Journal of Bioinformatics and Computational Biology
Vol. 10, No. 1 (2012) 1240009 (16 pages)

DOI: 10.1142/S0219720012400094

Imperial College Press
www.icpress.co.uk

PROBABILISTIC ENSEMBLES FOR IMPROVED INFERENCE IN PROTEIN-STRUCTURE DETERMINATION*

AMEET SONI[†,§] and JUDE SHAVLIK[‡]

Department of Computer Sciences
Department of Biostatistics and Medical Informatics
University of Wisconsin—Madison
Madison, WI 53706, USA
[†]*soni@cs.wisc.edu*
[‡]*shavlik@cs.wisc.edu*

Accepted 23 November 2011
Published 3 February 2012

Protein X-ray crystallography — the most popular method for determining protein structures — remains a laborious process requiring a great deal of manual crystallographer effort to interpret low-quality protein images. Automating this process is critical in creating a high-throughput protein-structure determination pipeline. Previously, our group developed ACMI, a probabilistic framework for producing protein-structure models from electron-density maps produced via X-ray crystallography. ACMI uses a Markov Random Field to model the three-dimensional (3D) location of each non-hydrogen atom in a protein. Calculating the best structure in this model is intractable, so ACMI uses approximate inference methods to estimate the optimal structure. While previous results have shown ACMI to be the state-of-the-art method on this task, its approximate inference algorithm remains computationally expensive and susceptible to errors. In this work, we develop Probabilistic Ensembles in ACMI (PEA), a framework for leveraging multiple, independent runs of approximate inference to produce estimates of protein structures. Our results show statistically significant improvements in the accuracy of inference resulting in more complete and accurate protein structures. In addition, PEA provides a general framework for advanced approximate inference methods in complex problem domains.

Keywords: Statistical inference; protein-structure determination; computational biology.

1. Introduction

Over the past decade, the field of machine learning has seen a large increase in the use and study of probabilistic graphical models due to their ability to provide a compact representation of complex, multidimensional problems.[1] Recently, the

* An extended version of this paper can be found in *The Proceedings of the ACM International Conference on Bioinformatics and Computational Biology 2011*, Chicago, USA.

§Current address: Department of Computer Science, Swarthmore College, 500 College Avenue, Swarthmore, Pennsylvania 19081, USA. E-mail: soni@cs.swarthmore.edu.

complexity of problems posed in many areas of data analysis has stressed the ability to reason in graphical models. New techniques for *inference* are essential to meet the demands of these problems in an efficient and accurate manner.

One such application is our group's work on ACMI (Automated Crystallographic Map Interpretation), a three-phase, probabilistic method for determining protein structures from electron-density maps.[2–5] The task of determining protein structures has been a central one to the biological community, with recent years seeing significant investments in structural-genomic initiatives. X-ray crystallography, a molecular imaging technique, is at the core of many of these initiatives as it is the most popular method for determining protein structures. The final step of crystallography involves constructing an all-atom protein model from an *electron-density map* (a three-dimensional (3D) image). This step remains a major bottleneck in need of automation, taking months of manual effort by a crystallographer to solve.

Previous results show that ACMI outperforms other automated density-map interpretation methods on difficult protein structures, producing complete and accurate protein structures where other methods fail.[3] ACMI uses a graphical model known as pairwise Markov random field (MRF)[6] to combine visual features derived from the electron-density map with biochemical constraints in order to identify the most probable locations for each amino acid in the electron-density map. Unfortunately, exact inference (i.e. finding the best protein structure model) is intractable due to the complexity of the MRF. ACMI, instead, must employ *approximate* inference techniques to estimate each amino acid's location in the density map.

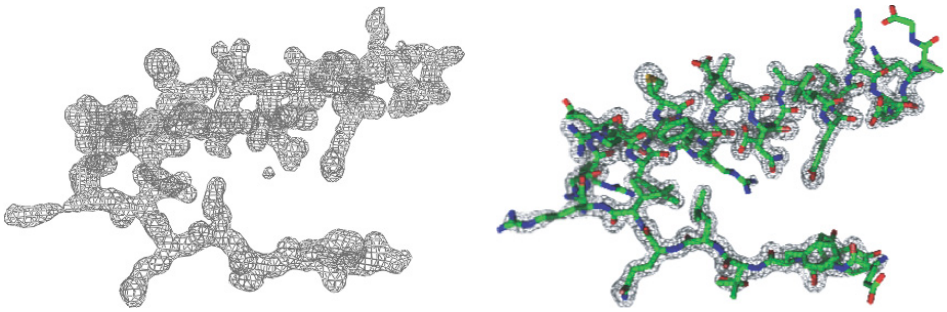
In this paper, we propose Probabilistic Ensembles in ACMI (PEA), a general framework for performing approximate inference in complex domains. Our previous approach produced a single probability estimate of the protein's location. PEA, instead performs multiple, independent runs of approximate inference in ACMI to produce multiple probability estimates of the protein's locations. Our results show PEA dramatically outperforms ACMI in both the quality of inference and accuracy of protein structures produced.

2. Background

2.1. *Protein X-ray crystallography*

Amino acids form the building blocks of proteins, linking end-to-end to form the linear protein sequence. The chain of atoms linking amino acids is known as the *backbone*, and the molecules hanging off of the backbone are called *side chains*. All side chains connect to the backbone via the Cα atom — the central atom in an amino acid — and are unique for each of the 20 types of amino acids.

X-ray crystallography is the most popular wet-lab technique for determining protein structures, producing ∼88% of protein structures in the Protein Data Bank.[7] The final step in the X-ray crystallography process is taking an electron-density map — a fuzzy, 3D image of a protein — and determining (or *interpreting*) the underlying protein molecular model that produced the image. Figure 1 shows a



(a) (b)

Fig. 1. The last step in the protein X-ray crystallography pipeline takes (a) an electron-density map (3D image) of the protein and finds (b) the most likely protein structure that explains the map.

sample density map and the resulting interpretation. Figure 1(a) is a contoured electron-density map, similar to what a crystallographer would see at the beginning of interpretation. In Fig. 1(b) we see the resulting protein structure with all non-hydrogen atoms in a stick representation. The crystallographer's task is: given a protein's amino-acid sequence and an electron-density map of the protein, produce the underlying protein structure.

Several factors make determining the protein structure a difficult and time-consuming process, mainly by affecting the quality of the electron-density map. The most significant factor is crystallographic resolution, which describes the highest spatial frequency terms used to assemble the electron-density map. Resolution is measured in angstroms (Å), with higher values indicating poorer quality maps with less detail. Additionally, crystallographers can only estimate the phases needed to calculate the electron-density map (the *phase problem*), reducing the interpretability of the image. Lastly, imperfections in the crystal structure and the stochastic nature of protein structure can create areas of distortion or smeared density in the image that contain very little or unreliable features.

The most popular method for automated density-map interpretation is ARP/wARP,[8] which efficiently finds solutions in maps with 2.7-Å resolution or better. TEXTAL[9] and RESOLVE[10] work on more difficult maps and have successfully interpreted density maps up to 3.2 Å in quality. A more detailed description and evaluation of these techniques can be found in our previous work.[2,3]

2.2. *Ensemble-learning methods*

Ensemble-learning methods come primarily from the supervised machine learning community. The goal of supervised learning is to develop a model (or classifier) with high predictive performance on future instances of a problem. Traditional learning methods yield a single-best model, $\hat{f}(x)$, to estimate the underlying (but unknown) true function, $f(x)$. Ensemble-learning methods, instead, develop a collection of

models, $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^N(x)$, that, in aggregate, produce a classifier with better performance than any single constituent model. Empirical evaluations of ensemble-learning methods (or *ensembles*) show that such methods outperform the best individual constituent models.[11,12]

There are two primary design choices in developing an ensemble-learning method. First, the learner must generate models that are *diverse*. A lack of diversity means each model will produce the same answer to a given instance and thus the collective performance will mirror individual performance. Second, the learner must aggregate the decisions (or predictions) of each model. This is often accomplished with majority voting, where each model gets a weighted or unweighted "vote" on the answer to a query instance.

While most work on ensembles is on supervised machine-learning problems, we are interested in structured-prediction problems such as ACMI. Weiss *et al.*[13] proposed Structural Ensemble Cascades (SECs), an iterative, hierarchical method for structured prediction problems. As in Wainwright *et al.*,[14] ensembles are used in inference, where an intractable graph is converted to a set of tractable (i.e. tree-structured) graphs.

3. Automated Crystallographic Map Interpretation

In previous work, our group developed ACMI (Automated Crystallographic Map Interpretation),[2–5] a probabilistic method for determining protein structures from low-quality electron-density maps (∼3 to 4 Å resolution). Figure 2 provides an overview of ACMI and its three-phase process. At the heart of ACMI is a probabilistic model known as MRF.[6] An MRF is an undirected graphical model that defines a probability distribution on a graph. *Vertices* (or nodes) are associated with random variables, and *edges* enforce pairwise constraints on those variables. In our task, ACMI seeks to probabilistically represent all possible structures of a protein in a compact manner. ACMI constructs a graph where each vertex describes the location, $\vec{u_i}$, of the Cα atom for the amino acid at position i in the sequence. Edges exist between every two amino acids in the sequence and model the interactions between the pair of connected amino acids. A sample MRF is show on the right side of the Phase 2 box in Fig. 2.

Formally, ACMI's MRF model $G = (V, E)$ consists of vertices $i \in V$ connected by undirected edges $(i, j) \in E$. We define the full-joint probability of all amino acid locations, $\mathbf{U}$, as

$$P(\mathbf{U}|\mathbf{M}) = \prod_{i \in V} \psi_i(\vec{u_i}|\mathbf{M}) \times \prod_{(i,j) \in E} \psi_{i,j}(\vec{u_i}, \vec{u_j}). \quad (1)$$

The first term, $\psi_i(\vec{u_i}|\mathbf{M})$, is associated with vertex i and is known as the *observation* potential function. It can be thought of as prior probability on the location of an amino acid given the map, $\mathbf{M}$, and ignoring all other amino acids in the protein. ACMI calculates this function in Phase 1 by calculating the correlation between

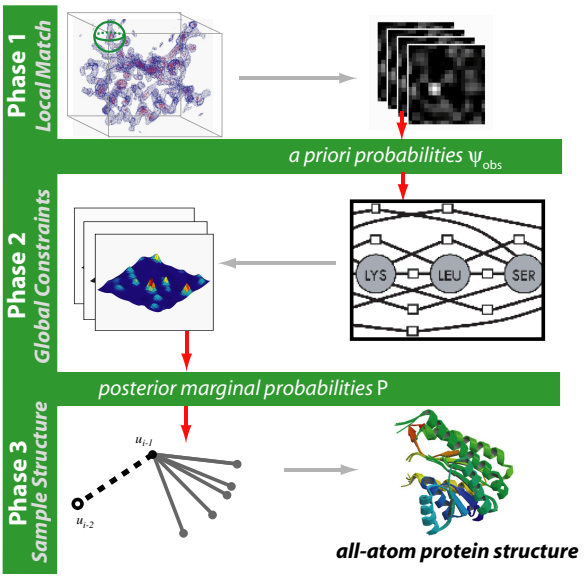


Fig. 2. The three-phase ACMI pipeline. Given an electron-density map and protein sequence, Phase 1 performs a local-match search independently for each amino acid. Phase 2 combines these results with global constraints to create posterior probabilities of each amino acid's location. Finally, Phase 3 uses these marginals to sample physically feasible, all-atom protein structures. The box on the right in Phase 2 shows a portion of an MRF for an example protein sequence.

(a) the electron-density map and (b) instances of amino acid i from previously solved structures from the Protein Data Bank (PDB).

The second term, $\psi_{i,j}(\vec{u_i},\vec{u_j})$, is associated with edges and represents one of two pairwise chemical constraints on the protein structure. Edges between neighboring amino acids in the linear sequence are represented by the *adjacency potential* — adjacent amino acids must maintain an approximate 3.8 Å spacing as well as proper angles (according to the distributions of bond lengths and angles in the PDB). Edges between non-neighboring amino acids contain an *occupancy potential* — no two amino acids can occupy the same space.

Given this MRF, we construct a three-phase pipeline (Fig. 2) to calculate the most probable protein structure for a given protein sequence and electron-density map. Phase 1 estimates the observation potential — the location of each amino acid in the density map independent of information about other amino acids. Phase 2 then takes these results and combines them with chemical constraints by performing inference on the MRF outlined earlier. Phase 3 uses these probabilities to sample physically-feasible, all-atom protein structures.

This paper concentrates on the role of inference in ACMI, which occurs in Phase 2. The model in Eq. (1) represents the full-joint probability distribution over all possible locations for each amino acid in the target protein. Calculating this probability exactly, however, is intractable due to the cyclical nature and large size

of ACMI's graph. ACMI, instead, employs loopy belief propagation (BP),[15] a fast approximate-inference algorithm, to calculate an approximate marginal probability distribution for the location of each amino acid's Cα atom.

Briefly, belief propagation calculates marginal probabilities by utilizing an iterative, local message-passing scheme to propagate information across a graphical model. A *marginal* probability represents the posterior probability of a single amino acid's location, incorporating all available information. A message, going from some amino acid i to some amino acid j, states, "Based on my current belief in my location, you should be here (with weight)." Details of ACMI's belief propagation implementation can be found in DiMaio *et al.*[2] and Soni *et al.*[5]

4. Methods

ACMI's Phase 2 utilizes an approximate inference technique known as loopy belief propagation (BP) to calculate the location of each amino acid in the protein sequence (see Sec. 3). Empirically, ACMI's Phase 2 rarely converges to a solution, and while ACMI performs well on difficult proteins, there are shortcomings in the inference process.[3,5] This section discusses the major contribution of this paper: the use of statistical ensembles to improve approximate inference solutions in ACMI.

4.1. *Probabilistic ensembles in ACMI*

With the well-documented success of ensemble methods in classification tasks, we seek to extend the idea of aggregating multiple estimates to probabilistic graphical models. As discussed in Sec. 2.2, current efforts in the area rely on simplifying the structure of an intractable graph to create a collection of tractable problems.[13,14] These techniques, however, do not easily extend to the graph in ACMI, which is fully connected and thus difficult to convert to the necessary number of tree-structured graphs. In addition, previous work in ACMI introduced an approximation that exploited redundancies in messages passing to dramatically reduce the complexity of inference.[2] Converting ACMI to a tree-structured graph loses the gains from this approximation as well as important information encoded in edges. Thus, unlike previous approaches, we are interested in an ensemble solution that boosts the *accuracy* of inference, not that *tractability*.

We propose PEA, shown in Fig. 3. PEA is a framework for generating and combining multiple approximate inference solutions to create more accurate protein structures. As with ensemble-learning methods in classification, there are two major design components to address: generating (diverse) solutions and aggregating multiple estimates.

4.1.1. *Generating ensemble components*

From Sec. 3, Eq. (1) calculates $P(U|M)$ — the probability distribution over all possible protein structures given the density map. Since this calculation is

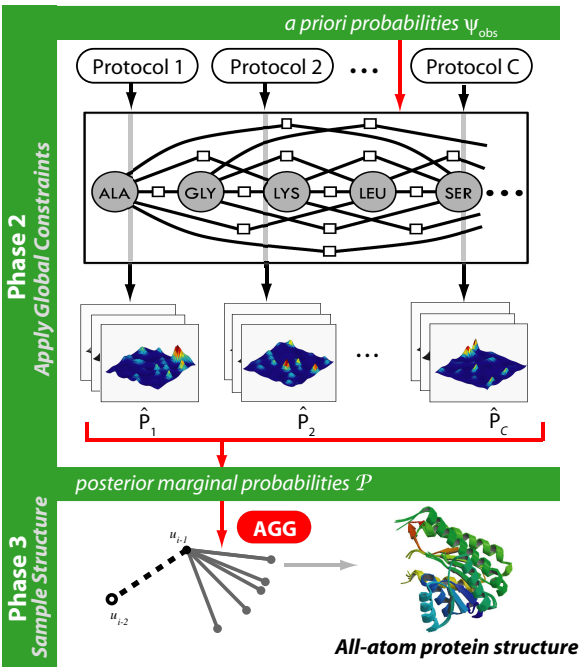


Fig. 3. Probabilistic Ensembles in ACMI (PEA). Phase 1 (omitted) is the same as in the ACMI framework (Fig. 2). Phase 2 performs C independent inference runs, each with a unique protocol. This results in a set of C marginal probabilities for each amino acid's location. Phase 3 aggregates the set of marginal probabilities to produce a protein structure.

intractable, Phase 2 of ACMI produces $\hat{p}_i$, the approximate marginal probability of each amino acid i's location. Rather than performing inference once, our proposed framework, PEA, performs several independent runs of inference. As shown in Fig. 3, each run (C in total) uses a unique protocol and outputs its own marginal probability distribution for each amino acid's location. Phase 2, in total, produces a matrix of probability distributions $\mathcal{P} = (\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_C)$, where each ensemble component c produces $\hat{P}_c = (\hat{p}_{1_c}, \hat{p}_{2_c}, \ldots, \hat{p}_{i_c})$. Here, $\hat{p}_{i_c}$ represents the probability of amino acid i's location in the density map according component c of the ensemble.

As mentioned in Sec. 2.2, a desired property of an ensemble is that the individual components are diverse. Fortunately, previous work[5] showed the choice of a message-passing protocol (i.e. what order to send and receive messages between nodes) has a large effect on the outcome of belief propagation in ACMI. Section 4.2 provides example protocols for generating ensemble components in PEA, each modifying how and when evidence is shared in the graph.

4.1.2. *Aggregating ensemble components*

In DiMaio *et al.*,[3] we developed Phase 3 of ACMI, which utilizes *particle filtering*,[16] a sampling algorithm, to generate all-atom protein structures given the posterior

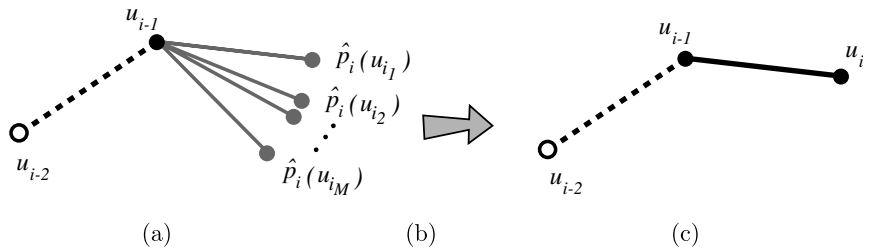


Fig. 4. ACMI's Phase 3 sampling step for amino acid i. In (a) Phase 3 samples M possible new locations, u_{i_m}. In (b) these locations are weighted by their agreement with the Phase 2 probability, $\hat{p}_i$. In (c) one location is chosen from the weighted distribution to be amino acid i's location, u_i.

marginal probabilities from Phase 2. Briefly, Phase 3 is an iterative process that sequentially grows a protein structure one amino acid at a time.[a] Figure 4 shows how, at a given iteration, Phase 3 samples the location, u_i, of amino acid i.

Phase 3 first samples M potential locations for the new amino acid based on the location of already placed amino acids in the sequence and a distribution of known angles and distances between neighboring amino acids. Next, Phase 3 assigns a weight, w_m, to each sampled estimate, u_{i_m}, correlated with the likelihood of amino acid i being in that location (i.e. the Phase 2 posterior marginal probabilities):

$$w_m \propto \hat{p}_i(u_{i_m}). \tag{2}$$

Lastly, Phase 3 chooses one of the M samples as its prediction for amino acid i's location in the structure. The sample is chosen with probability proportional to w_m.

In PEA, we no longer have one probability estimate for each amino acid's location, but C estimates. We propose several functions for combining these probabilities into a single weight measurement. First, we look at the average score over all ensemble components using a mixture model:

$$w_m \propto \sum_c \pi_c \cdot \hat{p}_{i_c}(u_{i_m}), \tag{3}$$

where π_c is the mixture weight, representing the confidence that component c provides the correct distribution.[b] The average score should perform well if the true location tends to have a high score across all runs of inference while false positives are uncorrelated between runs. False positives would be smoothed out and consistent peaks would maintain high probabilities.

[a] Phase 3 maintains multiple estimates (or particles) during the sampling process and uses separate steps to sample backbone and side-chain atoms. For simplicity, we only consider one particle's backbone placement in this description.

[b] π_c can be set by various measures, such as entropy or prior knowledge. We use uniform weights in our experiments.

Another proposed weight function is to instead take the maximum score for a given location across all components:

$$w_m \propto \max_c \hat{p}_{i_c}(u_{i_m}). \tag{4}$$

In difficult sections of a protein, it is very likely that most models will miss the correct location since there is very little evidence. Given multiple estimates, it is more likely that one model found the correct answer.

Lastly, we consider using a subsampling approach, where Phase 3 will randomly select one of the ensemble components to score the location:

$$\begin{aligned} c &\sim U[1, C], \\ w_m &\propto \hat{p}_{i_c}(u_{i_m}), \end{aligned} \tag{5}$$

where $U[1, C]$ returns an integer between 1 and C with uniform weight. This technique fits intuitively into the sampling framework of particle filtering where multiple structure estimates exist to explore several different paths to the end state.

4.2. *Experimental methodology*

In Sec. 5, we compare the performance of our original ACMI framework from DiMaio *et al.*[3] to our proposed algorithm, PEA. We use a set of 10 *experimentally phased* electron-density maps described in DiMaio *et al.*[3] for validation. This data was provided by the Center for Eukaryotic Structural Genomics (CESG) at UW−Madison. Based on the electron density quality, expert crystallographers selected these maps as the "most difficult" from a larger data set. These structures have been previously solved and deposited to the PDB, enabling a direct comparison with the correct model. However, all ten required a great deal of human effort to build the final atomic model. Test-set solutions are hidden from all phases of ACMI to prevent biasing results.

Phase 1 (performing an independent search for local features) is the same for both algorithms, meaning Phase 2 for both the original ACMI and proposed PEA algorithms begin with the same input. For the experiments in Secs. 5.1 and 5.2, we consider three variations of ACMI's belief-propagation protocol for Phase 2:

- ORIG, the original protocol of ACMI which is run for 40 iterations per amino acid in a round-robin fashion starting with amino acid 1, proceeding left to right, and then reversing at the end of each pass.
- EXT, an extended version of the original protocol going for 160 iterations.
- BEST, the top-performing individual version of ACMI from the four protocols considered for PEA (see below).

The BEST protocol provides an overly optimistic estimate of ACMI to see how PEA performs as an ensemble relative to its individual components. For PEA, we generate an ensemble of size 4 with each component having its own protocol:

- Protocol 1 is the same as ORIG above.
- Protocol 2 is similar to Protocol 1, but starts halfway through the sequence.

- Protocol 3 is similar to Protocol 1, but runs for 20 iterations
- Protocol 4 employs guided belief propagation introduced in Soni *et al.*[5]

For the learning curve in Sec. 5.3, 50 protocols were generated. All were based on the standard, round-robin schedule and executed for 40 iterations. Each varies in the starting location and the direction of the first iteration with half going left to right and the other half going right to left.

5. Results

Using the methodology described in Sec. 4.2, we compare the performance of our new approach of using Probabilistic Ensembles in ACMI (PEA) against the original ACMI algorithm.[3] We compare the results across a set of 10 difficult protein structures. Previous results show ACMI is the state-of-the-art technique for low-quality protein images, thus related approaches are not compared in this paper. Section 5.1 first assesses the quality of approximate inference by comparing the accuracy of the Phase 2 outputs by the two approaches. In Sec. 5.2, we feed these Phase 2 probabilities to Phase 3 to measure the accuracy of the all-atom protein structure models produced by PEA and ACMI. Lastly, Sec. 5.3 shows how the accuracy of PEA changes as the number of ensemble components increases.

5.1. *Approximate inference*

Our first experiment assesses the quality of approximate inference solutions produced in Phase 2 for both ACMI and PEA by examining the accuracy of posterior marginal probabilities. In this experiment, ACMI and PEA use the same Phase 1 outputs to run their respective Phase 2 algorithms and halt before executing Phase 3. PEA runs Phase 2 with four ensemble components using the protocols specified in Sec. 4.2. We consider the maximum score aggregator from Eq. (4) and the average score aggregator from Eq. (3) (MAX and AVG, respectively). The sampling algorithm from Eq. (5) performs aggregation as a step in Phase 3 and cannot be compared here. For ACMI, we test the original, round-robin protocol (ORIG), an extended run of inference (i.e. 160 iterations) in ACMI (EXT), and the best-performing *individual* component of PEA (BEST).

Figures 5(a) and 5(b) show the results of running these techniques on a set of difficult protein images. Figure 5(a) shows the percentile rank which represents how highly ranked the correct solution (i.e. location from the deposited structure in the PDB) is in the posterior marginal probabilities. The optimal score of 100 means the true location had the highest probability value in the map. In Fig. 5(b), the negative log-likelihood is the probability value for the true location, transformed as a negative-log score. Here, we desire lower values as they indicate higher probabilities.

Both figures show that the ensemble method, PEA, drastically outperforms the existing, single inference version of ACMI across all protocols. Both the maximum and average aggregators obtain scores in the 89th percentile compared to the

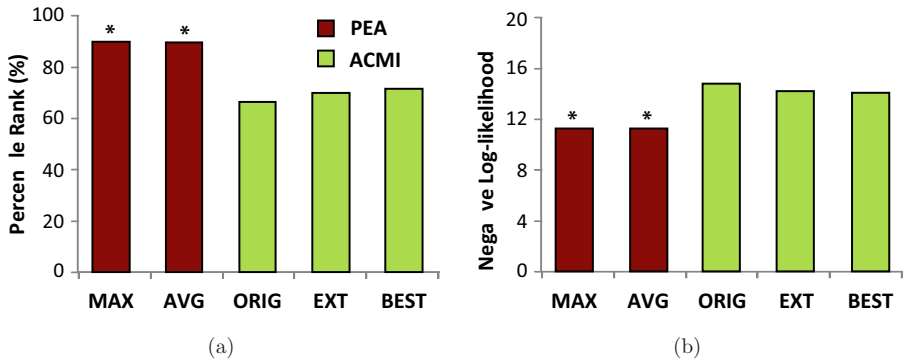


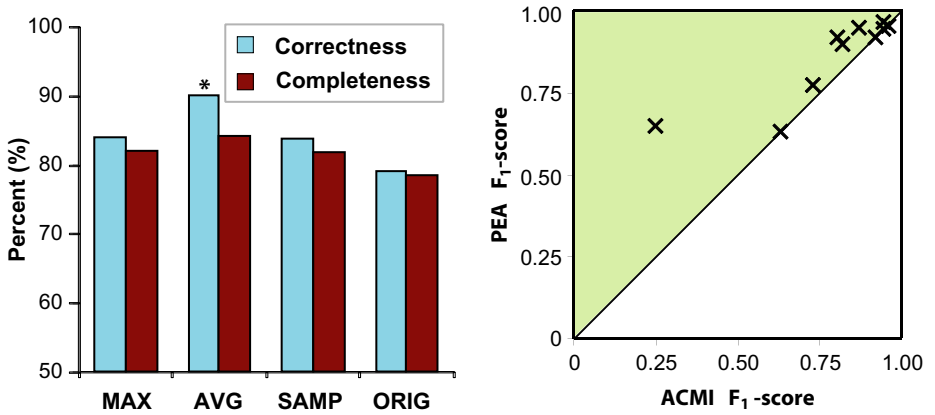
Fig. 5. Accuracy of inference solutions. In (a), the percentile rank of the true solution's probability is shown. In (b), the negative log-likelihood of the true solution is shown. Lower scores mean a higher probability value for the correct answer. In both, columns are the average score over all amino acids in all test-set proteins. Dark (red) bars represent variations of PEA, while light (green) bars represent variations of ACMI. *denotes a statistically significant difference with ORIG at $p < 0.01$.

original ACMI protocol which averages scores in the 66th percentile. This implies that, on average, there are three times as many false positives in ACMI versus PEA. The negative log-likelihoods tell a similar story; the probability scores improve by over three orders of magnitude by using ensembles. The results for the best individual component of PEA are only slightly better than standard ACMI, showing that PEA benefits from combining multiple, good models rather than from generating one very good model. The extended run of standard ACMI shows minor improvements as well, but comes nowhere near the performance of PEA, showing that the gains of our ensemble method cannot be explained away by an increase in CPU resources. In fact, the results of all pairwise differences between the PEA variations and the three ACMI variations are statistically significantly at scores of $p < 0.01$ for both metrics in Fig. 5 based on a paired t-test.

5.2. *Protein structures*

While the previous results indicate our ensemble technique improves the accuracy of approximate inference probabilities, biochemists are more interested in the actual protein structures produced. As a follow-up experiment, we used the marginal probabilities from Sec. 5.1 as the input for Phase 3 of the ACMI and PEA algorithms, respectively, to produce all-atom protein structures for all 10 of our test-set proteins. We use the completeness and correctness of the resulting protein structures to compare our proposed aggregators for PEA against ACMI.

Figure 6(a) shows the averaged results of our experiments. The first three pairs of columns represent the maximum (MAX), average (AVG), and sampling (SAMP) aggregators for PEA presented in Sec. 4.1.2. The fourth pair of columns represent the original ACMI protocol. Within each pair, the first column represents the *correctness* of the predicted protein structure — what percentage of amino acids



(a) (b)

Fig. 6. Protein-structure prediction accuracy. In (a), we show correctness (light blue) and completeness (dark red) of PEA. *indicates statistically significant difference compared to ORIG at $p < 0.05$. In (b) we show a detailed comparison of F_1-scores for ACMI (x-axis) and PEA using the averaging aggregator (y-axis). Each point represents one protein and the shaded region indicates better scores for PEA.

predicted were within 2 Å of their corresponding true-solution location. This is similar to the *precision* metric used in information retrieval. The second column represents the *completeness* of the predictions — what percent of amino acids available in the PDB solution were accurately predicted (within 2 Å). This is akin to a *recall* metric. Each column represents an average over all 10 test proteins. The top performer across both metrics was PEA using the averaging function to aggregate ensemble components. On average, 90.3% of its predicted amino acid locations were correct (compared to 79.3% for the original ACMI algorithm) while completing 84.3% of the real structure (78.6% for ACMI). Importantly, all three PEA methods outperform ACMI in both correctness and completeness measures.

Figrue 6(b) provides a closer comparison of PEA versus ACMI. Here, each datapoint represents the results of one protein in our test set. The x-axis value is the accuracy of the original ACMI algorithm and the y-axis is the accuracy of PEA using the average aggregator. To assess accuracy, we use an F-measure to combine the correctness and completeness metrics from Fig. 6(a). The F-measure is commonly used in the information retrieval community to balance both the need for high precision and high recall. Here, we use the traditional F_1 metric, which is the harmonic mean of correctness and completeness.

The line represents equivalent performance, and the shaded region represents values, where PEA outperforms ACMI. In every test case, PEA performs better than or equal to ACMI in the F_1 metric, affirming the results from Fig. 6(a). The largest improvement comes in the most difficult test case, with the F_1-score improving from 0.25 to 0.66. This corresponds to an extra 41 percentage points of the true structure being built and 42 percentage points of extra predictions being

correct. Overall, PEA shows substantial improvement in 6 of the 10 proteins, with equal performance in the other 4, although these values are not statistically significant.

Figure 6(b) only considers the average aggregator for PEA since it performed better than the alternative options. As hypothesized in Sec. 4.1.2, the averaging aggregator's main advantage is that it can smooth away "noisy" probabilities. The maximum aggregator and sampling aggregator also produced improved inference probabilities but did not translate into the same level of improvement in structure quality as the averaging aggregator. It is difficult to pinpoint the exact reason, but the areas of major difference happened to be in regions of the map with this least amount of signal, implying the averaging aggregator handles noise the best.

5.3. *Ensemble learning curve*

As a final experiment, we consider how the size of an ensemble effects the accuracy of inference in PEA. Due to resource limitations, we could not run larger ensembles sizes for the previous experiments. Instead, for the seven smallest test-set proteins, we generated ensembles with various number of components, ranging from 1 to 50. We assessed each using percentile scores as described for Fig. 5(a). Figure 7 shows the learning curve for seven of our test-set proteins as the number of ensemble components increases (the values past 30 are not shown since no change occurred). PEA uses the mixture-model average aggregator to combine posteriors. As Fig. 7 shows, PEA gains accuracy from adding more components, making its largest leap in performance with the first 10 ensemble components before seeing very little improvement after 20 component ensembles.

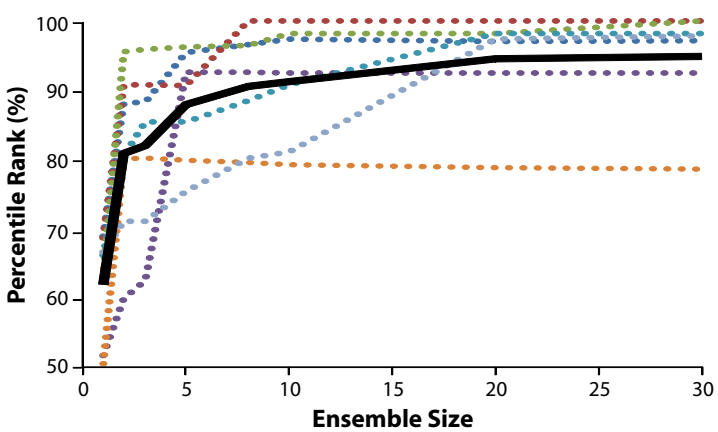


Fig. 7. Learning curve for ensemble inference. Each dashed line represents one protein's percentile scores for Phase 2 posteriors as the number of ensemble components increases. The solid black line represents the average learning curve.

6. Conclusions and Future Work

While ACMI was previously shown to outperform other automated density-map interpretation methods in building all-atom protein structures in low-quality electron-density maps,[3] performing approximate inference in ACMI's model is an expensive process in need of advanced inference methods. In this work, we developed a new approximate inference method based on the concept of ensemble-learning methods from the supervised machine learning community. Our new framework, PEA, executes several independent runs of inference to provide multiple, diverse solutions to the problem. We suggest several protocols for generating unique solutions for each component of the ensemble as well as different techniques for aggregating these models to produce a single accurate prediction of the protein structure.

Our results show PEA provides improved performance on a test-set of 10 difficult protein images. This improvement is seen in the accuracy of the inference process, where the probability distributions from PEA were statistically significantly better in terms of both percentile rank and probability value assigned to the correct location of each amino acid. The results show that this improvement could not be explained by extra CPU resources or by using the single-best component of PEA. More importantly, PEA's improved inference translates into more complete and correct protein structures. Future work will gather a larger set of evaluation proteins, including membrane proteins which present many difficulties for crystallographers.

While we presented ensembles of approximate inference solutions for the task of protein-structure determination, our method can generally be applied to difficult inference problems where the complexity of probabilistic graphical models limits the accuracy of current methods. In future work, we look to find such applications, and to provide an in-depth comparison to related inference techniques that rely on simplifying the graph structure.[14]

Acknowledgments

This work is supported by NLM grant R01-LM008796 and NLM training grant T15-LM007359. Support for our collaborators at the University of Wisconsin Center for Eukaryotic Structural Genomics (CESG) is provided by NIH Protein Structure Initiative Grant GM074901. ACMI and the data set of experimentally-phased density maps is available online at http://www.cs.wisc.edu/acmi/.

References

1. Koller D, Friedman N, *Probabilistic Graphical Models: Principles and Techniques — Adaptive Computation and Machine Learning* (MIT Press, Cambridge, 2009).
2. DiMaio FP, Shavlik JW, Phillips GN, A probabilistic approach to protein backbone tracing in electron-density maps, *Bioinformatics* **22**(14):e81−89, 2006.
3. DiMaio FP, Kondrashov DA, Bitto E, Soni AB, Bingman CA, Phillips GN, Shavlik JW, Creating protein models from electron-density maps using particle-filtering methods, *Bioinformatics* **23**:2851−2858, 2007.

4. DiMaio FP, Soni AB, Phillips GN, Shavlik JW, Spherical-harmonic decomposition for molecular recognition in electron-density maps, *Int J Data Mining Bioinformatics* **3**(2):205−227, 2009.
5. Soni AB, Bingman CA, Shavlik JW, Guiding belief propagation using domain knowledge for protein-structure determination, *Proc 1st ACM Int Conf Bioinformatics and Computational Biology* (Niagara Falls, NY, USA, 2010).
6. Geman S, Geman D, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans Pattern Anal Mach Intell* **6**:721−741, 1984.
7. Protein Data Bank (PDB), PDB current holdings breakdown, August 2011. Available at http://www.rcsb.org/pdb/statistics/holdings.do.
8. Perrakis A, Morris R, Lamzin V, Automated protein model building combined with iterative structure refinement, *Nat Struct Mol Biol* **6**(5):458−463, 1999.
9. Ioerger T, Sacchettini K, The TEXTAL system: Artificial intelligence techniques for automated protein model building, *Method Enzymol* **374**:244−270, 2003.
10. Terwilliger T, Automated main-chain model building by template matching and iterative fragment extension, *Acta Crystallog D* **59**(1):38−44, 2003.
11. Dietterich T, Ensemble methods in machine learning, *Lecture Notes in Computer Science* **1857**:1−15, 2000.
12. Maclin R, Opitz D, An empirical evaluation of bagging and boosting, *Proc 14th Nat Conf Artificial Intelligence* pp. 546−551, 1997.
13. Weiss D, Sapp B, Taskar B, Sidestepping intractable inference with structured ensemble cascades, in Lafferty J, Williams C, Shawe-Taylor J, Zemel R, Culotta A (eds.), *Advances in Neural Information Processing Systems*, Volume 23, pp. 2415−2423, 2010.
14. Wainwright M, Jaakkola T, Willsky A, Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching, *Workshop on Artificial Intelligence and Statistics*, 2003.
15. Murphy K, Weiss Y, Jordan M, Loopy belief propagation for approximate inference: An empirical study, *Proc 15th Conf Uncertainty in Artificial Intelligence*, 1999.
16. Arulampalam MS, Maskell S, Gordon N, Clapp T, A tutorial on particle filters, *IEEE Trans Signal Process* **50**:174−188, 2001.

Ameet Soni received his Ph.D. in 2011 from the University of Wisconsin−Madison in the Department of Computer Sciences for his work on probabilistic inference in protein-structure determination. Dr. Soni is currently is a Visiting Assitant Professor in the Department of Computer Science at Swarthmore College in Swarthmore, Pennsylvania. His research interests are in the field of machine learning, and particularly in applications to problems in biology and medicine.

Jude Shavlik is a Professor of Computer Sciences and of Biostatistics and Medical Informatics at the University of Wisconsin−Madison and is a Fellow of the American Association for Artificial Intelligence. He has been at Wisconsin since 1988, following the receipt of his Ph.D. from the University of Illinois for his work on Explanation-Based Learning. Dr. Shavlik's current research interests include machine learning and computational biology, with an emphasis on using rich sources of training information, such as human-provided advice.