Contact Prediction using Mutual Information and Neural Nets

George Shackelford
Kevin Karplus*

September 24, 2007

Abstract

Prediction of protein structures continues to be a difficult problem, particularly when there are no solved structures for homologous proteins to use as templates. Local structure prediction (secondary structure and burial) is fairly reliable, but does not provide enough information to produce complete three-dimensional structures. Residue-residue contact prediction, though still not highly reliable, may provide useful guide for assembling local structure prediction into full tertiary prediction.

We develop a neural network which is applied to pairs of residue positions and outputs a probability of contact between the positions. One of the neural net inputs is a novel statistic for detecting correlated mutations: the statistical significance of the mutual information between the corresponding columns of a multiple sequence alignment. This statistic, combined with a second statistic based on the propensity of two amino acid types being in contact, results in a simple neural network that is a good predictor of contacts.

Adding more features from amino-acid distributions and local structure predictions, the final neural network predicts contacts better than other submitted contact predictions at CASP7, including contact predictions derived from fragment-based tertiary models on free-modeling domains.

It is still not known if contact predictions can improve tertiary models on free-modeling domains.

Availability: `http://www.soe.ucsc.edu/research/compbio/SAM_T06/T06_query.html`

Keywords: contact prediction, CASP7, SAM_T06, neural net, significance of mutual information in contingency tables, gamma distribution

1 Introduction

Predicting the structure of a protein given its sequence continues to be a difficult problem in molecular biology. Knowing the structure of a protein aids in determining its function, but experimental determination of structure is still slow and expensive. The protein sequence databases are growing much faster than the protein structure databases, so structure prediction is becoming an increasingly important problem.

The current best methods involve constructing 3D models based on using templates and fragments selected from known protein structures. Local structure predictions are also used to guide the model building.

Residue-residue contact predictions are predictions of the probability that the two residues are within close proximity of each other. If we can predict even a small set of residue pairs that have high probability of being close to each other, we can use these predictions as extra constraints to guide the model building. This paper discusses only the contact prediction itself and not its application to tertiary modeling.

2 Definitions

There are several different definitions of residue-residue contact in common use. The most physically meaningful one is that atoms of the two residues are within Van der Waals distance of each other, but this definition is not commonly used. In this paper we follow the definition used by the CASP assessors, that *two residues are in contact* if their respective C_β atoms are within 8 Å of each other (C_α for glycines). This definition has the advantage of depending mainly on backbone conformation and not sidechain rotamer, and so is easier to predict. It has the flaw that residues on adjacent strands of a beta sheet may be deemed to be in contact even though they are on opposite faces.

We are working on developing definitions that correspond better to intuitive notions of contact, but which retain the predictability and ease of computation of the CB-8 definition, but these are outside the scope of this paper.

The *separation* of two residues is defined as $|i-j|$ where i and j are the indices of the two residues in the sequence. The likelihood that two residues are in contact varies greatly with their separation (see Figure 1), and all null models for contact prediction should be based on separation.

*Biomolecular Engineering
University of California
Santa Cruz, CA 95064
karplus@soe.ucsc.edu
phone: 1-831-459-4250

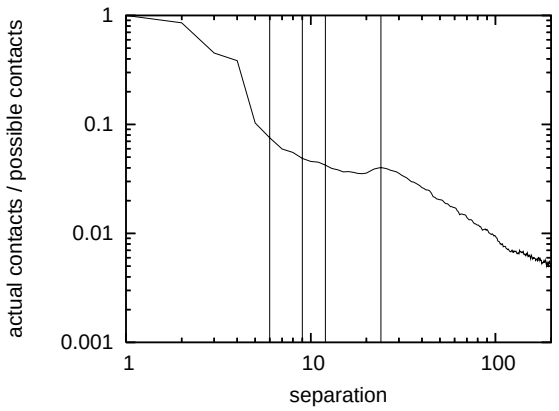

Figure 1: Using the definition of contacts from CASP7 (C_β within 8Å), we plot the probability that two residues are in contact versus their separation along the chain. The data are drawn from a low-redundancy set of 2191 chains from the Protein Data Bank (PDB). The lines at 6, 12, and 24 represent the minimum separation for predictions in the three CASP7 categories of contact predictions. The line at 9 represents the minimum separation used for training and testing of our neural network.
We speculate that the dip at $4 <$ separation < 24 is caused by the prevalence of long helices that reduce the probability of contact. Alternatively, the peak at 24 may be caused by alternating α/β structures such as TIM-barrels.

3 Inputs for Prediction

3.1 Multiple Sequence Alignments

All of our predictions are based on multiple sequence alignments (MSAs) of protein sequences. We take sequences from NCBI's non-redundant protein database NR[1] that are sufficiently similar to our target sequence and align them to the target sequence. When we predict contacts, we are predicting contact between columns of the multiple sequence alignment, so we chose a multiple alignment procedure where the columns correspond exactly to the residues of our target sequence.

The most commonly used program for selecting sequences and creating the multiple sequence alignments is psiblast[2], but we used our SAM-T04 hidden-Markov model method[3] for selecting and aligning the sequences.

3.2 Thinning

When aligned sequences are highly similar the signals we are looking for are easily confused with other signals cause by the repetition of sequences from particular subfamilies. To reduce this artifact, we *thin* our MSAs with a simple greedy procedure. We will mark each sequence either to be kept or to be discarded. The sequences in the MSA are examined one at a time and are compared to all sequences already kept. If the residue identity is too high between the new sequence and any already kept sequence, the new sequence is discarded. There is a trade-off between the diversity and number of the kept sequences—more aggressive thinning provides less duplication and so less confusion from sample biases, but fewer sequences and so less sensitive tests for correlated mutation.

We tested various paired-column statistics thinning to 35%, 40%, 62%, and 70% residue identity. Most paired statistics did best at 62% while propensity (discussed in Section 3.5) did best at 40%. We settled on using a thinning of 50% for paired statistics we used in training the neural network.

3.3 Paired Column Statistics

To predict whether a pair of residues are in contact, we compute various statistics from the corresponding columns of the multiple sequence alignment. These statistics may be specific to a single column or to the pair of columns, but the most informative statistics are those that are computed for a pair of columns.

All of the pairwise statistics can be computed from a 20×20 contingency table $C_{x,y}$, which records how often each pair of amino acids x and y occurs in the two columns of the MSA. Although one could also include gap characters to make 21×21 contingency tables, we have omitted sequences which have a gap in either column when constructing the tables and computing the pair statistics. Note: we used no sequence weighting in constructing the contingency tables, although some statistics may have improved predictive value if such weighting were used.

Our pair statistics come in two types: those that look for a correlation of mutations in the two columns and those that look for amino acid types that are particularly likely to make contacts (paying no attention to mutation information).

3.4 Correlated Mutation Statistics

When one residue mutates in a protein structure, there is a possibility that there will be a compensating mutation nearby[4]. Such pairs are called *correlated mutations* or *co-evolving mutations* and form the basis for many contact prediction methods. Fodor and Aldrich[5] published an evaluation of several of such statistics including mutual information, however we ended up using two other correlation statistics, the first of which is a novel statistic based on mutual information.

Mutual Information E-value We found mutual information to be a poor contact predictor (data not show), despite the theoretical attractiveness of the statistic. Our hypothesis was that the problems arose because of the sparseness of the contingency tables, and that the observed mutual information was more a function of small-sample effects than of the correlated mutation signal we were interested in.

What we want to determine is how significant the observed mutual information is, given the marginal counts. To do this we need to estimate a distribution of mutual information values for random contingency tables that have identical marginal counts, then use that distribution to estimate the probability of seeing the observed mutual information by change (the p-value). (We report the result as an E-value, which is just the p-value multiplied by the number of pairs of columns tested.)

We construct random contingency tables with the same marginal counts (this is equivalent to randomly permuting one of the two columns) and measure the mutual information of the random tables. When we did the computations for large numbers of random tables, we observed that the resulting distributions were well fit by a gamma distribution (data not shown). We have no theoretical justification for the gamma distribution, but the fit seems too good to be accidental. For densely filled tables, the mutual information asymptotically approaches a χ^2 distribution[6], but we could find no theoretical analysis of mutual information for sparse contingency tables.

Because we have many pairs of columns to evaluate, our prediction method only generates 50 random tables per pair, and uses moment matching to fit a gamma distribution. For the $4L$ (where L is the length of the sequence) pairs which have the lowest estimated p-values, we recompute the p-values using 500 random tables.

This statistic is the best single statistic we have evaluated for contact prediction (comparison not shown), and we incorporate it into our final neural network. The other correlation statistic we use as input to the neural network is joint entropy.

Joint Entropy measures how much variation there is in pair of columns:

$$\text{Ent} = \sum_{x,y} p(x,y) \log p(x,y) \ .$$

Pairs of highly conserved positions, which have low joint entropy, are more likely to be in contact than highly variable positions. The joint entropy also provides an upper bound on the mutual information for a pair of columns.

3.5 Propensity

We also use a paired-column statistic that is not a correlation statistic. The *propensity* for two residue types a, b to be in contact is the log odd of a contact between the residues versus the probability of the residues occurring independently: $\log\left(P(a,b)/(P(a)P(b))\right)$. Propensity mainly encodes hydrophobicity (and cysteine clustering)[7].

We use a slightly modified propensity that weights high-separation contacts more than low-separation contacts. For a large training set, we do a weighted count of contacts for each pair of amino-acid types:

$$W_{a,b} = \sum_{i,j:i\geq j+9} \frac{\delta(r_i = a, r_j = b, i \text{ contacts } j)}{P(\text{contact}|\text{sep} = i-j)} \ ,$$

where a, b are residue types and r_i, r_j are the residues at positions i and j. We also count the number of potential contacts:

$$N_{a,b} = \sum_{i,j:i\geq j+9} \delta(r_i = a, r_j = b) \ .$$

The propensity is the log of the ratio of these counts:

$$\text{Prop}_{a,b} = \log(W_{a,b}/N_{a,b}) \ .$$

We can use propensity as a statistic for a pair of columns by averaging the propensities for all the pairs of residues. While not the best pair statistic of those we examined, propensity is surprisingly powerful and uses different information than the correlation measures.

3.6 Rank vs. Value as Input

When we use paired statistics as features, we need to consider whether to use the statistic's value for a pair of columns or the rank of the statistic's value in the list of values for all pairs of columns. For single statistics, it does not matter whether we use rank or the raw value of the statistic, but it does matter when we try to combine multiple statistics using a neural net. In that case, we found that using rank generally worked better (data not shown).

4 Single-column Statistics

We have not used the single-column statistics alone for making contact predictions, but do use them to supplement the paired-column statistics. For each alignment column, we compute the probability of each amino acid (using a Dirichlet mixture regularizer[8]), the entropy of this amino-acid distribution, predicted secondary structure (using the 13-class str2 alphabet[3]), and predicted burial (using the 11-class near-backbone-11 alphabet[3]).

4.1 Input Windows

The results used paired features for only the (i, j) pair itself. However besides single-column features for i and j, we have examined using windows of single-column features $i \pm n$, $j \pm n$. In training and testing, we exclude indices where the window contains unknown or undefined residues such as beyond the two ends of the sequence. We experimented with values for n from 1 to 4, and found that a value of 2 (window width=5) gave the best results.

5 Neural Networks

We use a neural network for classification of pairs as in contact or not. Neural networks make excellent classifiers, even though they do not make explanations of the reasons for the classification easy.

Neural networks take a vector of numeric values as inputs and output a classification value between 0 and 1—ideally a probability, though the neural net package we are using, Fast Artificial Neural Network (`fann`)[9], does not provide such well-calibrated outputs.

They are trained by supervised learning, with a training set of feature vectors and the correct classifications. Each pass over the training set is called an *epoch* and training consists of a series of epochs. The weights of the neural network can be adjusted either after each example (on-line) or after each epoch (batch). For this work, we used batch training, with improved resilient back-propagation[10] which has been shown to work well in classification problems[11].

Because there are considerably more negative examples than positive examples, we had to deal with whether to balance the input or not. Balancing the inputs makes it easier for the training algorithm to learn, but results in miscalibrated outputs. A preliminary experiment with the naturally unbalanced input had poor results, with the neural net predicting no contacts. We finally settled on balancing by randomly selecting 5% of the negative examples each epoch. This way the neural network was likely to see most of the negative examples over the course of the training, while providing enough balance to keep the neural net from getting stuck at zero. A different training protocol and objective function might have avoided the need to balance the inputs and provided better calibrated outputs.

While neural networks are capable of learning which input features are relevant to the classification, they generally perform much better if redundant and irrelevant inputs are removed. Much of our work in creating a contact predictor was in identifying which features were most useful for the predictions.

5.1 Training Data

For training, cross-training, and testing of neural nets, we started with a set of 1335 proteins from the structures in the Protein Data Bank (PDB)[12]. The set was selected by the Pisces server[13] to contain chains with resolution ≤ 1.8Å with R-factor of 0.25, with no two sequences having more that 30% identity similarity. A few were were removed due to excessive clashes, microheterogeneity, or shortness, resulting in 1329 proteins. We divided this into a set of 421 chains with no missing backbone atoms in the PDB files and the remaining set of 908 chains. The set of 421 was used for neural network training with a subset of 350 sequences used for training and 50 for cross-training (the other 21 were not used). We arbitrarily selected a set of 300 sequences from the set with missing backbone atoms for final testing (we had to limit the size of the sets due to software and memory limitations).

Because our library of multiple alignments was built using only residues that had coordinates for all four backbone atoms, our training and testing set contains many alignments in which the target sequence has insertions relative to the alignment columns. We make no predictions for these insertion residues.

To reduce computational cost and meory usage in the neural nets, we generate the statistics for all possible pairs, but only output pairs which come in the top $4L$ pairs (where L is the length of the protein) on at least one of the statistics that we are evaluating. The resulting set of pairs is invariably larger than $4L$ and provides a good sample for both training and testing. Using this enriched set reduces the problem of unbalanced training for the neural net, so long as the same enrichment procedure is used for both training and prediction.

5.2 Neural Net Training

The contact-prediction neural nets are trained with RPROP back-propagation[10]. Every four epochs we test with the cross-training set. We terminate training when 100 epochs pass with no improvement on the cross-training set at $L/2$ and $L/10$ predictions. We output the network with the best scores on the cross-training set during the entire training run. Results are reported for the performance of this network on the testing set, which was not used at all in the training.

6 CASP7 Network And Testing Results

For CASP7, we trained a network with the following features: log(sequence length), log(separation), single column inputs (using a window of five):

- amino acid distribution (20 classes)
- secondary structure predictions (13 classes)
- residue burial predictions (11 classes)
- entropy (single inputs, no window),

and paired-column inputs:

- number of pairs
- mutual information e-values (value and log rank)
- propensity (log rank)
- joint entropy (log rank).

Figure 2 compares the CASP7 network on the full cross-validation test with single pair features and the best neural network that combined two pair features. The extra inputs of the CASP7 network make a substantial improvement in performance.

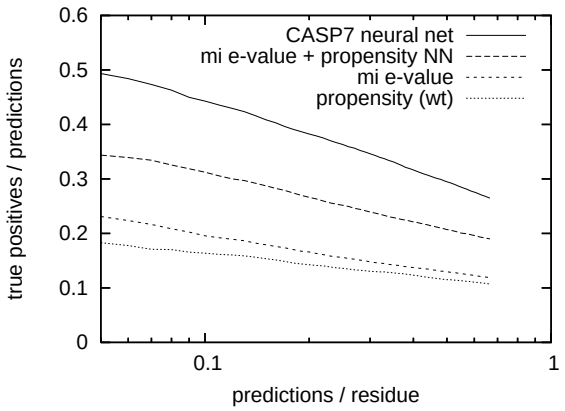

Figure 2: The plot shows accuracy vs. coverage for the full cross-validation test. The final CASP7 predictor uses 449 inputs compared to the next highest which uses only 5 inputs: weighted propensity (value and rank), mutual information e-value (value and rank), and log of separation. The accuracy for the column-pair statistics alone are included for comparison.

7 CASP7 Predictions

For CASP7 evaluation, the targets were split into domains and the domains into two difficulty classes: template-based domains and free-modeling domains. Only the free-modeling target domains were used for assessing the residue-residue contact predictions by the assessors. Here we consider both classes in our analysis.

To gauge the potential usefulness of our contact predictor, we compared our predictions to ones derived from the Baker group's models, which did best in tertiary predictions for free-modeling domains. We create contact predictions from their tertiary models by sorting pairs based on increasing residue-residue distance and taking the closest $L/10$ pairs with separation ≥ 9. Figure 3 shows the comparison between our contact predictor and contacts extracted from the Baker group's first models for template-based and free-modeling domains.

In general, for template-based domains, using the template results in much better predictions than using our neural net, though there are a surprising number of points where the neural net did better. For free-modeling domains, our contact predictions are generally much better than the Baker group's.

Although we do better at contact predictions than tertiary prediction methods for free-modeling domains (at least, when only $L/10$ predictions are made), the overall accuracy on such domains is substantially lower than on template-based domains (see Figure 4). The difference in accuracy stems mainly from the difference in the number of sequences in the multiple alignments, but there are exceptions. The free-modeling targets T0353 and T0382 have accuracy 0.67 and 0.58 at L/10 predictions, while

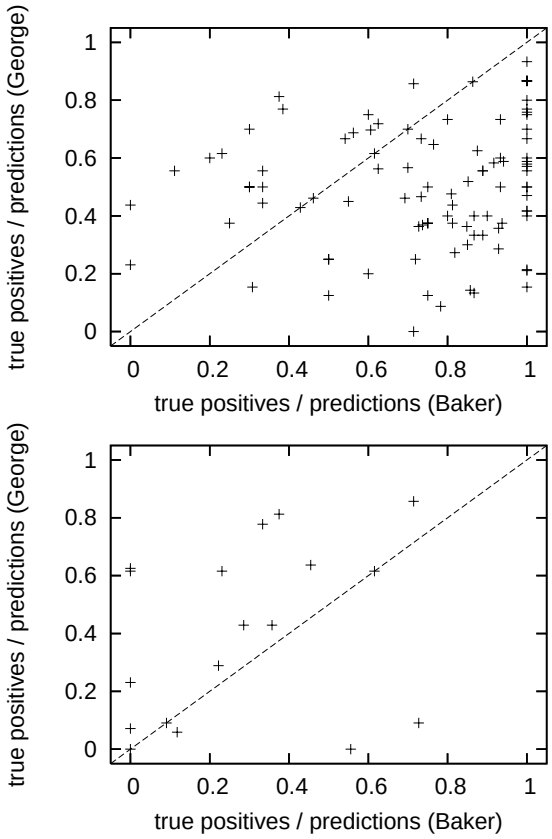

Figure 3: Two scatter-plots comparing contact predictions by our neural network compared to contact predictions drawn from the Baker group model 1. The diagonal line separates predictions where we do better (above the line) compared to when the Baker group does better. The first plot is for targets considered template-based by the assessors. The second plot is for the more difficult free-template targets. Both sets are limited to the best 0.1*(sequence length) predictions and separation ≥ 9.

having 6 and 7 sequences in the MSA respectively. If we can determine why these worked so well, we may be able to improve the accuracy of predictions for other MSAs with few sequences.

8 Conclusions and Future Work

We introduced the use of gamma distributions to estimate the significance of mutual information in sparse contingency tables, and we showed that this statistic is a good predictor of residue-residue contacts. Supplementing the E-value of mutual information with propensity and local properties predicted from the sequence produced a neural network that does a very good job of predicting contacts.

The neural net can undoubtedly be improved by using somewhat different combinations of input features and by training on a larger training set. We plan to explore various ways of improving the network.

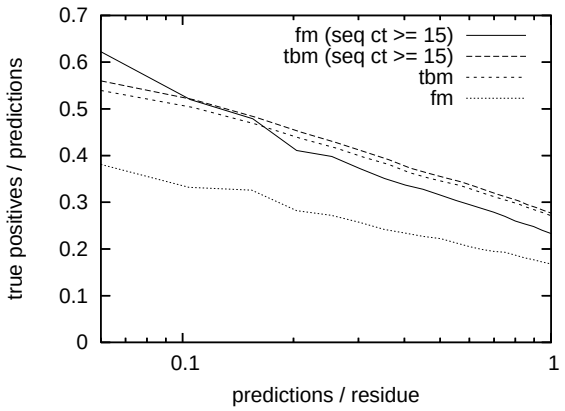

Figure 4: We plot accuracy vs. predictions per residue for free-modeling (fm) and template-based modeling (tm). The difference in quality appears to be due to the number of sequences in the multiple alignments—when we restrict our predictions to only those targets that have 15 or more sequences in the multiple sequence alignment, the difference in accuracy between free modeling and template-based modeling domains almost disappears.

For free-modeling targets, our neural network already provides a limited (L/10) set of contact predictions that are better than ones derived from the the best group's first model tertiary predictions. Despite this, we have so far no proof that using the contact predictions results in better 3D models.

We plan to test the usefulness of contact predictions by providing them as soft constraints to a model builder, and determining whether the model builder creates better models with or without the predictions. One drawback of this approach is that it depends how sensitive the model builder (e.g., UNDERTAKER[3], Modeller[14], or rosetta[15]) is to errors in constraints.

Another approach for using contact prediction is as a quality assessment method for models from a wide variety of servers. This avoids the problem of dependence on a specific modeling program, but does not allow the modeling programs to use the information about predicted contacts, which may be too limiting.

Acknowledgments

This work was supported in part by NIH grant 1 R01 GM068570-01.

References

1. NR (All non-redundant GenBank CDS translations+PDB+SwissProt+PIR+PRF Database) Distributed via anonymous FTP from `ftp://ftp.ncbi.nlm.nih.gov/blast/db`. Information on NR is available at `http://www.ncbi.nlm.nih.gov/BLAST/blast_databases.html`.
2. Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E., and Altschul, S. F. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Research 29(14):2994–3005, July, 2001.
3. Karplus, K., Katzman, S., Shackleford, G., Koeva, M., Draper, J., Barnes, B., Soriano, M., and Hughey, R. SAM-T04: What's new in protein structure prediction for CASP6. Proteins: Structure, Function, and Bioinformatics 61(S7):135–142, 2005.
4. Göbel, U., Sander, C., Schneider, R., and Valencia, A. Correlated mutations and residue contacts in proteins. Proteins: Structure, Function, and Genetics 18:309–317, 1994.
5. Fodor, A. and Aldrich, R. Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. Proteins: Structure, Function, and Bioinformatics 56:211–221, 2004.
6. Schervish, M. Theory of Statistics, 459. Springer-Verlag, 1995.
7. Cline, M. S., Karplus, K., Lathrop, R. H., Smith, T. F., Rogers Jr., R. G., and Haussler, D. Information-theoretic dissection of pairwise contact potentials. Proteins: Structure, Function, and Genetics 49(1):7–14, 1 October, 2002.
8. Sjölander, K., Karplus, K., Brown, M. P., Hughey, R., Krogh, A., Mian, I. S., and Haussler, D. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. Computer Applications in the Biosciences 12(4):327–345, August, 1996.
9. Nissen, S. and Nemerson, E. Fast artificial neural network. `http://fann.sourceforge.net/`, October, 2004.
10. Riedmiller, M. and Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, 586–591 (IEEE, San Francisco, CA, 1993).
11. Igel, C. and Hüsken, M. Empirical evaluation of the improved Rprop learning algorithm. Neurocomputing 50(C):105–123, 2003.
12. Bernstein, F., Koetzle, T. F., Williams, G. J., Meyer, E. E., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T., and Tasumi, M. The Protein Data Bank: a computer-based archival file for macromolecular structures. Journal of Molecular Biology 112:535–542, November, 1977.
13. Wang, G. and Dunbrack, Jr., R. L. PISCES: a protein sequence culling server. Bioinformatics 19:1589–1591, 2003.
14. Sali, A. and Blundell, T. Comparative protein modelling by satisfaction of spatial restraints. Journal of Molecular Biology 234:779–815, 1993.
15. Simons, K. T., Bonneau, R., Ruczinski, I., and Baker, D. Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins: Structure, Function, and Genetics Supplement 3(1):171–176, 1999.