of 40 SCOP domains(*45*). GR-align and AI-Eigen use a Needleman-Wunsch algorithm (*49*) where the cost for matching two residues is based on “graphlet” degree similarity and weighted eigenvalues respectively. MSVNS is a stochastic local search method that aims to find good solutions by adding and removing pairs of residues using different strategies. *Map_align* is able to find better contact map alignments on average because of the iterative updating of the similarity matrix.

The MRFalign (*50*) program elegantly compares two sequence families evaluating whether the co-evolution patterns are similar, and achieves very sensitive remote homologue detection. For our model building purposes, we are searching for structural fragments which fit contacts independent of evolutionary relatedness, and hence direct comparison of co-evolution derived predicted contact map to contact maps from known structure is most useful.

The source code for the algorithm can be downloaded from GitHub: http://github.com/sokrypton/map_align. The Pseudocode for map_align algorithm for aligning two input contact maps (map_a and map_b) is provided below.

```
for sep_x (0,1,2)
	for sep_y (1,2,4,8,16,32)
		ini_mtx = initialize_matrix(sep_x,sep_y)
		for gap_e (-0.2,-0.1,-0.01,-0.001)
			mtx = ini_mtx
			alignment = get_alignment(mtx,-1,gap_e)
			score = SWalign(mtx,-1,-0.01)/2
			if (score > best_score)
				best_alignment = alignment
				best_score = score
print(best_alignment)

function initialize_matrix(sep_x,sep_y)
	for ai (columns in map_a)
		for bi (columns in map_b)
			for aj (values in column ai)
				for bj (values in column bi)
					sa = ai-aj
					sb = bi-bj
					if (sa>0 and sb>0 or sa<0 and sb<0)
						s_dif = ||sa|-|sb||
						s_min = min(|sa|,|sb|)
						s_std = sep_y*(1+pow(s_min-2,sep_x))
						w = sep_weight(s_min)*gaussian(0,s_std,s_dif)
						M[aj][bj] = map_a[ai][aj] * map_b[bi][bj] * w
					else
						M[aj][bj] = -1
			mtx[ai][bi] = SWalign(M,0,0)
	return mtx

function get_alignment(mtx,gap_open,gap_extention)
	for i (0..20) //iterate
		alignment = SWalign(mtx,gap_open,gap_extention)
		for ai (columns in map_a)
			for bi (columns in map_b)
				sco = 0
				for aj (values in column ai)
					bj = alignment[aj]
					sa = ai-aj
					sb = bi-bj
					if (sa>0 and sb>0 or sa<0 and sb<0)
						w = sep_weight(min(|sa|,|sb|))
						sco += map_a[ai][aj] * map_b[bi][bj] * w
				mtx[ai][bi] = i/(i+1) * mtx[ai][bi] + sco/(i+1)
	return alignment

function sep_weight(sequence_seperation)
	if (sequence_seperation <= 4) return 0.50
	else if (sequence_seperation == 5) return 0.75
	else return 1.00

function SWalign(matrix,gap_open,gap_extention)
	return local alignment
```

The algorithm tries different *sep* (sequence separation difference) and gap extension (gap_e) penalties and reports the best alignment. For sep_x we try a constant, linear and quadratic function with different scaling factors (sep_y). The alignment that maximizes the number of contacts (while minimizing the number of gaps) is reported at the end.

Structure prediction

Each contact map was examined and trimmed at the N and C terminus to remove regions not constrained by strong non-local contacts. If the contact map was larger than 300 residues, it was split into multiple overlapping domains of 300 residues or less. 66 out of 921 protein families modeled were parsed into 2 or more overlapping domains (20 of the 612 converged models are from these parsed domains). 10,000 *de novo* models were generated using the standard Rosetta AbInitio protocol using sigmoid restraints as described before (*21*). An additional 10,000 models were generated also including bounded restraints (*51*) which improve convergence in many cases because large restraint violations are given large penalties. The bounded restraints were only used during the coarse-grain sampling and disabled during the full atom refinement and minimization. See end of this section for flags and parameters used for the AbInitio protocol.

In addition to the *de novo* structure prediction calculations which start from an extended chain, models were generated by recombining portions of structures identified by the *map_align* contact map matching protocol described above. The search was performed against a subset of PDB(s) with a maximum mutual sequence identity of 30% (*52*). The predicted contacts were aligned against PDB contact maps (see section above). The PDB contact maps were defined using a 5 Å distance cutoff between any pair of heavy atoms for residues with sequence separation of 3 or higher. The top 20 hits were input into the Rosetta hybrid protocol (*32*): the input models are first superimposed and then split into secondary structure elements which are recombined. In addition to recombination, fragment insertion is allowed at all positions, allowing sampling of structures not seen in any of the input templates. 4,000 models are produced at this stage. See end of this section for flags and parameters used for the Rosetta hybrid protocol.

In all structure prediction calculations, structures were ranked based on their Rosetta energies and their fit to the contact restraints. A simple linear combination of the two metrics was used with a scale factor found previously (*21*) to give the two roughly equivalent dynamic range.

30 cluster centers of the whole structural pool, containing 500 top-scoring *de novo* structure prediction models and 100 top-scoring *map_align* models, are selected as initial structures for further refinement using iterative hybridization. For clustering, starting from the lowest energy conformation the next lowest energy conformation is added to the pool if it is not close to any of pool structures added (TMscore < 0.4), and this is stopped when the pool size becomes 30. If the pool size is smaller than 30 after looking at all conformations, we repeat this step again starting

from the lowest unadded conformation with more generous TMscore cut until the pool size reaches to 30.

At each iteration of refinement, Rosetta hybridization protocol is applied 60 times on different combinations of 5 randomly selected models from the structural pool. Once 60 new models are generated, structural pool is updated for the next iteration; new structures are scanned in ascending order of their combined score of Rosetta energy and coevolution restraint scores, and the structure "A" in the original pool is replaced by this new structure "B" if i) it has higher (= unfavorable) energy *and* ii) "A" and "B" are structurally similar or "B" is structurally different from any of the original structures (*53*). Structural similarity criteria linearly changes from 0.4 to 0.7 TMscore from first to 18th iteration, ands keeps unchanged until the end. Therefore size of structural pool (30 structures) is maintained throughout the refinement stage. This is repeated for 30 iterations generating 1,800 structures. For final model selection, structural averaging (*54*) is performed on this entire structural pool, followed by model relaxation to idealize local geometry of the model.

Total computational time for structure calculation of 200-residue protein takes approximately 13,000 core hours: generating 20,000 *de novo* models, 4,000 map_align models, and running structural refinement take 10,000, 2,000, and 1,000 core hours, respectively. This process can be highly parallelized; each of *de novo* and *map_align* models are modeled in parallel using Rosetta@home, and refinement is carried out using 64 cores in parallel.

AbInitio protocol parameters and flags:

```
AbinitioRelax # name of rosetta app
-abinitio::increase_cycles 10
-abinitio::fastrelax
-abinitio::rg_reweight 0.5
-abinitio::rsd_wt_helix 0.5
-abinitio::rsd_wt_loop 0.5
-constraints:cst_weight 3
-constraints:cst_file SIG_BND_cst # sigmoid + bounded constraints
-constraints:cst_fa_weight 3
-constraints:cst_fa_file SIG_cst # only sigmoid constraints for full atom refinement mode
-in::file::fasta t000_.fasta
-frag3 t000_.200.3mers.gz
-fragA t000_.200.9mers.gz
-fragB t000_.200.3mers.gz
-nstruct 10000
```

Rosetta hybrid protocol parameters and flags:

```
rosetta_scripts # name of rosetta app
-frag_weight_aligned 0.1
-beta # this flag enables the latest rosetta score function
-in:file:fasta t000_.fasta
-parser:protocol hyb.xml # rosetta script (see below)
-relax:minimize_bond_angles
-relax:jump_move true
-relax::dualspace
-default_max_cycles 200
-relax:min_type lbfgs_armijo_nonmonotone
-hybridize:stage1_probability 1.0
-hybridize:stage1_4_cycles 400
-nstruct 4000

<ROSETTASCRIPTS>
       <TASKOPERATIONS></TASKOPERATIONS>
       <SCOREFXNS>
              <stage1 weights="stage1.wts" symmetric=0>
                     <Reweight scoretype=atom_pair_constraint weight=3/>
              </stage1>
              <stage2 weights="stage2.wts" symmetric=0>
                     <Reweight scoretype=atom_pair_constraint weight=3/>
              </stage2>
              <fullatom weights="beta_cart.wts" symmetric=0>
                     <Reweight scoretype=atom_pair_constraint weight=3/>
              </fullatom>
       </SCOREFXNS>
       <FILTERS></FILTERS>
       <MOVERS>
              <Hybridize name=hybridize stage1_scorefxn=stage1 stage2_scorefxn=stage2
              fa_cst_file=SIG_cst fa_scorefxn=fullatom batch=1 stage1_increase_cycles=2.0
              stage2_increase_cycles=1.0 linmin_only=0 skip_long_min=1>
                     <Fragments 3mers="t000_.200.3mers.gz" 9mers="t000_.200.9mers.gz"/>
                     <Template pdb="X.pdb" weight="X" cst_file="SIG_BND_cst"/>
                     <Template pdb="Y.pdb" weight="Y" cst_file="SIG_BND_cst"/>
                     <Template pdb="Z.pdb" weight="Z" cst_file="SIG_BND_cst"/>
              </Hybridize>
       </MOVERS>
       <APPLY_TO_POSE></APPLY_TO_POSE>
       <PROTOCOLS>
              <Add mover=hybridize/>
       </PROTOCOLS>
       <OUTPUT scorefxn=fullatom/>
</ROSETTASCRIPTS>
```

Benchmark with UniProt only for sweeping over Nf values

The dataset was chosen using PDB20 database from PISCES (resolution limited to 3.0 or better, length >= 50, date 03Mar2015). Multiple sequence alignments were generated for each using HHblits (version 2.0.15; -n 8 -e 1E-20 -maxfilt ∞ -neffmax 20 -nodiff -realign_max ∞), and HHfilter (-id 90 -cov 75) using a clustered UniProt database from 2015_06. No metagenomic sequences were used for this benchmark. Alignments with at least 64 Nf and no homologs in the PDB prior to 2011 were selected. Homology detection was carried out using *blastpgp* (version 2.2.26; -t 1 -e 0.05) (*55*) against a database of PDB sequences from 2011, using a checkpoint file generated by *csblast* (*56*) from the multiple sequence alignment. The benchmark was further filter by removing targets that had missing internal density, reducing the size from 37 to 25

targets. In addition to these proteins, two targets from CASP11 were included: T0806 and T0824. These were the only targets with enough sequences during the CASP11 experiment, and were highlights of the BAKER group human efforts in CASP11; we wanted to test how robust our protocol was in automating their accurate prediction. See Table S1 for a list of these targets and ranges modeled. To avoid bias in the modeling, structures deposited in PDB before 2011 were used for both fragment picking and template selection using *map_align*.

For the subsampled MSAs (multiple sequences alignments), sequences were shuffled and added one at a time until the desired Nf was achieved. Since we are subsampling from a large MSAs, these are likely to result in MSAs that are more diverse on average than a natural MSA of the same Nf. To compensate for higher diversity and still have the same Nf, the subsampled MSAs will need to contain lower number of sequences than natural MSAs. To test if these compositional differences of the natural and subsampled MSAs affect accuracy, we binned our PDB30 set from the "Nf (effective number of sequences) calculation" section (see above) into different Nf categories and compared the distribution of accuracies of the predicted contacts (Fig. S6). We find the accuracies to be on average very similar (Fig. S6A), even though there does indeed exists a compositional differences (Fig. S6B-C).

Additional benchmark with metagenomic sequences for testing the modeling protocol

For the additional benchmark set, a more stringent threshold for selecting protein families with no homology in the PDB, no restriction to source and quality of experimental density, and combination of UniRef100 and metagenomic sequences were used. Each PFAM hmm (provided by HHsuite) was used to search against two different PDB HMM databases; one from 01Jan2012 and one from 13Aug16. HHsearch (default options with -glob flag) was used for the HMM-HMM alignment. For 496 PFAMs with no hits in 2012 (E-value of the top PDB hit > 1) and a strong hit in 2016 (E-value of the top hit < 1E-10), *hmmsearch* (default options with -T 27 flag) was used to collect sequences from the master database containing both the UniRef100 and metagenomic PFAM sequences. For each of 126 PFAMs with more than 64 Nf, the untrimmed UniProt sequence corresponding to the top PDB hit from 13Aug16 was used as the representative sequence. Of the 126, only 70 pfams had that at least 64 Nf for the full length UniProt sequence and the remainder were discarded (there was no trimming to PFAM boundaries to avoid incorporating any native structural bias in PFAM). For three cases where the UniProt sequences were over 400 residues in length (Q96MU8, A9JTH8 and P24043), the "Family & Domains" information from UniProt was used to trim the sequences to domain boundaries. To confirm that these boundary definitions were not influenced by structure, we used hmmscan (from the hmmer package) to scan against the 2012 hmm boundaries (Pfam 26, before structures for these families was released) and the 2016 hmm boundaries (Pfam 30). The boundaries were identical. Contact prediction and model generation were carried out exactly as described above for the families with unknown structures, unbiased by knowledge of the correct

structure. In all, 86 domains were modeled (11 of the targets were split based on the contact map into multiple overlapping domains). Five of the domains were excluded from analysis, as they contained no density in the target PDB. For a list of the remaining targets see Table S5.

Convergence Criteria

As shown in Fig. S2, there is a strong correlation between model accuracy and the extent of convergence of the structure prediction calculations. Based on the results shown in this figure, we used as a measure of convergence *max(con_DN, con_MP, iDN_iMP)*, in which DN and MP refer to *de novo* and *map_align*, respectively, and *iDN_iMP* is the consistency between independent runs of the iterative hybrid protocol on the DN and MP models. Criteria of 0.65 is used for the selection of 614 among 1024 families. The first two measures are the average pairwise TMscore (*23*) between the top-10 models produced using each method (Fig. S2 A and B) while the third measure is the TMscore between the top scoring model of each method (Fig. S2 C). This convergence criteria is based on the result from Fig. S2 that final models -- derived by iterative hybridization of the DN and MP models -- are likely to be accurate when a) either the DN or MP method converges on its own or b) the models produced by the two independent methods are similar to each other.

SCOPe classification and new fold detection

A filtered subset of SCOPe domains (version 2.06) (*40*) was downloaded from Astral with sequence identity of 40%. TMalign was used to superimpose each model to each SCOPe domain and compute fold similarity. If the the best hit (best TMalign score scaled by length of model) is greater than 0.5 TMalign score, it is used to classify the model, otherwise the model is labelled as new fold.

For the 137 models with no hits in SCOPe, an all vs. all analysis was performed. The TMscore >= 0.5 scaled by the average of two length was used for clustering. 129 connected components were found. This includes one cluster with 3 members and six clusters with 2 members.

Evaluation of recently solved structures

The solved structures are often close homologs of the predicted structures (rather than the identical protein), making the TMscore calculation (that requires identical sequence) difficult. For evaluation, we instead used TMalign, which is a sequence independent structure comparison method. Though, before running TMalign, HHsearch (from HHsuite package; version 2.0.15) is used to identify regions that are common to both predicted model and PDB homolog. Regions that are not aligned are removed from both model and PDB homolog. These trimmed structures are used in Fig. 1 and Fig. S3, and used in TMalign score calculation.

Supplementary Text

Limitation of modeling

While all of our benchmark tests and comparisons of recently determined crystal structures to previously generated models suggest that the structures presented in this paper are likely to be quite accurate over their entire length (TMscore > 0.7), there are several systematic limitations users of these models should be aware of. For membrane proteins the lipid bilayer is not modeled explicitly, and intra and extracellular domains can dip into the membrane region (for example Fig. S8A). Ligands and co-factors are also not modeled explicitly and models can collapse to obscure binding sites (Fig. S8A-C; if the co-factors are known they can be easily incorporated during modeling). Finally short segments can have the incorrect local secondary structure in the context of an overall correct topology (Fig. 1E). A potential additional source of error is confounding intra- and inter-domain contacts in homo-oligomers, but our benchmark set calculations suggest that the convergence based selection criterion eliminates almost all such cases except for intertwined homo-oligomers where it is possible to make all the predicted contacts within the monomer (an example of such case is 5AN6 (Fig. S2F)).

Models for groups of functionally related proteins

As noted in the main text, it is a challenge to fully present the large numbers of new structures described in this paper. A number of the structures fall into groups with related functions or classifications. These include four cobalamin biosynthesis (Fig. S9), two citrate lyase (Fig. S10), ten sporulation (Fig. S11), and eight immunity proteins (Fig. S12).

Since we prepared our Report, additional work has been published in which coevolution contacts were used to make models (*58*-*63*).

References

41. S. R. Eddy, A new generation of homology search tools based on probabilistic inference. *Genome Inform* **23**, 205-211 (2009).
42. M. Remmert, A. Biegert, A. Hauser, J. Söding, HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods* **9**, 173-175 (2012).
43. S. Seemayer, M. Gruber, J. Söding, CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*. **30**, 3128–3130 (2014).
44. T. F. Smith, M. S. Waterman, Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195–197 (1981).
45. N. Malod-Dognin,, Nicola Yanev, and Rumen Andonov. "Comparing protein 3d structures using a_purva." PhD diss., INRIA, 2010.
46. N. Malod-Dognin, N. Pržulj, GR-Align: fast and flexible alignment of protein 3D structures using graphlet degree similarity. *Bioinformatics*. 30, 1259–1265 (2014).
47. P. Di Lena, P. Fariselli, L. Margara, M. Vassura, R. Casadio, Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*. 26, 2250–2258 (2010).
48. D. A. Pelta, J. R. González, M. Moreno Vega, A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*. 9, 161 (2008).
49. S. B. Needleman, C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443–453 (1970).
50. J. Ma, S. Wang, Z. Wang, J. Xu, MRFalign: protein homology detection through alignment of Markov random fields. *PLoS Comput. Biol.* **10**, e1003500 (2014).
51. D. E. Kim, F. Dimaio, R. Yu-Ruei Wang, Y. Song, D. Baker, One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. *Proteins*. **82 Suppl 2**, 208–218 (2014).
52. G. Wang and R. L. Dunbrack, Jr. PISCES: a protein sequence culling server. Bioinformatics, 19,1589-1591 (2003).
53. J. Lee, H.A. Scheraga, and S. Rackovsky, New optimization method for conformational energy calculations on polypeptides: Conformational space annealing, *J Comput. Chem.* 18, no. 9, 1222-1232 (1997).
54. H. Park, F. DiMaio, D. Baker, The origin of consistent protein structure refinement from structural averaging, *Structure* **23**, 1123-1128 (2015).
55. A. A. Schäffer *et al.*, Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.* **29**, 2994–3005 (2001).
56. Angermüller, A. Biegert, J. Söding, Discriminative modelling of context-specific amino acid substitution probabilities. *Bioinformatics*. **28**, 3240–3247 (2012).
57. Team, R. Core. R: A language and environment for statistical computing. (2013).

58. M. M. Kassem, Y. Wang, W. Boomsma, K. Lindorff-Larsen, Structure of the Bacterial Cytoskeleton Protein Bactofilin by NMR Chemical Shifts and Sequence Variation. *Biophys J* **110**, 2342-2348 (2016).
59. A. Kedrov *et al.*, Structural Dynamics of the YidC:Ribosome Complex during Membrane Protein Biogenesis. *Cell Rep* **17**, 2943-2954 (2016).
60. D. Lloyd Evans, S. V. Joshi, Elucidating modes of activation and herbicide resistance by sequence assembly and molecular modelling of the Acetolactate synthase complex in sugarcane. *J Theor Biol* **407**, 184-197 (2016).
61. D. G. Schep, J. Zhao, J. L. Rubinstein, Models for the a subunits of the Thermus thermophilus V/A-ATPase and Saccharomyces cerevisiae V-ATPase enzymes by cryo-EM and evolutionary covariance. *Proc Natl Acad Sci U S A* **113**, 3245-3250 (2016).
62. M. J. Skwark, M. Michel, D. M. Hurtado, M. Ekeberg, A. Elofsson, Accurate contact predictions for thousands of protein families using PconsC3. *bioRxiv*, 079673 (2016).
63. W. R. Taylor, T. R. Matthews-Palmer, M. Beeby, Molecular Models for the Core Components of the Flagellar Type-III Secretion Complex. *PLoS One***11**, e0164047 (2016).

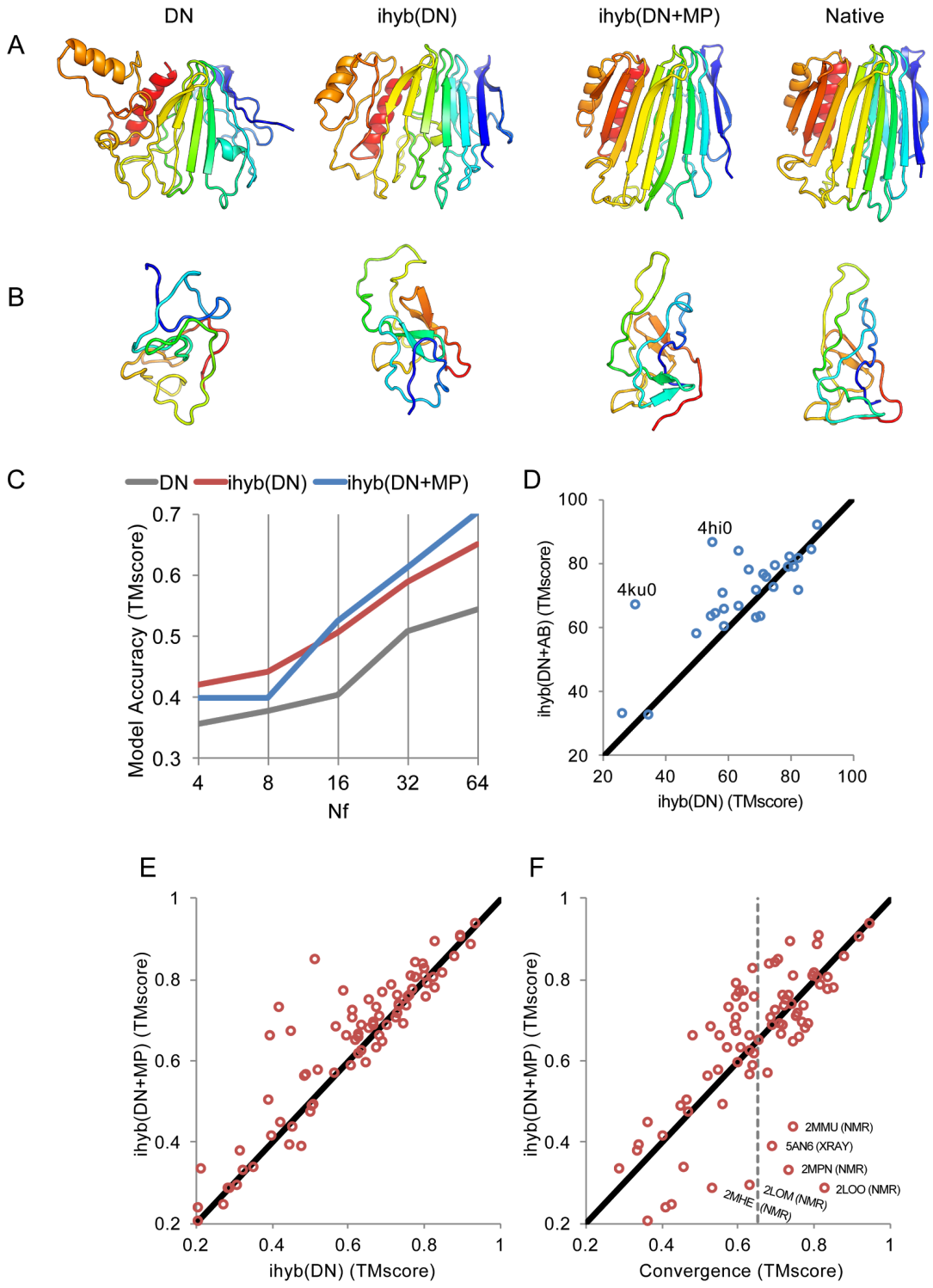

Fig. S1. Adding partial threads detected by *map_align* (MP) improves accuracy of the final model more frequently than reducing it for Nf > 16. A) Models for 4hi0 at different stages of the protocol. B) Models for 4ku0 at different stages of the protocol. C) Average model accuracy with and without *map_align* partial threads for different Nf bins over 27 protein benchmark. D)

Model accuracy with and without *map_align* partial threads for each protein in the 27 protein benchmark at Nf=64. For 5 targets, the TMscore increases by more than 10 with partial threads, and for only 1 target does it decrease by more than 10 with partial threads. Abbreviations in the figure: DN, *de novo*; ihyb(DN), iterative hybridization of *de novo* models; ihyb(DN+MP), iterative hybridization of *de novo* models with *map_align* partial threads. E) Same comparison as in D for 81 protein benchmark with Nf > 64. For 9 targets, the TMscore increases by more than 10 with partial threads, and for no targets, does it decrease by more than 10 with partial threads. To test the significance of these improvements (C-E) we performed Wilcoxon Signed-Rank Test using R (57), for more details see SI Table S6. F) Correlation between convergence criteria and model quality. With the exception of small transmembrane proteins (~100 length) solved by NMR and an 5AN6 (intertwined dimer), the convergence cutoff of 0.65 (grey dotted line) is a good predictor of accuracy (average TMscore ~ 0.7). For details of the outliers see Supplementary Fig. S13.

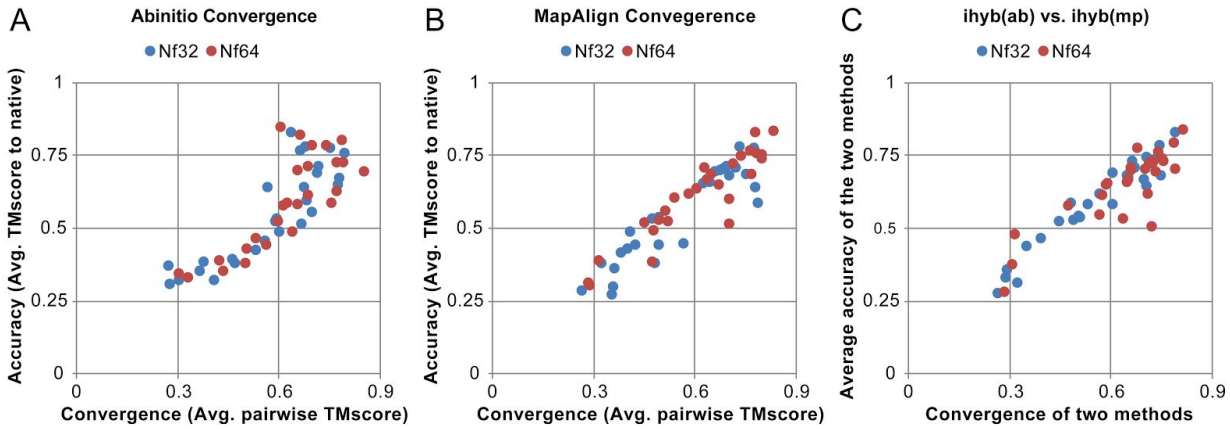

Fig. S2. Correlation between convergence (as measured by average pairwise TMscore) and model accuracy over benchmark set. The extent of convergence correlates with model accuracy for (A) the top ranked (using the linear combination of Rosetta energy and contact score described in the Methods) ten out of 10,000 *de novo* models and (B) the top ranked 10 of 4,000 *map_align* models after a single round of the Rosetta hybridization protocol. C) Following multiple rounds of iterative hybrid refinement of the *de novo* models and the map-align models independently, the similarity between the top model from each protocol also correlates with model accuracy. The convergence criterion we use in this paper is that the maximum of the three metrics is greater than 0.65; this corresponds to an average TMscore to native of 0.7 over the benchmark set.

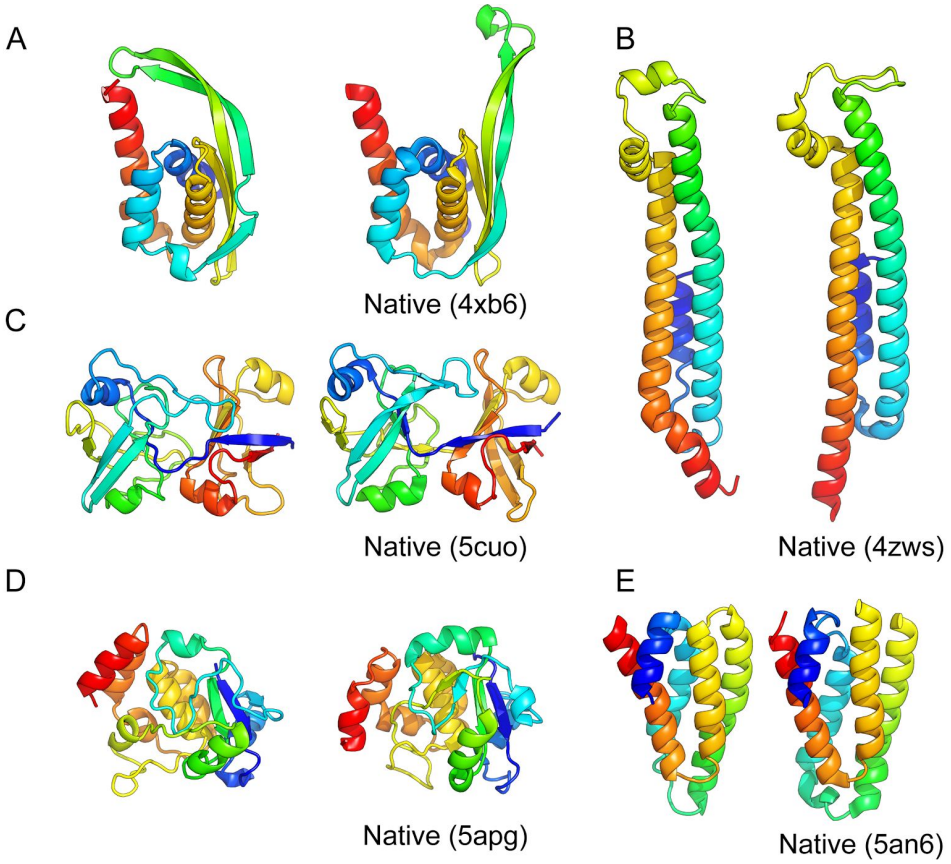

Fig. S3. Structures solved while the manuscript was in preparation. For TMalign score and convergence criterion values see Table S4.

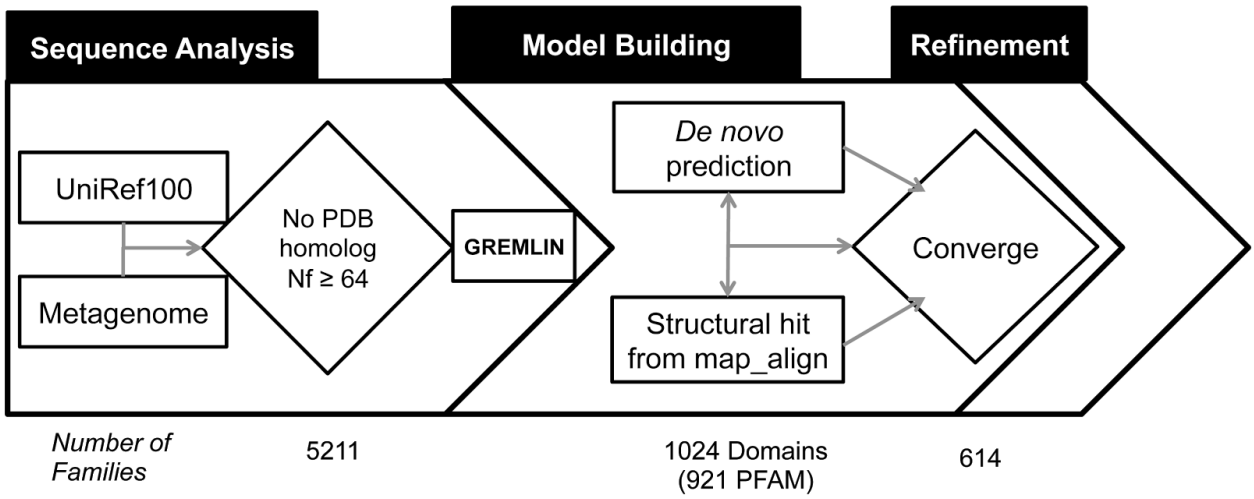

Fig. S4. Flowchart of method. Details of the steps are explained in Supplementary text.

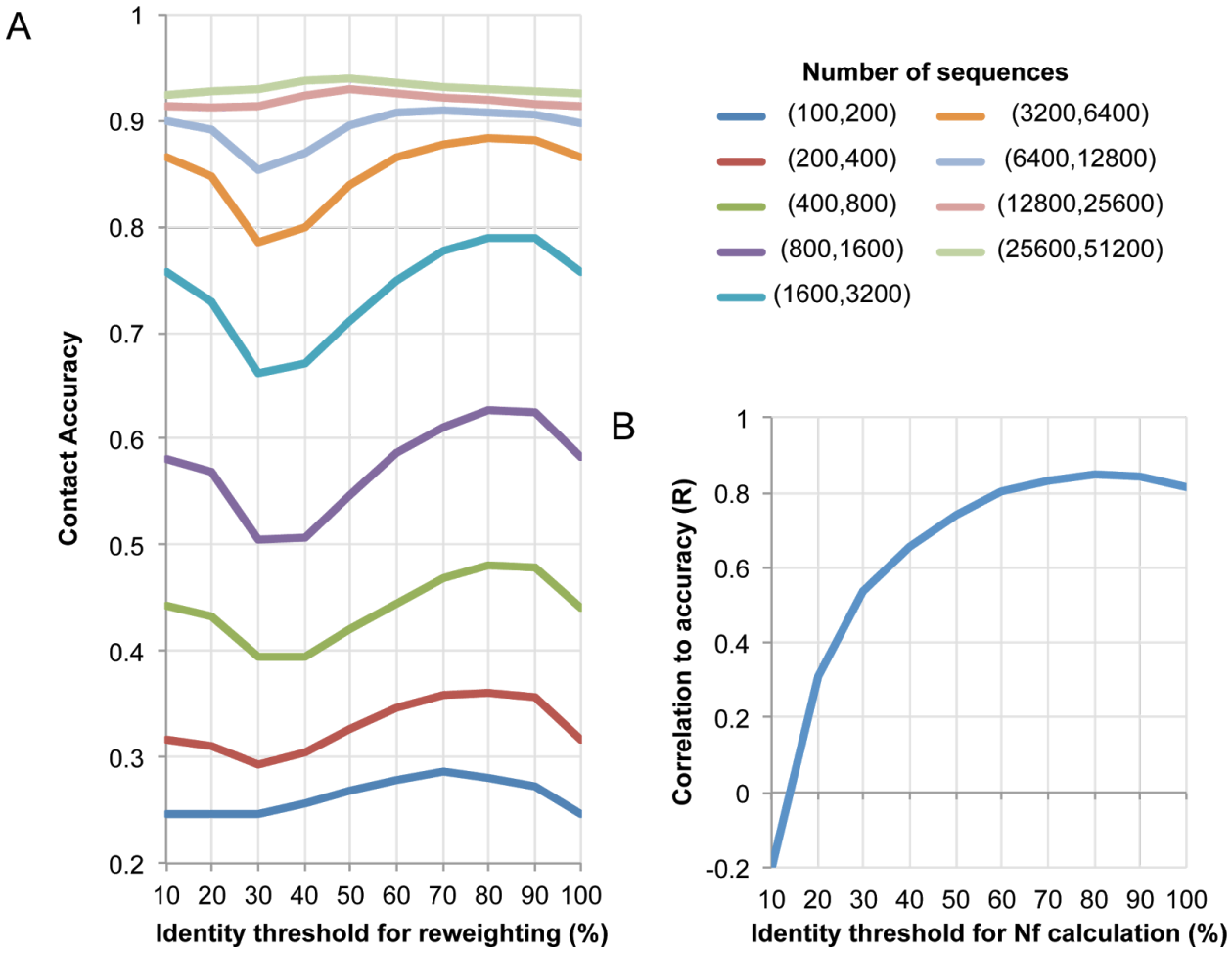

Fig. S5. Accounting for sequence redundancy and diversity in the MSAs (multiple sequence alignments) results in both higher accuracy and better prediction of accuracy. For the PDB30 set (see methods), each sequence in MSA is given a weight of 1/(number sequences > X% identity), where X is search parameter to be determined. These weights are used in GREMLIN to downweight redundant sequences. A) For a broad range of protein families sizes (different lines) the best accuracy is achieved at X=80% sequence identity cutoff. The weights computed with identity threshold of 10% or 100% result in uniform weights for each sequence and hence result in identical accuracies. The accuracy is computed using the top 0.5L (L is the length of the sequence) contacts with sequence separation greater than or equal to 6. A contact is considered made when the smallest distance between any two heavy atoms is less than 8 Å. B) The sum of weights divided by square-root of length is used to calculate the Nf value. Here we try different identity thresholds for the Nf calculations and compute the correlation to accuracy computed with the default identity threshold (80%).

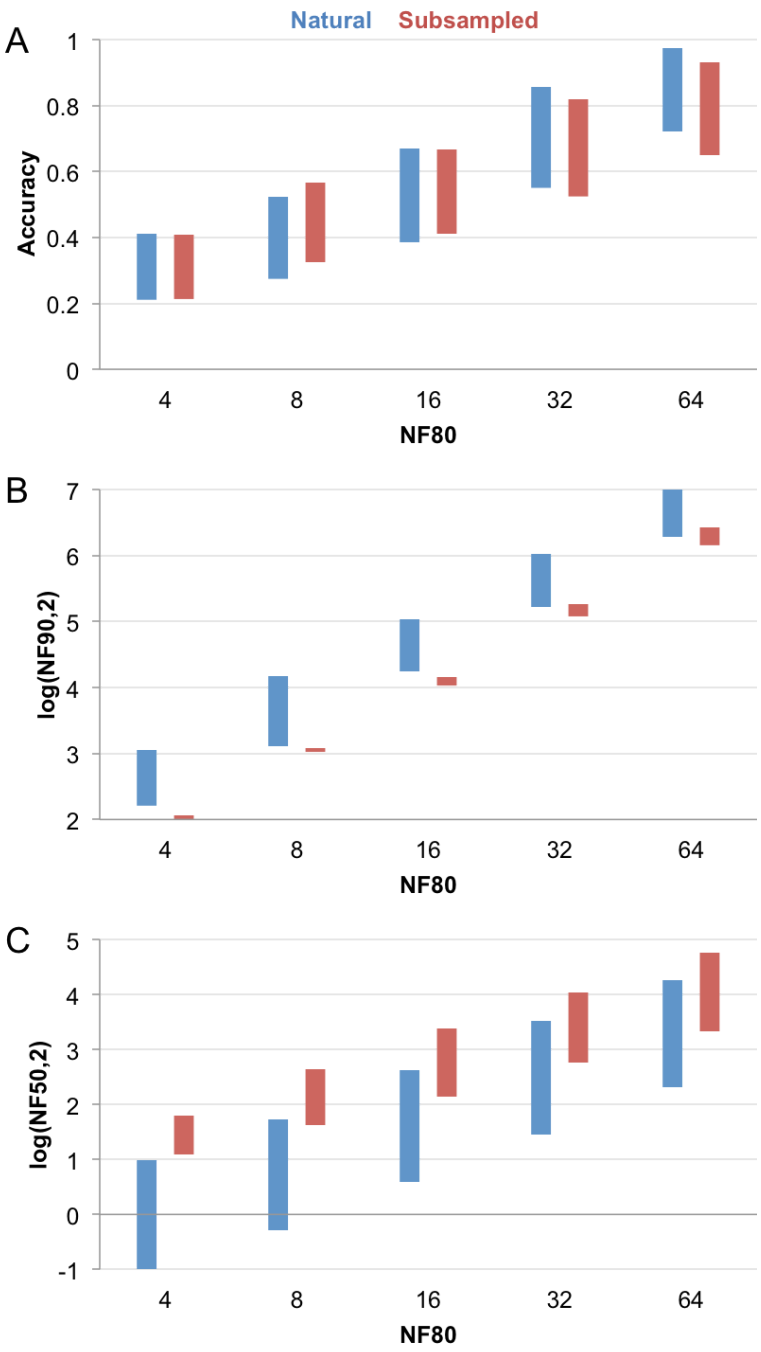

Fig. S6. The distribution of accuracies and diversities of natural and subsampled MSA (multiple sequence alignments). The natural set comes from the PDB30 set (see Supplementary text) where the MSAs are binned based on the Nf calculation at 80% sequence identity threshold. The subsampled set comes from the 27 benchmark set. A) The contact accuracy of both the natural and subsampled MSAs is the same. B) There are typically less sequences in the subsampled MSAs, but this is compensated by C) higher diversity. The top and bottom of each bars indicates +/- standard deviation from the mean.

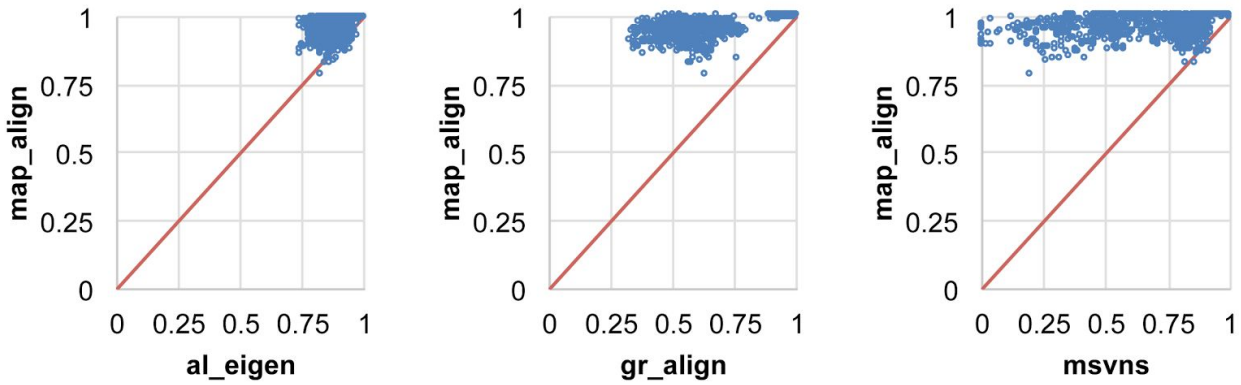

Fig. S7. Comparison of *map_align* to previously published methods for detecting structures satisfying a given set of residue-residue contacts. For the Skolnick dataset of 780 SCOP protein pairs brought from Malod-Dognin *et al* (45), *map_align* recovers a larger fraction of contacts than the other methods. The x and y-axes are the fraction of max contacts recovered, as computed by Malod-Dognin *et al* by the exact method a_purva.

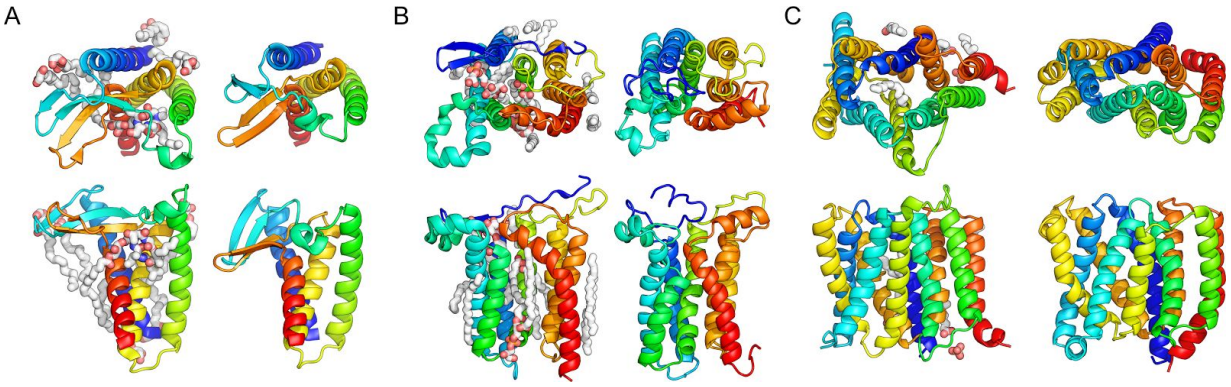

Fig. S8. Inaccuracies in models due to missing ligands. A) Lipoprotein signal peptidase II; B) Prolipoprotein diacylglyceryl transferase; C) the DMT superfamily transporter YddG. Left, crystal structures; right models. Top view on top row, side view on bottom row. In all three cases, the models overlap with ligands in the crystal structures shown in spheres.

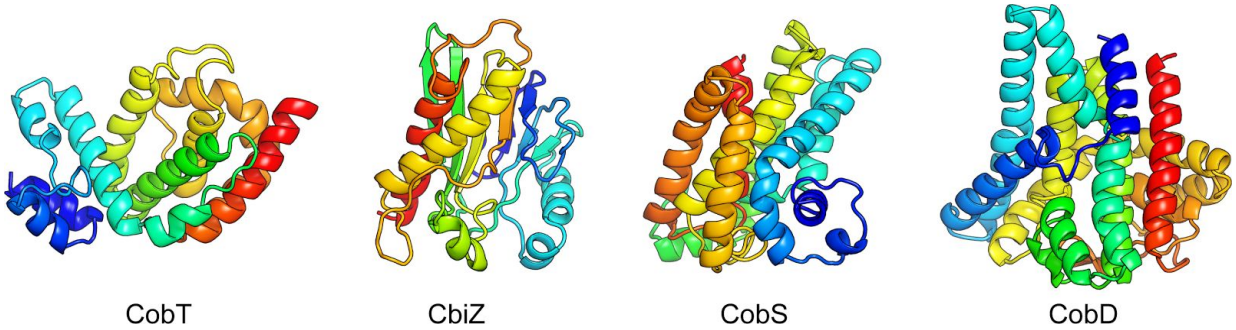

Fig. S9. Models for four proteins in the Cobalamin biosynthesis pathway.

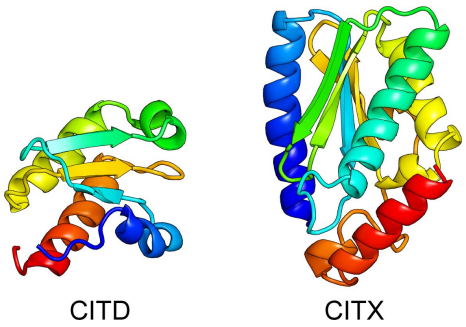

Fig. S10. Models for two subunits of citrate lyase.

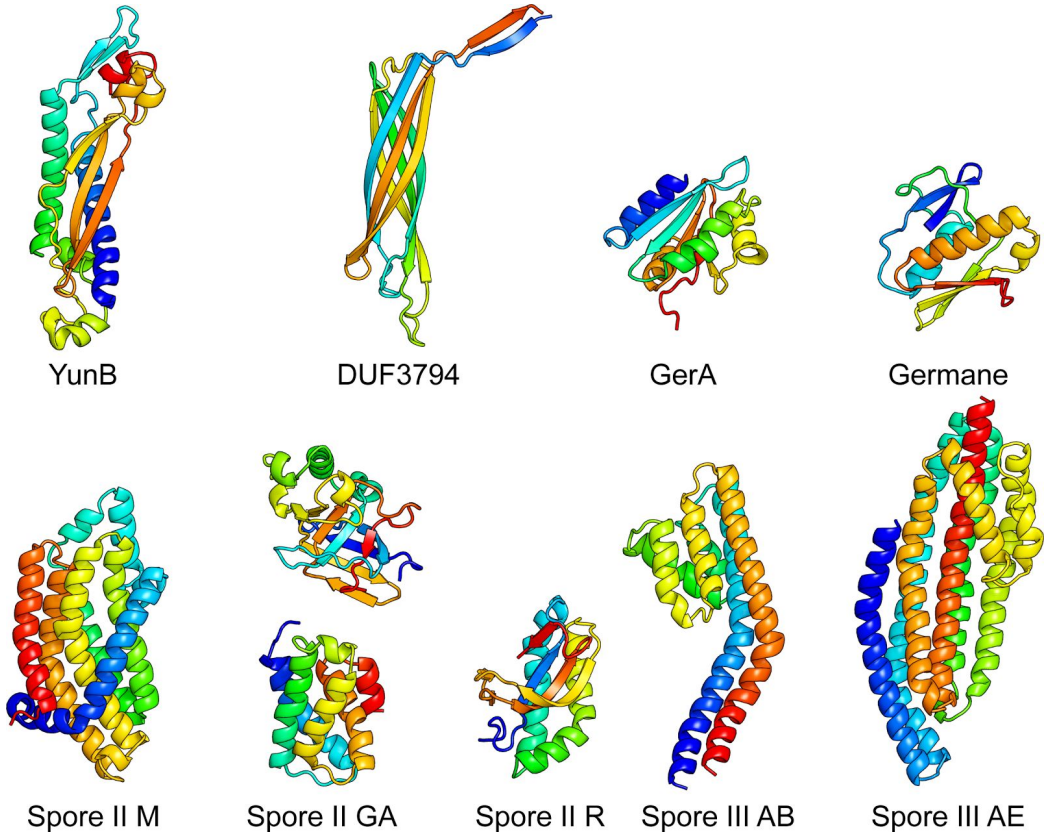

Fig. S11. Models of proteins involved in sporulation and germination

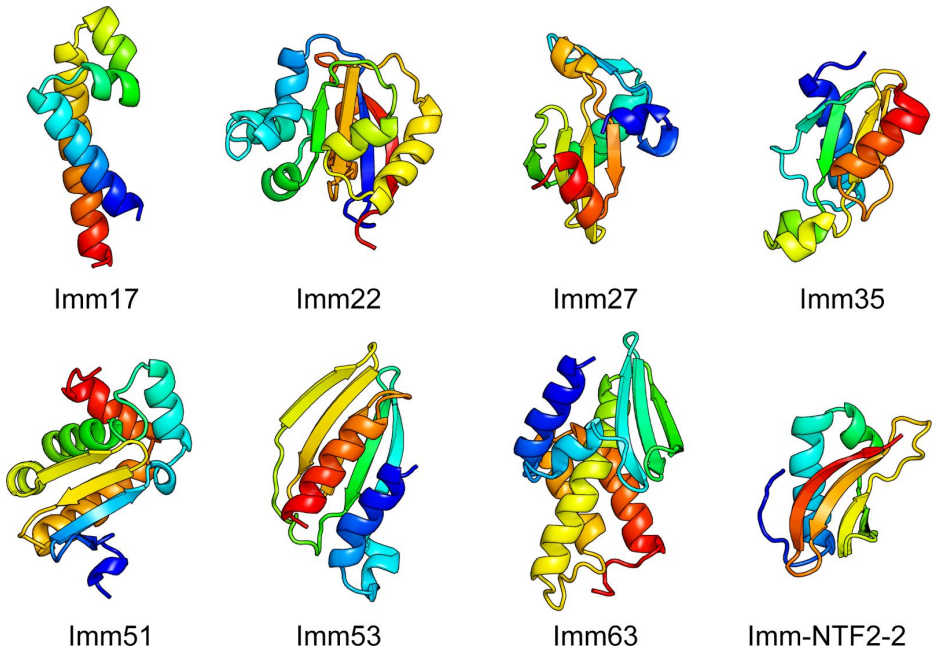

Fig. S12. Models of Bacterial immunity proteins.

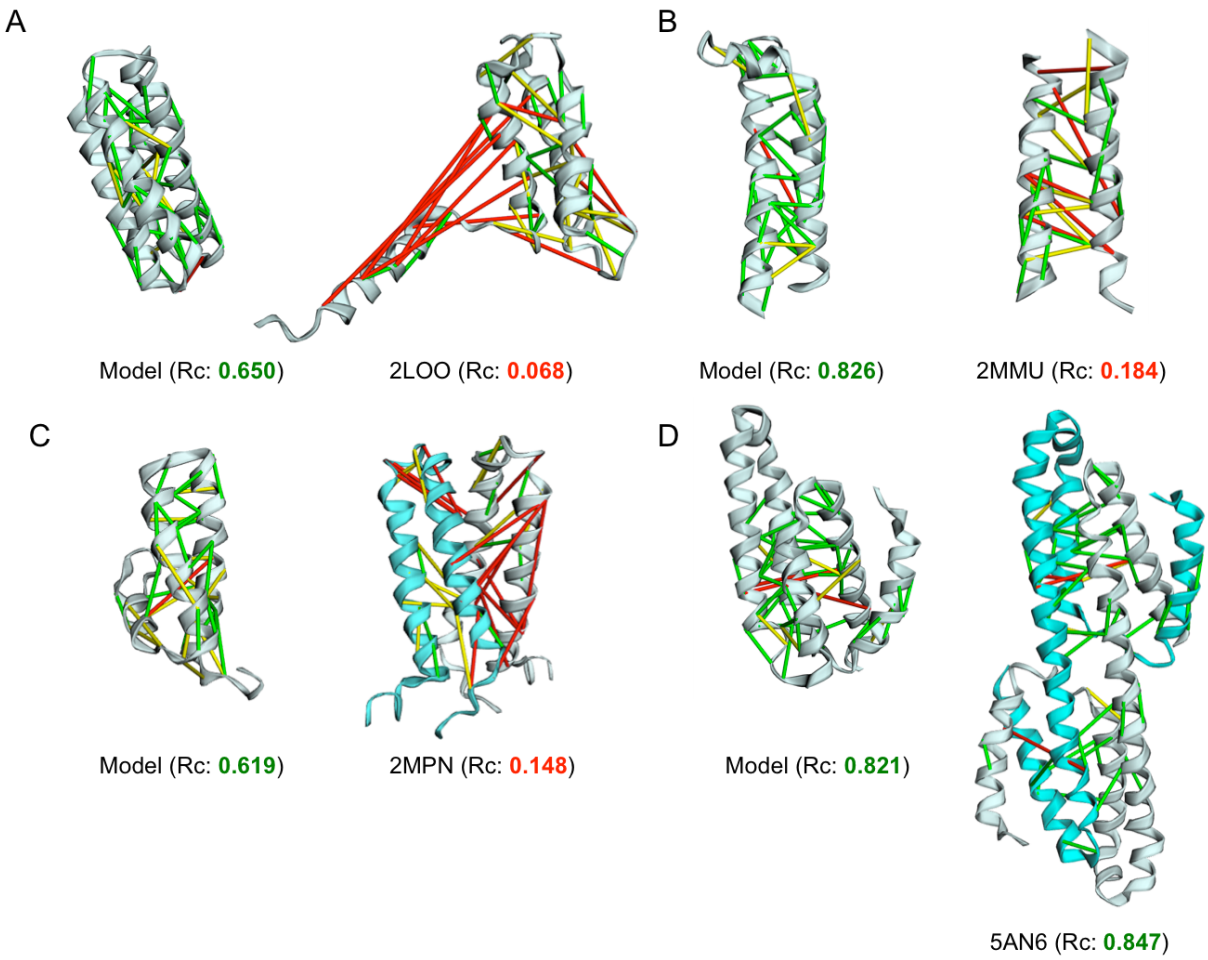

Fig. S13. Cases from the additional benchmark set that converged but did not agree with the experimentally determined structure. The top 3L/4 contacts are shown as lines. Green (less than 5 Å), Yellow (between 5-10 Å) and red (greater than 10 Å). Rc is a ratio of contacts made divided by the expected number of contacts (See ref *21*). 3 of the top 4 outliers (A-C) are all small transmembrane proteins; these include A) 2LOO: human membrane protein TMEM14A, B) 2MMU: CrgA, a Cell Division Structural and Regulatory Protein, C) 2MPN: inner membrane protein YgaP. D) 5AN6: Csm2 is an soluble x-ray structure that is also an intertwined dimer. The same set of contacts that are formed at the dimer interface can also be made at the monomer level.

Supplementary Tables

Table S1. Benchmark set used to evaluate the performance of the method at different Nf (number of effective sequences). Nf values and the TMscore of the top scoring model are reported for each target. The average across all targets is reported in Fig. 2A (solid line) and Fig. S1C (blue line).

Table S2. Nf values for each year (2009-2015) for each of the protein families with currently no detectable homologs using HHsearch before and after addition of metagenomic sequences. The HHsearch log10(E-value) and probability of PDB hit being the same fold is reported . For modeling we selected protein families from this list that had an E-value ≥ 1 and had at least Nf of 64 in the year of 2015 after the addition of metagenomic sequences.

Table S3. List of domains that converged in the structure prediction calculations and the top TMalign hit against the SCOP domain database.

Table S4. List of targets solved while this manuscript has been in preparation and their agreement to crystal structure. Since the newly determined structures have somewhat different sequences than the family representative we chose to model, a sequence independent measure (TMalign score) was used to evaluate model accuracy. In the table we report the convergence criteria (CON) and fraction of contacts made (RC). The RC value (described (*21*)) is the ratio of the number of contacts made divided by the number of contacts expected given the number of sequences and gremlin score.

Table S5. Additional benchmark with metagenomic sequences for testing modeling protocol. TMscore of the top scoring models is reported.

Table S6. The p-value for each of the datasets using the Wilcoxon signed-rank test. To compute these we used the R function wilcox.test(A,B,paired=TRUE,alternative = c("less")), where A are the TMscores from ihyb(DN) and B are the TMscores from ihyb(DN+MP). Abbreviations in the table: DN, de novo; ihyb(DN), iterative hybridization of de novo models; ihyb(DN+MP), iterative hybridization of de novo models with map_align partial threads. The improvement is significant at cutoff of 0.05 for the new benchmark set (of 81 proteins) and the old benchmark set (of 27 proteins) when Nf ≥ 32.