Model Quality Assessment
Guessing how good protein structure predictions are

Kevin Karplus, Martin Paluszewski, John Archie

karplus@soe.ucsc.edu

Biomolecular Engineering Department

Undergraduate and Graduate Director, Bioinformatics

University of California, Santa Cruz

Outline of Talk

- Protein structure prediction
- CASP and Metaservers
- What is Model Quality Assessment (MQA)?
- Types of MQA methods
- Contacts from alignments
- Optimizing multiple cost functions
- Results

What is a protein?

- There are many abstractions of a protein: a band on a gel, a string of letters, a mass spectrum, a set of 3D coordinates of atoms, a point in an interaction graph, ...
- For us, a protein is a long skinny molecule (like a string of letter beads) that folds up consistently into a particular intricate shape.
- The individual “beads” are amino acids, which have 6 atoms the same in each “bead” (the *backbone* atoms: N, H, CA, HA, C, O).
- The final shape is different for different proteins and is essential to the function.
- The protein shapes are important, but are expensive to determine experimentally.

Folding Problem

The *Folding Problem*:
If we are given a sequence of amino acids (the letters on a string of beads), can we predict how it folds up in 3-space?

```
MTMSRRNTDA ITIHSILDWI EDNLESPLSL EKVSERSGYS KWHLQRMFKK
ETGHSLGQYI RSRKMTEIAQ KLKESNEPIL YLAERYGFES QQTLTRTFKN
YFDVPPHKYR MTNMQGESRF LHPLNHYNS
```

↓

CASP ~~Competition~~ Experiment

- Everything published in literature “works”
- CASP set up as true blind test of prediction methods.
- Sequences of proteins about to be solved released to prediction community.
- Predictions registered with organizers.
- Experimental structures compared with solution by assessors.
- “Winners” get papers in *Proteins: Structure, Function, and Bioinformatics*.

Metaservers

- For the past several CASPs, some of the best predictions came from groups that did no prediction themselves.
- Instead, they looked at the results from several servers, and selected the model they thought was best (or made a new model by copying parts from different models or did some minor re-optimization of the models they thought were best).
- Servers that do this selection (and possible optimization) from the results of other servers are called *metaservers*.

What is Model Quality Assessment?

- A key step in making a metaserver is evaluating the models from the primary servers (or even other metaservers) and selecting the best one(s).
- Ranking or scoring the models without knowing the true structure is known as *Model Quality Assessment* (MQA).
- A good MQA method provides a high correlation between its score and some measure of the real quality (determined after the structure is known).
- CASP7 (2006) started evaluating MQA functions without requiring constructing metaservers.

Correlation

There are several notions of correlation we can use. The most popular are

- Pearson's r (which assumes a linear relationship)
- Spearman's ρ (which is Pearson's r on the ranks)
- Kendall's τ (which also depends only on ranks)

Since we have no reason to assume or require that the MQA score be linearly related to real quality, Spearman's ρ or Kendall's τ provides a better measure.

Weighted Kendall's τ

Define

$$\mathrm{W}_{\alpha,i} = e^{-\alpha i/(n-1)}$$

where i is the rank of the decoy by cost,
α is an arbitrary weighting parameter,
and n is the total number of models.

$$\tau_\alpha = 2\frac{\sum_i \mathrm{W}_{\alpha,i} \sum_{j \neq i} \mathrm{C}_{i,j}}{\sum_i \mathrm{W}_{\alpha,i}(n-1)} - 1$$

where $C_{i,j}$ is 1 if the model with better cost is superior,
0 if the model with better cost is inferior,
and 0.5 if the models are tied in either cost or quality.

Weighted Kendall's τ interpreted

- If α is zero, this measure is Kendall's $\tau = 2p - 1$, where p is probability that for a random pair the model with a better cost has the better quality.
- As $\alpha \to \infty$, $\tau_\alpha \to \ 2q - 1$, where q is the fraction of models of lower quality than the lowest-cost one.

Single-model measures

- Physicists like to use MQA functions that look only at single models, and address the question “how realistic is this model?”
- Single-model MQA measures are often made from physics-like energy functions or from empirical functions that try to capture how “protein-like” a model is.
- They have not worked well in metaservers or in CASP7 MQA evaluation.

Consensus methods

- A successful approach for metaservers (like Pcons) is to assume that several of the primary servers do a good job, but that each one can goof sometimes.
- If many servers agree on a similar model, then that model is more likely to be right than models proposed by only one or a few servers.
- One way to be right, many ways to be wrong.

Anonymous Consensus methods

- Consensus methods can know what models come from what servers and keep information about how much each server is to be trusted, or can treat all models as being equally trustworthy, discarding information about who created them.
- A simple anonymous consensus method is to measure the similarity of all pairs of models from the servers and to rank each model by its median similarity score to the rest of the models.
- Median GDT or median TM-score work very well—as well as the consensus methods that keep track of who created which model.

Similarity to good prediction

- One of the simplest MQA methods at CASP7 (by Lee's group) worked very well.
- They had good predictions of structures from their method, so they just measured similarity of models to their own prediction.
- This always rated their model as lowest cost (though it was rarely best).

Info from alignments

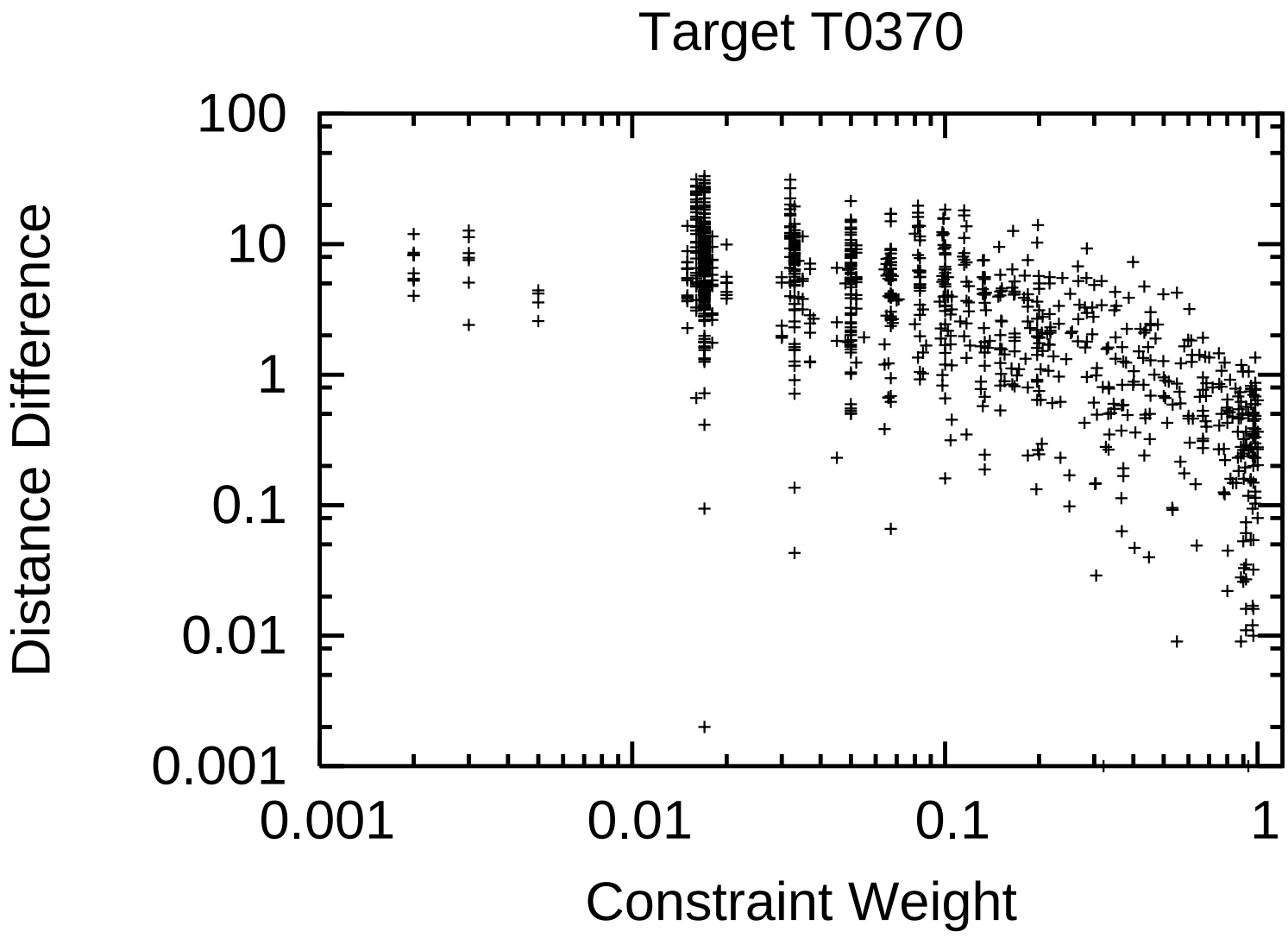
- We wanted an MQA method that took advantage of the prediction work we had done on a protein, but which did not just measure similarity to our favorite model.
- Our strength originally was in fold recognition: finding known structures that have detectable sequence similarity to our target and aligning them.
- So we decided to extract information from our top-scoring alignments to evaluate server models.

Contacts from alignments

- To reduce the information from the alignments to templates, we extract contact information: pairs of residues that align to residues that are close in a template.
- Find residue pairs whose C_β atoms are within

  8 Ångstroms in some template (ignoring pairs less than 9 apart along the backbone of the target).
- For each pair that has a contact in any alignment, compute desired distance as a weighted average of distances in alignments that has contacts for pair.
- Weights come from our confidence in the alignment.

How good are distances?



Constraint cost function

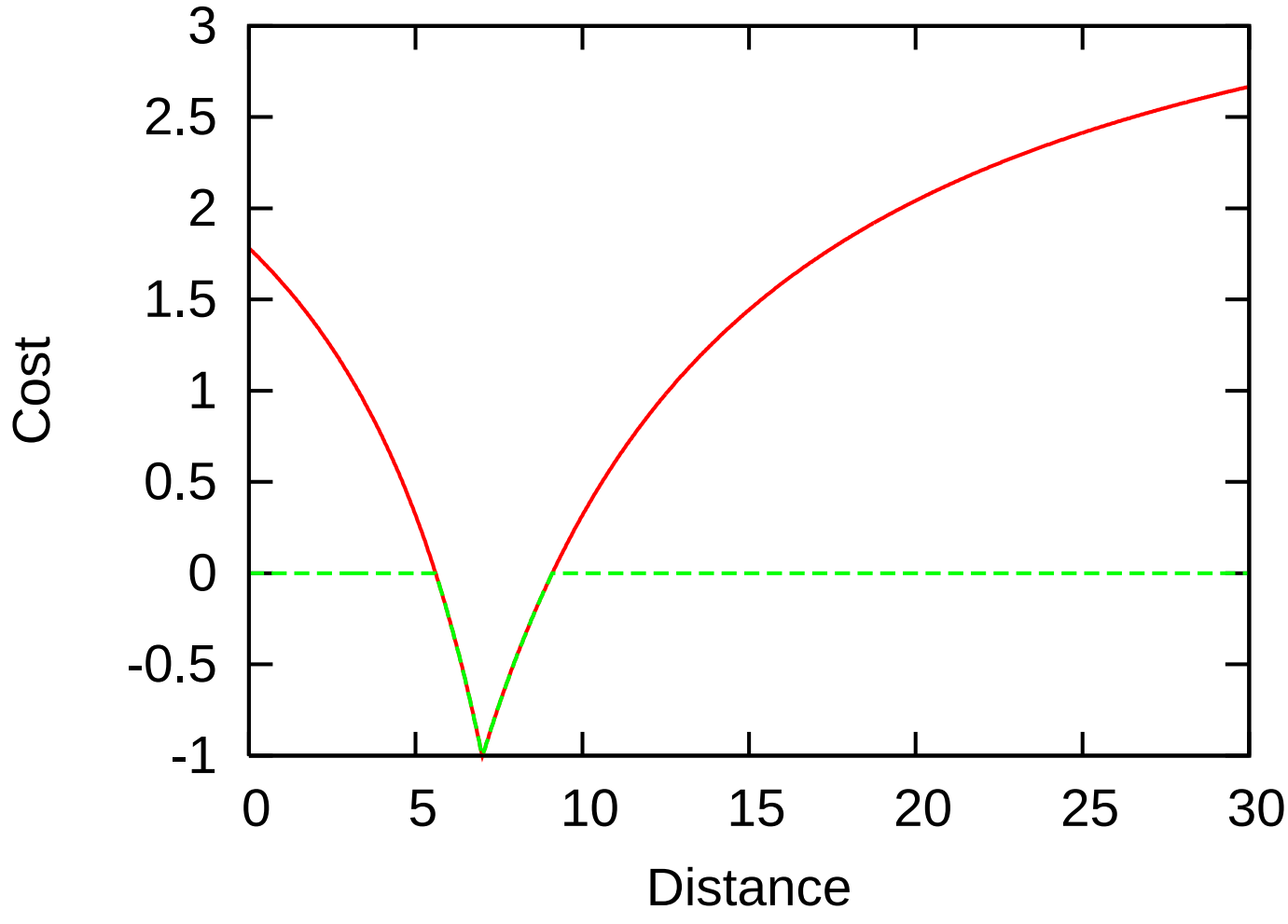
We made a cost function from desired distances:

$$
\begin{aligned}
C(\delta_{ij}) &= W_{ij} \frac{\alpha S_{ij}^2 + (1-\alpha) S_{ij} - 1}{\beta S_{ij}^2 + (\alpha - 1) S_{ij} + 1} \\
S_{ij} &= \frac{(\delta_{ij} - D_{ij})}{(L_{ij} - D_{ij})} \\
L_{ij} &= \begin{cases} 1.3 D_{ij} & \text{if } \delta_{ij} \geq D_{ij} \\ 0.8 D_{ij} & \text{otherwise} \end{cases}
\end{aligned}
$$

- minimum at desired dist: $C(D_{ij}) = -W_{ij}$
- $C'(D_{ij}) = 0$
- $C(0.8 D_{ij}) = C(1.3 D_{ij}) = 0$

Constraint cost function plot

$$D_{ij} = 7,\ \alpha = 200,\ \beta = 50,\ W_{ij} = 1$$



Optimizing contact set

- Not all predicted contacts are good.
- We often have too many contacts predicted.
- We can predict (with neural nets) how many contacts each residue should have (probability distribution of # contacts at each position).
- Thinning the list of contacts to maximize weights and probability of number of contacts improves predictions.

Optimizing multiple cost functions

- Have several cost functions from alignment constraints, secondary structure prediction, burial prediction, hydrogen bonds, ...
- Want a linear combination that maximizes correlation to real cost (say GDT similarity to true structure).
- If correlation were Pearson's r, this would be linear regression.
- Need method for Kendall's τ.
- Want all weights to be positive.
- Want most weights to be zero.

Greedy optimization

1. Compute correlation for each component separately. Pick component with max correlation.
2. Try adding each unused component to existing combination, optimizing weight of new component with simple search.
3. Add the component that increases correlation the most.
4. Re-weight components by doing a simple search on the weight of each component one at a time.
5. If correlation increases enough, repeat from step **??**

Results

Correlations on complete models with GDT_TS (average 5-fold cross-validation, 91 CASP7 targets):

Group	$\bar{r}$	$\bar{\rho}$	$\overline{\mathrm{GDT}}$	$\bar{\tau}_0$	$\bar{\tau}_3$
under+TM	**0.90**	**0.84**	**61.8**	**0.69**	**0.66**
under	*0.86*	*0.78*	*61.0*	*0.62*	*0.59*
Qiu	0.85	0.74	60.5	0.58	0.55
LEE	0.80	0.72	58.4	0.58	0.53
align	0.83	0.72	57.9	0.57	0.54
Pcons	0.85	0.74	58.0	0.56	0.51
TASSER	0.63	0.69	60.4	0.54	0.52
ModFOLD	0.70	0.62	57.0	0.46	0.44

Web sites

CASP8 working files: `http://www.soe.ucsc.edu/~karplus/casp8/`

List of my papers:

`http://www.soe.ucsc.edu/~karplus/papers/paper-list.html`

These slides: `http://www.soe.ucsc.edu/~karplus/papers/`

`MQA-talk-oct-08.pdf`

UCSC bioinformatics (research and degree programs) info:

`http://www.soe.ucsc.edu/research/compbio/`