An Optimized Neural Network for Contact Prediction

George Shackelford

Kevin Karplus

University of California, Santa Cruz

Using Contact Predictions

- 3D structure prediction is hard.
- Local structure predictions like secondary structure predictions are good.
- Tools for searching fold space are good but challenged by complexity.

With contact predictions we would use a small but accurate number of contact predictions as constraints in Undertaker, Rosetta.

But contact prediction is hard.

Residue-Residue Contact Definitions

Contact between residues is not actual contact (i.e. van der Waals distance).

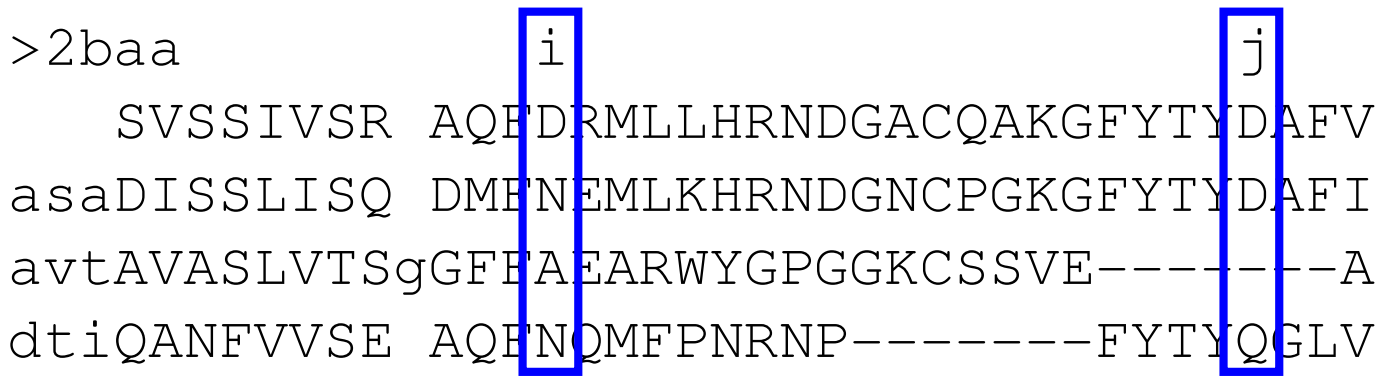
- CASP: Contact between two residues i, j is when the distance between their respective C_β atoms is less than 8 Å.
- We define *separation* as $|i - j|$

Method: Neural Network

- Upside: can provide excellent classification.
- Downside: black box - gives little or no information about feature relationships.
- Software based on `fann`, fast artificial neural network.
- Used Improved Resilient Back-propagation.
- CASP6 approach: used all inputs we could.
- CASP7 goal: use good inputs while eliminating weak or redundant inputs.

Multiple Sequence Alignment

We use multiple sequence alignments from SAM-t04 as a source of evolutionary data:



We have features for single columns, i and j, and for paired columns, (i, j).

Thinning the Sequence Alignment

If the sequences are too similar, we tend to see false correlations.

We use *thinning* to reduce the sample bias.

To thin a MSA to 50%, we remove sequences from the set until no pair of sequences has more than 50% percent identity.

- 80% thinning and sequence weighting for single column features.
- 50% thinning and NO weighting for paired features.

Single-column Features

- Distribution of residues in the column.
  - Regularized by using mixtures of Dirichlet distributions.
- Entropy over distribution.
- Predicted local features.
  - A secondary structure alphabet (str2)—13 classes.
  - A burial alphabet—11 classes

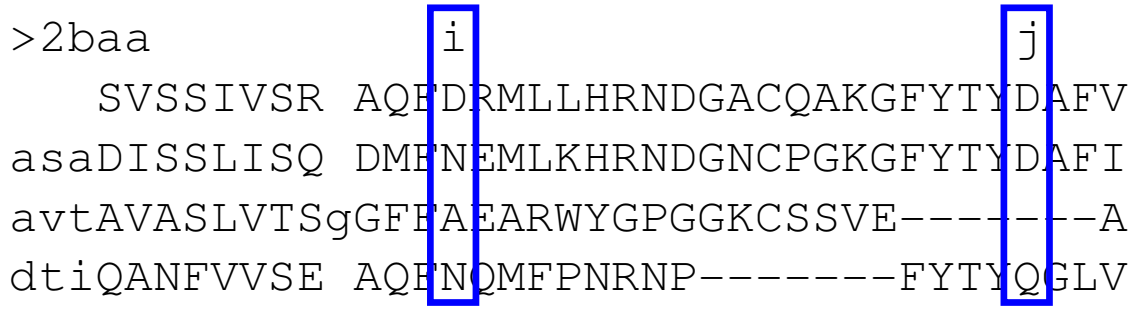
Inputs: Using Windows

For single columns we input values from features for
$i-2, i-1, i, i+1, i+2$,
$j-2, j-1, j, j+1, j+2$.
Tests indicated this window width was the best.

Exception is entropy with no window.

$(20+13+11)*5*2+2=442$ inputs—so far!

Paired-columns Features



Yields pairs: `DD, ND, NQ`. No pairing with gaps.
For features:

- Contact propensity
- E-values from mutual information
- Joint entropy
- Number of pairs between the two columns
- Log(|i-j|)

Contact Propensity

The log likelihood two amino acids (A, L) are in contact.

- Contact propensity is log(prob(contact(x, y))/prob(x)prob(y)).
- Contact propensity is largely due to the hydrophobicity (M. Cline et al. '02).
- Some very small part is due to other signals.
- We average the propensity over all sequences.
- Results show a significant increase in the signal.

Correlated Mutations

When a residue in a protein structure mutates, there is a possibility that a nearby residue will also mutate in compensation.

- beta bridges
- sidechain-sidechain interactions
- functional regions

We can detect these correlated mutations with correlation statistics.

Mutual Information

$$\mathrm{MI}_{i,j} = \sum_{k=1}^{T} p(r_{i,k}, r_{j.k}) \log \frac{p(r_{i,k}, r_{j,k})}{p(r_{i,k}) p(r_{j,k})}$$

where $r_{i,k}$ is the residue in column i, pair k.

- Mutual information is a very weak predictor by itself.
- We can improve by calculating an E-value over possible MI values.

Mutual Information E-value

- Shuffle residues in one column and calculate the mutual information value.
- Repeat 500 times recording the MI values.
- Determine parameters for Gamma distribution by using moment matching.
- Use that distribution and original MI value to derive a p-value.
- Derive E-value from p-value.

Joint Entropy

$$\mathrm{Ent}_{i,j} = \sum_{x \in R} \sum_{y \in R} \frac{C_{x,y}^{i,j}}{T} \log \left(\frac{C_{x,y}^{i,j}}{T} \right)$$

- i and j represent the indices of the pair of columns,
- R is the set of twenty residues and T is the number of valid residue pairs,
- $C_{x,y}^{i,j}$ is the count of amino acid pairs, x, y, for columns, i, j.

There are a LOT of Pairs

- We track only the top (10*length) values for each statistic.
- We sort each list according to value to get a rank.
- We calculate the Z-values using means and s.d. over all pairs ($i + \text{separation} <= j$).
- We form a final set over the intersection of the lists.

We keep data on value, Z-value, and rank.

Use Rank and/or Value for Inputs?

We experimented with using the rank, values, and Z-values of the pair values.

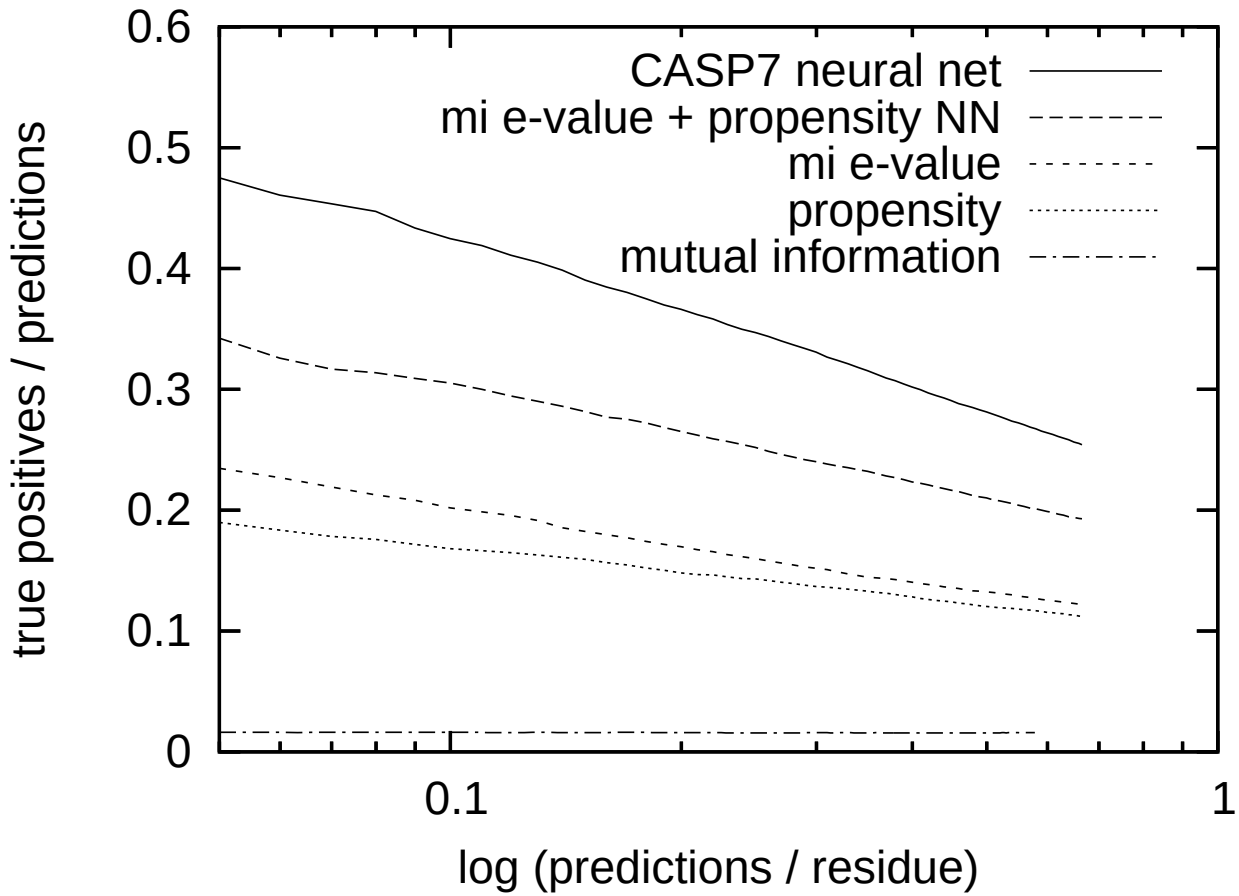
- For contact propensity we use -log(rank).
- For MI E-values we use -log(rank) and Z-value.
- For joint entropy we use -log(rank).

Misc. Input

for input number 449:

- Log of sequence length.

Evaluation: Comparing Predictors



Results for separation $>= 9$.

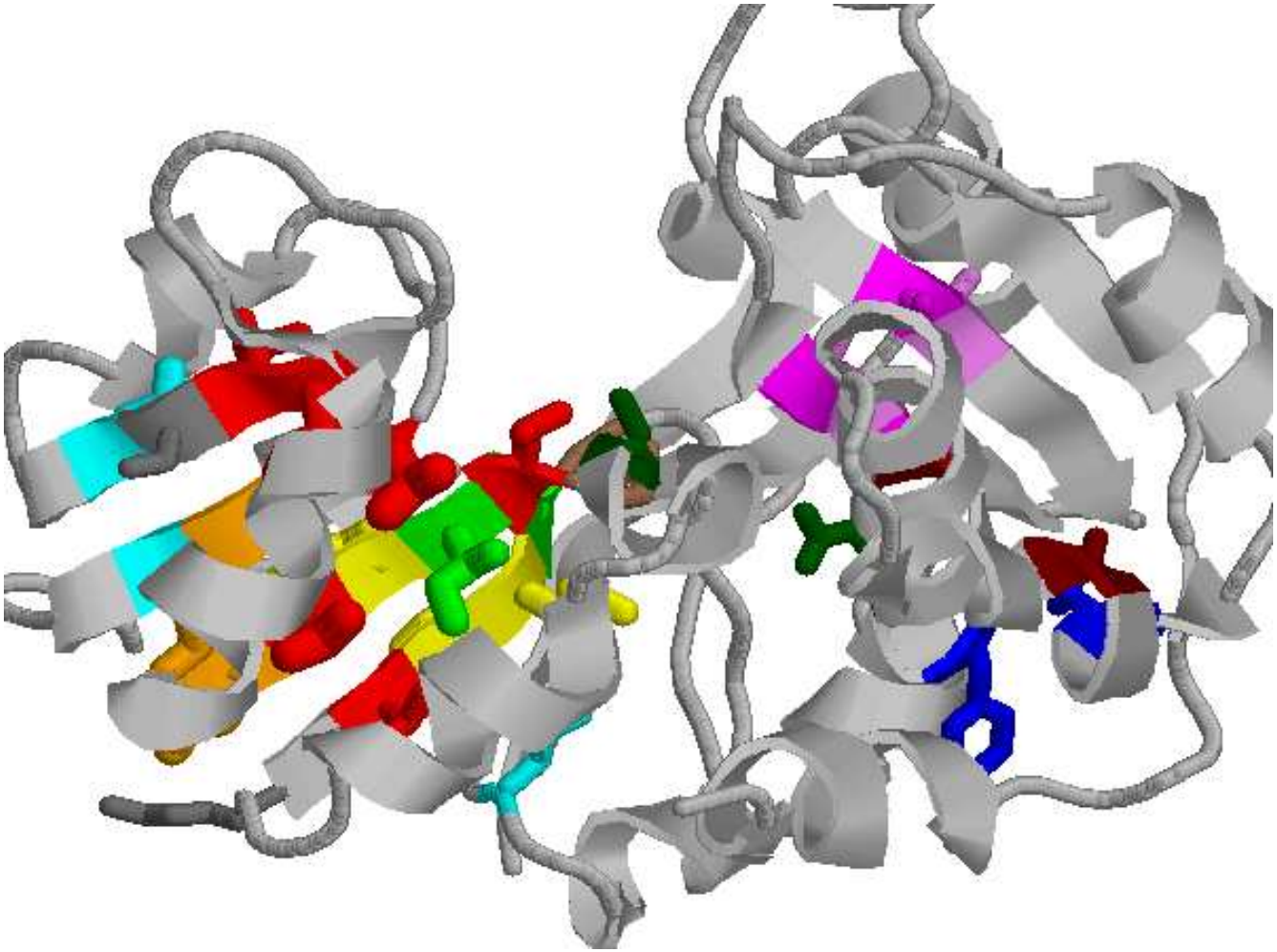
CASP7 Results

For 0.1 predictions/residue and separation $>= 12$.

- We show two good results: T0321 and T0350.
- We show a bad result: T0307.
- We compare accuracy to difficulty of target
  - using BLAST E-values.
  - using Zhang Server GDT.
- We compare accuracy to number of sequences in MSA.
- We examine how confident can we be in the neural net output scores.

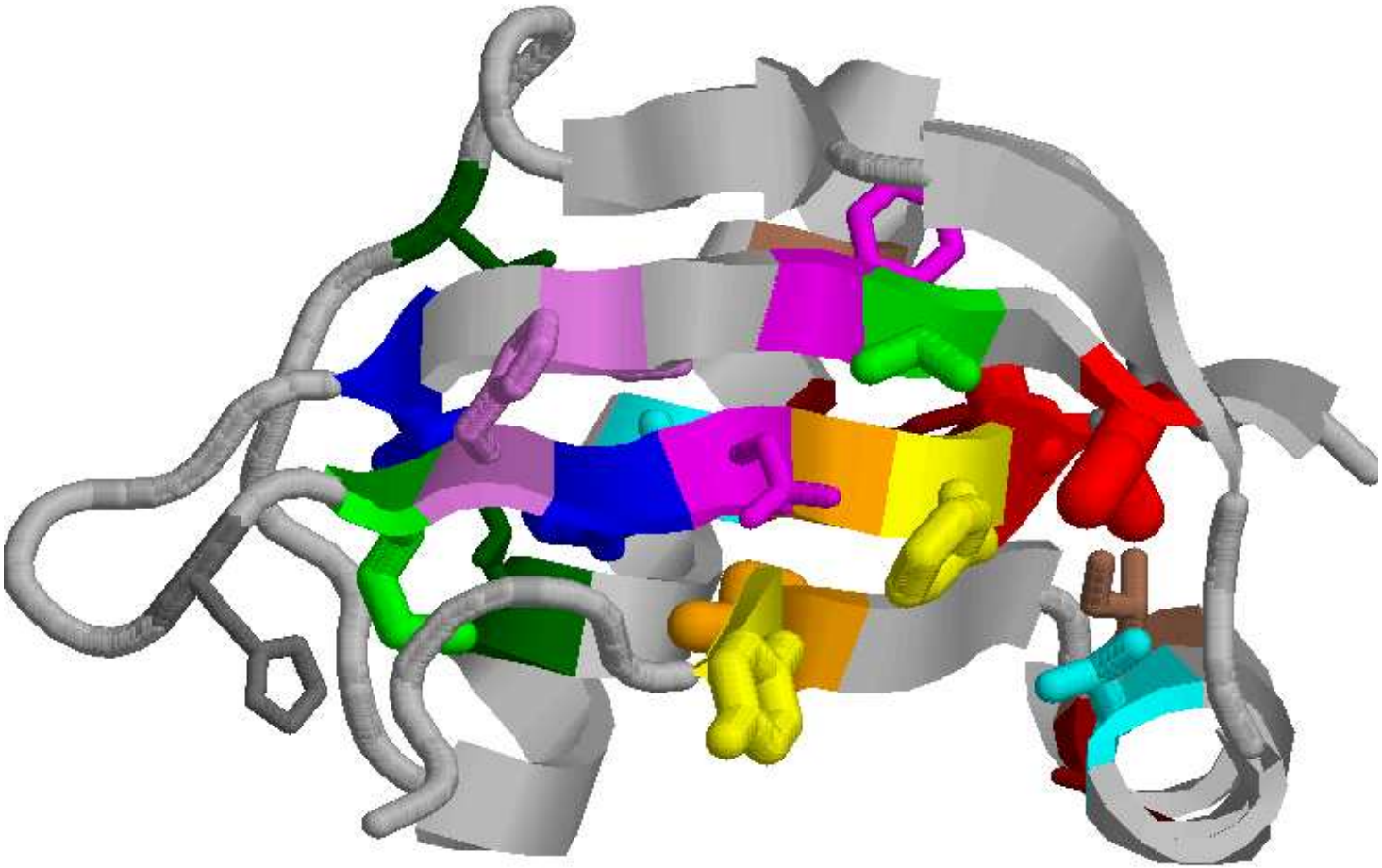
The Good: T0321

Thickness of side-chains represents neural net output.

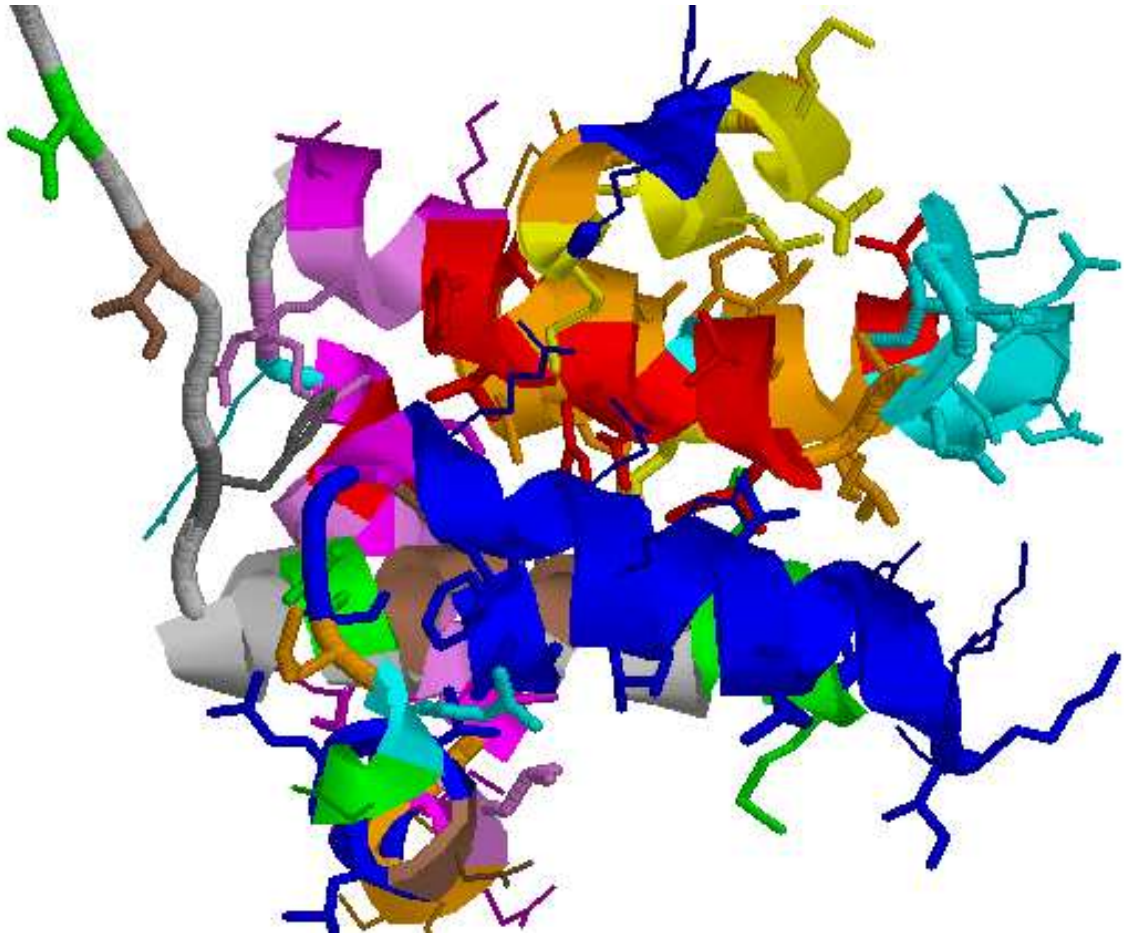


The Good and Difficult: T0353

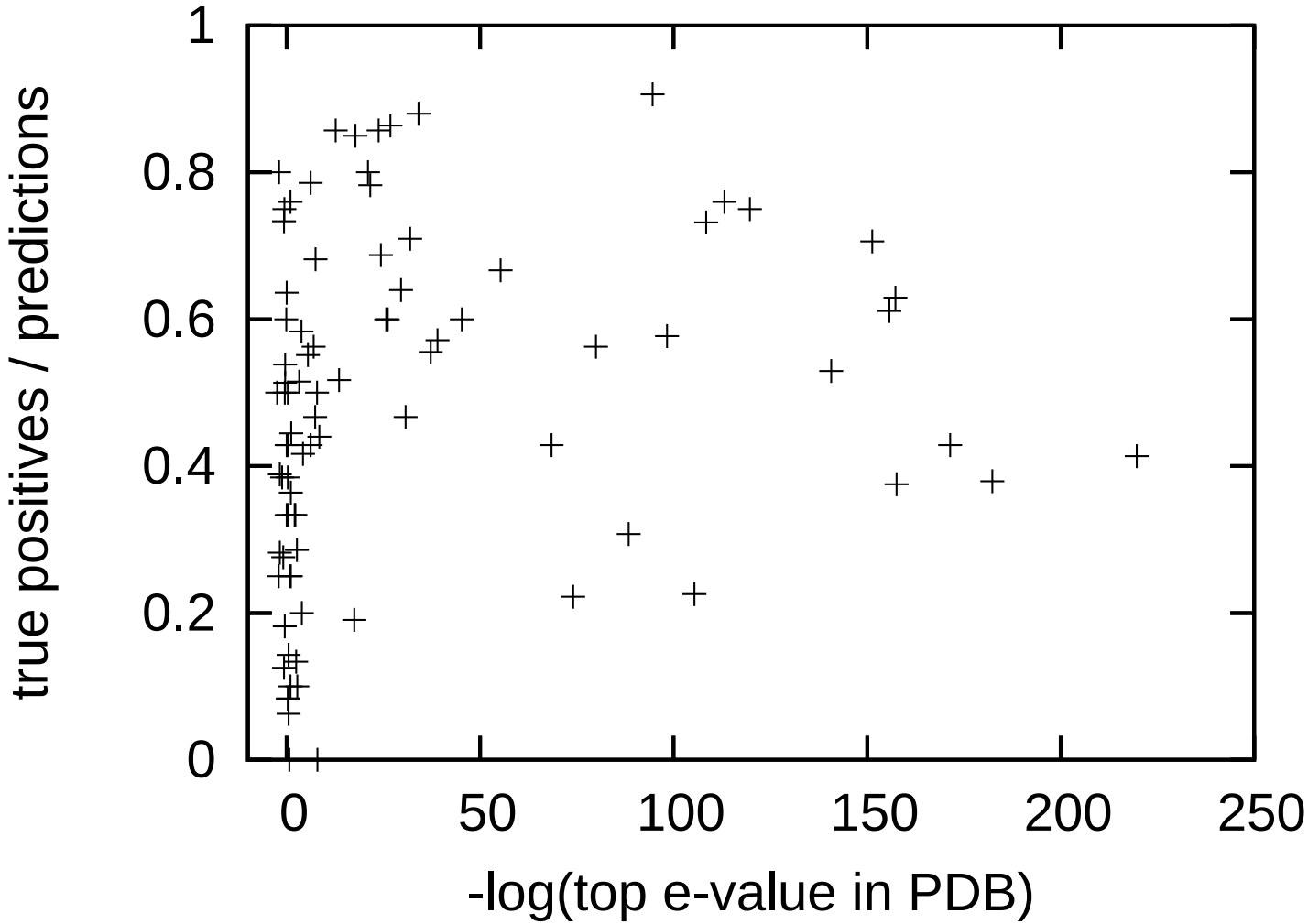
T0353 is from the free modeling class.



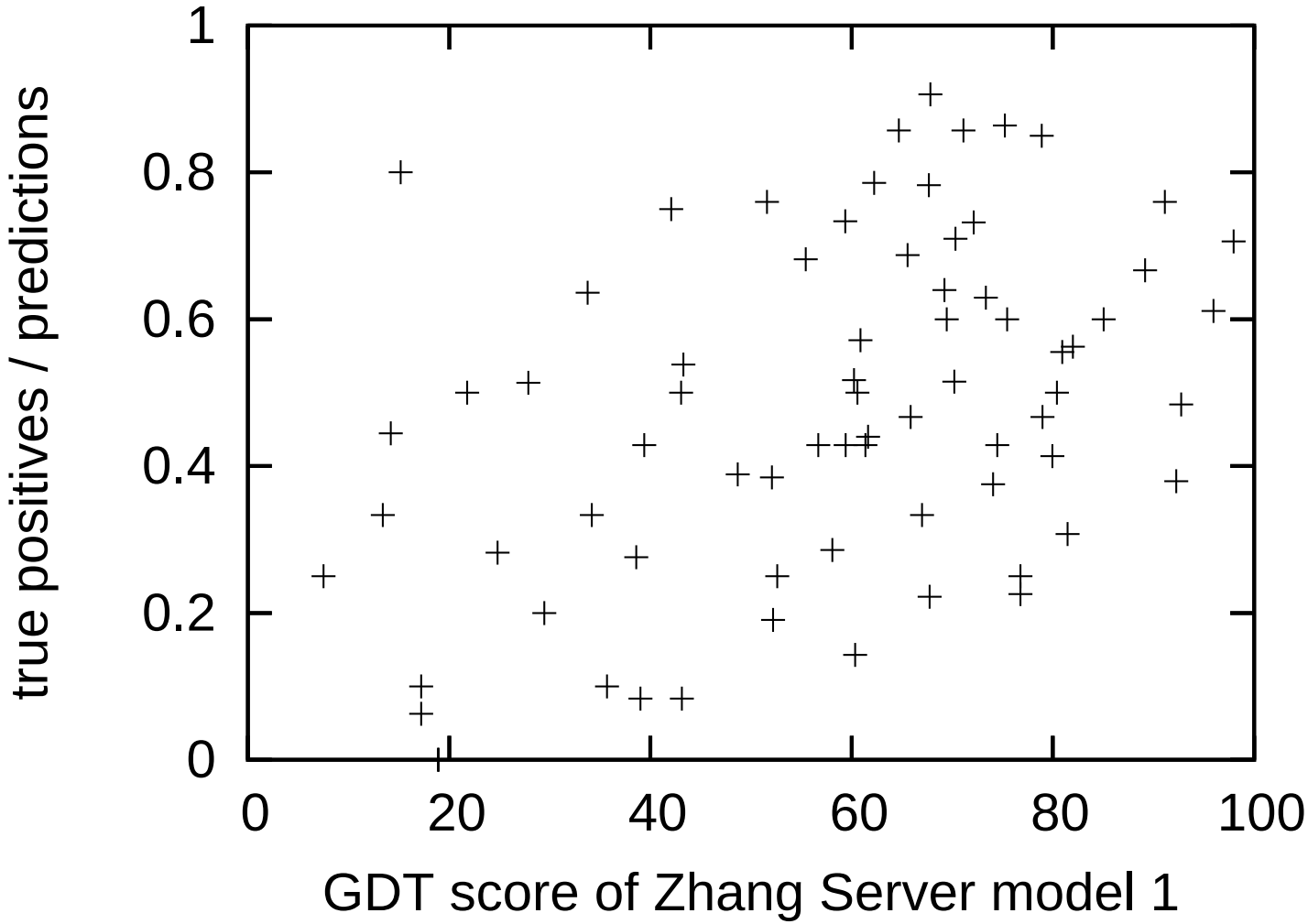
The Bad: T0307



Accuracy vs. Log BLAST E-value

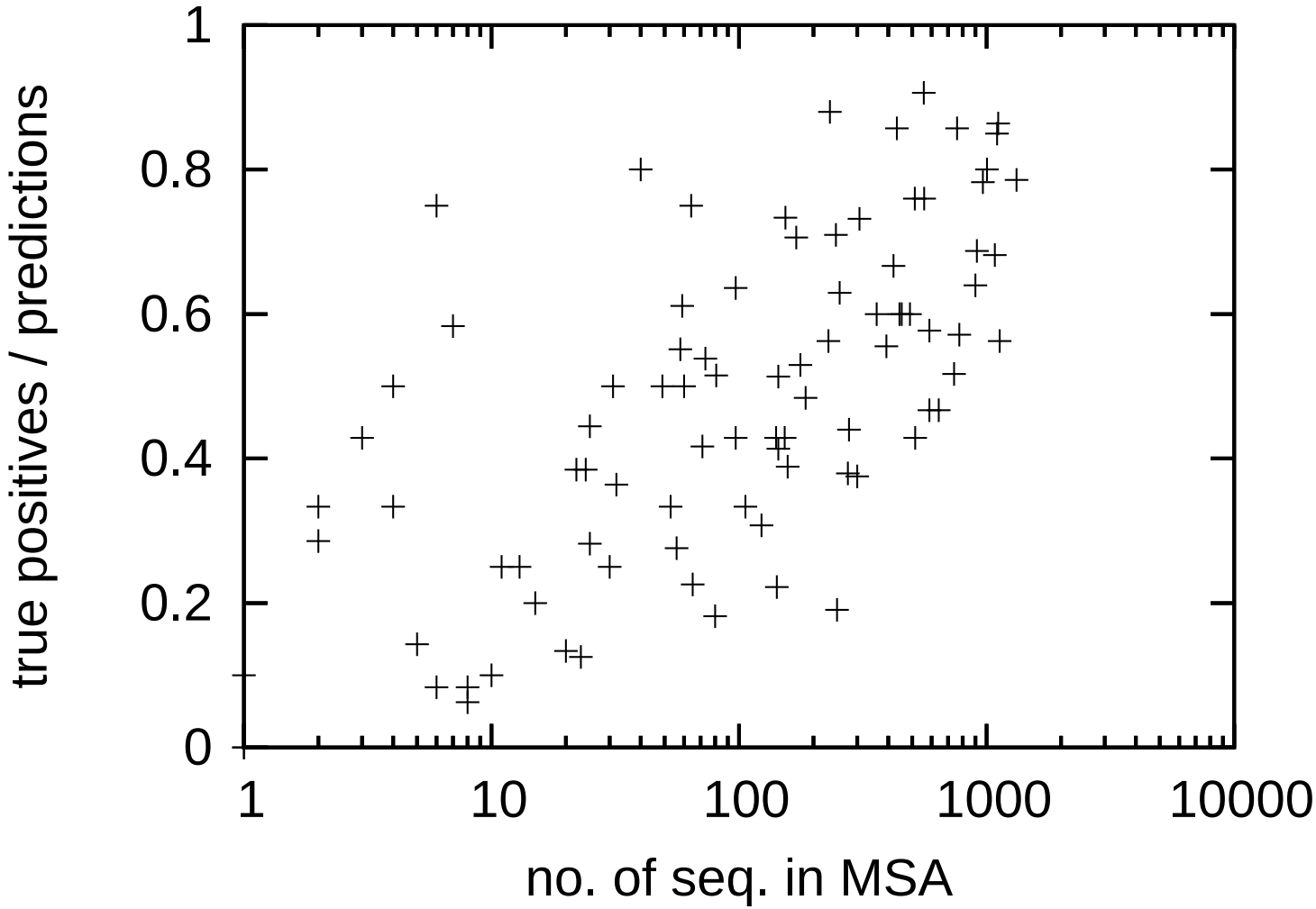


Accuracy vs. Zhang Server GDT

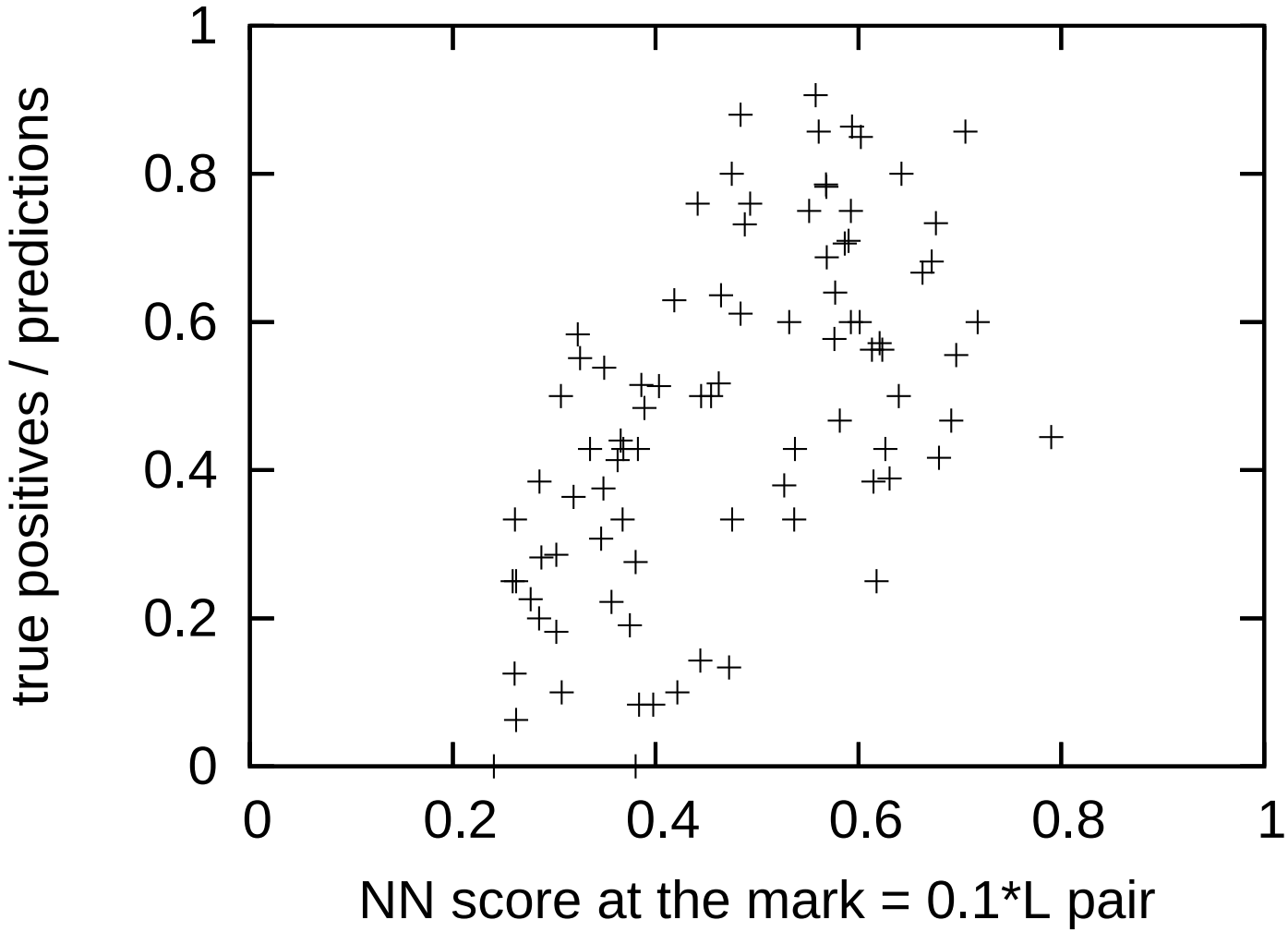


Pearson's r: 0.45

Accuracy vs. # Seq. in MSA



Accuracy vs. Neural Net Score



Pearson's r: 0.58

Conclusion and Future Work

Conclusions:

- Contact predictions go from very poor to very good.
- Contact predictions may sometimes be useful.
- Poor correlation between neural net score and accuracy.

Future work:

- Improve calibration of neural network score.
- Investigate separate predictor(s) for small alignments.
- Demonstrate usefulness of contact predictions.

Thanks and Acknowledgments

Kevin Karplus (adviser)
Richard Hughey (SAM software)
Rachel Karchin (Johns Hopkins)

Postdoc Martin Madera

Graduates Firas Khatib, Grant Thiltgen, Pinal Kanabar, Chris Wong, Zach Sanborn

Undergraduates Cynthia Hsu, Silvia Do, Navya Swetha Davuluri, Crissan Harris

Last but not least Anthony Fodor at Stanford for Java software

In memory of my father, John Cooper Shackelford