PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Automatic Prediction of Protein 3D Structures by Probabilistic Multi-template Homology Modeling

Armin Meier[1,2], **Johannes Söding**[1,2]*

1 Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry, Göttingen, Germany, **2** Gene Center, Ludwig-Maximilians-Universität München Munich, Munich, Germany

* soeding@mpibpc.mpg.de

OPEN ACCESS

Citation: Meier A, Söding J (2015) Automatic Prediction of Protein 3D Structures by Probabilistic Multi-template Homology Modeling. PLoS Comput Biol 11(10): e1004343. doi:10.1371/journal.pcbi.1004343

Editor: Nir Ben-Tal, Tel Aviv University, ISRAEL

Received: February 23, 2015

Accepted: May 19, 2015

Published: October 23, 2015

Data Availability Statement: Data can be found in the supplemental files.

Funding: This work was funded by the German Federal Ministry of Education and Research (BMBF) within the framework of e:Med (grant e: AtheroSysMed, 01ZX1313A-2014), by the Deutsche Forschungsgemeinschaft (http://www.dfg.de/en/) grant numbers: GRK1721, SFB64 and by BioSysNet (http://www.biosysnet.de/) to JS. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Homology modeling predicts the 3D structure of a query protein based on the sequence alignment with one or more template proteins of known structure. Its great importance for biological research is owed to its speed, simplicity, reliability and wide applicability, covering more than half of the residues in protein sequence space. Although multiple templates have been shown to generally increase model quality over single templates, the information from multiple templates has so far been combined using empirically motivated, heuristic approaches.

We present here a rigorous statistical framework for multi-template homology modeling. First, we find that the query proteins' atomic distance restraints can be accurately described by two-component Gaussian mixtures. This insight allowed us to apply the standard laws of probability theory to combine restraints from multiple templates. Second, we derive theoretically optimal weights to correct for the redundancy among related templates. Third, a heuristic template selection strategy is proposed.

We improve the average GDT-HA model quality score by 11% over single template modeling and by 6.5% over a conventional multi-template approach on a set of 1000 query proteins. Robustness with respect to wrong constraints is likewise improved. We have integrated our multi-template modeling approach with the popular MODELLER homology modeling software in our free HHpred server http://toolkit.tuebingen.mpg.de/hhpred and also offer open source software for running MODELLER with the new restraints at https://bitbucket.org/soedinglab/hh-suite.

Author Summary

Since a protein's function is largely determined by its structure, predicting a protein's structure from its amino acid sequence can be very useful to understand its molecular functions and its role in biological pathways. By far the most widely used computational approach for protein structure prediction relies on detecting a homologous relationship with a protein of known structure and using this protein as a template to model the

structure of the query protein on it. The basic concepts of this homology modelling approach have not changed during the last 20 years. In this study we extend the probabilistic formulation of homology modelling to the consistent treatment of multiple templates. Our new theoretical approach allowed us to improve the quality of homology models by 11% over a baseline single-template approach and by 6.5% over a multi-template approach.

This is a *PLOS Computational Biology* Methods paper.

Introduction

Homology modeling is by far the most widely used computational approach to predict the 3D structures of proteins, and almost all protein structure prediction servers rely chiefly on homology modeling, as seen in the community-wide blind benchmark "Critical Assessment of Techniques for Protein Structure Prediction" (CASP) [1–3].

Homology modeling consists of four steps: (1) Finding homologous template proteins of known structure, (2) Selecting the best template or set of templates, (3) Optimizing the multiple sequence alignment (MSA) between query and template protein sequences, and (4) Building the homology model for the query sequence that resembles as closely as possible the structures of the templates, accommodating for deletions and insertions of query residues with respect to the template structures.

During the last 15 years, much progress has been made regarding the sequence-based steps 1 to 3. This is mainly owed to the development of more sensitive and accurate methods for sequence searching and alignment that compare sequence profiles or profile hidden Markov models (HMMs) with each other [4–6]. In contrast, improvements to the last step have been marginal. This is illustrated by the fact that, although a number of tools for protein homology modeling exist, to our knowledge all are older than 12 years (see [7, 8] for reviews). ModSeg/ENCAD [9] copies template coordinates and bridges gaps by short fragments that match the framework of the target structure. SWISS-MODEL [10] generates a core model by averaging template backbone atom positions. NEST [11] implements an artificial evolution algorithm where changes from the template structure such as substitutions, insertions and deletions are made one at a time, and each mutation is followed by an energy minimization. This process is repeated until the whole query is modeled.

These tools rely on of various heuristics. MODELLER [12], with 7500 citations clearly the most popular and according to two studies [7, 8] also the most successful homology modeling software to date, stands out by being based on a statistical approach to homology modeling. MODELLER is essentially unchanged at its core since its publication 22 years ago, while extensions such as refined energy functions [13] or loop modeling [14] have led to relatively minor improvements of its already excellent performance. We therefore believe MODELLER's success is owed to the consistent, statistical approach at its core.

MODELLER proceeds in two steps: (1) Derive from the MSA and template structures a list of restraints and (2) find the model structure that minimizes the restraint violations. Each restraint is a probability density function. The most important class of template-dependent restraints are the probability density functions for the spatial distances of pairs of atoms in the query protein.

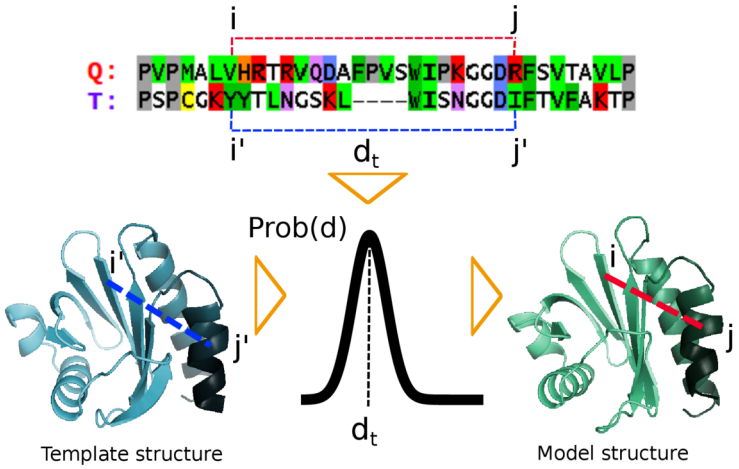

Fig 1. MODELLER's statistical approach to homology modeling: The unknown distance *d* between two atoms in residues *i* and *j* of the query protein (Q) is described by a probability distribution Prob(*d*) that is peaked around the distance d_t between the corresponding atoms in residues i' and j' of the template protein (T). This distribution Prob(*d*) is a probabilistic distance restraint for the distance *d*. To model a protein, tens to hundreds of thousands of such distance restraints between pairs of atoms in the query protein are derived. The product of all these restraint functions, which is called the likelihood function in statistics, quantifies how well a model structure satisfies all restraints at the same time. Therefore, the model structure that maximises the likelihood function represents the best solution.

doi:10.1371/journal.pcbi.1004343.g001

The true distance d will be distributed around the distance d_t of the equivalent atoms in the template structure, where equivalent residues are those that are aligned to each other (Fig 1). MODELLER assumes for simplicity a Gaussian distribution for d. Its mean equals d_t and its standard deviation is predicted based on the sequence similarity between query and template. The restraint minimization in the second step amounts to a maximum likelihood optimization, where the likelihood is approximated as the product over the density functions of the individual restraints. This factorisation of the likelihood assumes that the individual restraints represent information independent of each other, because in probability theory the joint probability of two random variables (X and Y) is the product of their probabilities, $p(X, Y) = p(X)\ p(Y)$, if and only if they are independent of each other. Although the assumption of independence of restraints sounds rather drastic, the approximation turned out to work well in practice.

To aggregate the information from several templates, however, MODELLER does not multiply the density functions of all restraints as probability theory would suggest. Instead, it relies on an empirical observation that the distribution of the target distance informed by multiple template distances is multi-modal. Thus, MODELLER reverts to a heuristic approach and computes an additive mixture of the density functions, each derived from an individual template, to restrain a single target distance based on multiple templates.

Here, we develop a rigorous statistical treatment of multiple template homology modeling. We first show that the distance distributions for $\log(d)$ are very well described by two-component Gaussian mixture distributions. In contrast to MODELLER's one-component densities, these two-component densities allow us to combine density functions by multiplication. Second, we derive an algorithm to compute weights that take the statistical dependence of the distance information from the templates into account. Third, we propose a heuristic scheme for template selection. We demonstrate that the new HHpred modeling pipeline and in particular the new constraints yield substantially improved model qualities.

Materials and Methods

Modeling distance restraints

Our approach to multi-template homology modeling is based on the statistical approach to homology modeling introduced by MODELLER. Our software computes improved spatial restraints and calls the MODELLER software, which then reads in the restraints and finds a structure that optimally satisfies these restraints. We briefly recall MODELLER's approach of homology modeling here.

MODELLER's maximum likelihood approach to homology modeling. MODELLER proceeds in two steps to compute a model structure for a query sequence that is aligned to a set of templates with known structures. In the first step, it generates a list of hundreds of thousands of restraints for the distance between pairs of atoms in the query, based on the distance of corresponding atoms in the templates. E.g. if residue i of the query q is aligned to residue i' of a template t and similarly j is aligned to j', then the distance d between the C_α atoms of residues i and j in q will be restrained to be similar to the known distance d_t between the C_α atoms of residues i' and j' in t (Fig 1). In statistics, a restraint is described as a probability density function $p(d)$, and in MODELLER this distance restraint is modelled by a Gaussian function with mean d_t. The standard deviation of the Gaussian describes the expected deviation of the distance d from d_t. Distance restraints are generated for each pair of residues (i, j) for which aligned residues i' and j' exist and for various combinations of atom types, for which equivalent atoms exist in the aligned template residues, e.g. $C_\alpha - C_\alpha$, N − O, $C_\alpha - C_\gamma$ etc.

In the second step, MODELLER uses stochastic optimisation to find the model structure for the query sequence that maximises the likelihood. The likelihood is the probability of the data, i.e. the alignment and template structures, given the model structure. When a single template is used for modeling, MODELLER approximates the likelihood as the product of the probability density functions over all restraints. Although this approximation corresponds to assuming the independence of all restraints, it has turned out to work well in practice.

Sali and Blundell [12] observed that the expected deviation $d - d_t$ depended on (1) the fraction of identically aligned residues between the two sequences, (2) the average solvent accessibility of the two aligned residue pairs (i, i') and (j, j'), (3) the average distance of i, i', j and j' from a gap, and (4) the distance d_t. They modelled the standard deviation of the Gaussian restraint as functions of the four discretized variables. To fit these functions, they analysed a large set of structurally aligned, homologous proteins for which they measured the distances $d = d_{ij}$ and $d_t = d_{i'j'}$ between equivalent atoms in two pairs of structurally aligned residues, (i, i') and (j, j'). Four different functions are trained, one for each of the following combinations of atom types: $C_\alpha - C_\alpha$, N − O, side chain—main chain, side chain—side chain.

New distance restraints that account for alignment errors. Because the analysis in [12] relied on structurally alignable residue pairs in structure-based alignments, they were basically free of alignment errors and therefore the distance in the query was always similar to the distance in the template. In practice, the sequence alignment will contain errors and i and i' (or j and j') might not be homologous to each other. In this case, d_t does not contain information about d and may be vastly different. When the pairs of residues (i, i') and (j, j') are sampled from real sequence alignments, this may lead to a stark deviation of the distance distribution from a Gaussian.

Fig 2(A)–2(C) shows distributions of $\log(d) - \log(d_t)$ for sets of residue pairs (i, i') and (j, j') sampled from alignments with successively lower quality. In Fig 2A only very reliable alignments have been sampled, with a posterior probability (pp) for (i, i') and (j, j') to be correctly aligned larger than 0.9 and with a sequence similarity (sim) above 0.75 bits per aligned pair. (See the Supporting Information for the definition of pp and sim.) Consequently, the empirical

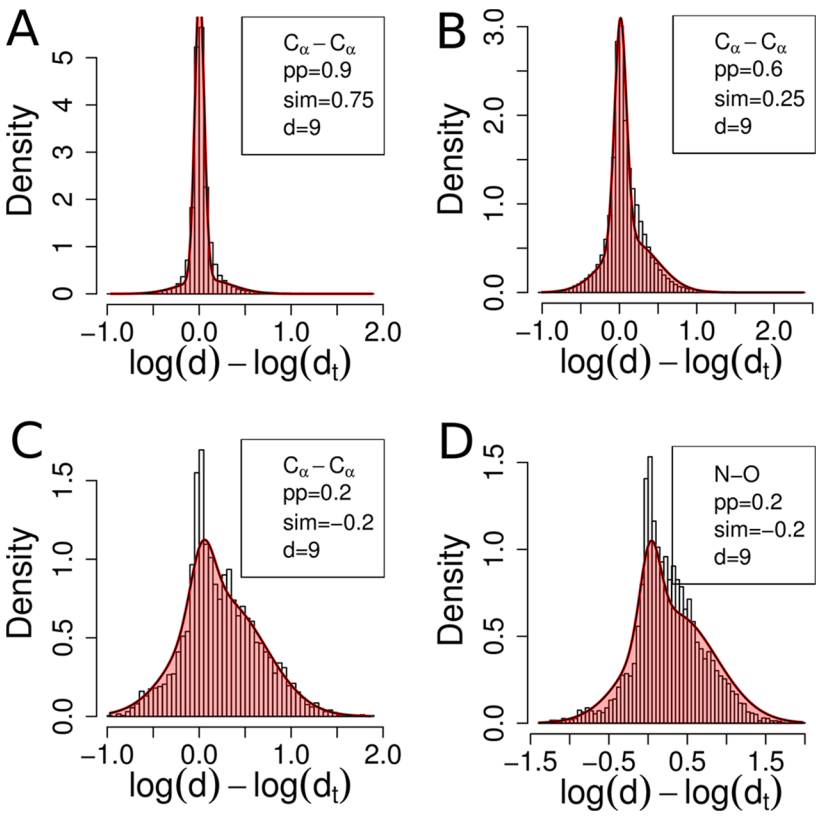

Fig 2. Empirical log distance distributions between pairs of atoms are well modelled by a two-component Gaussian mixture composed of a signal component and a background component. The background component originates from pairs of residues with an alignment error. The plots show the empirical distribution of $\log d - \log d_t = \log d_{ij} - \log d_{i'j'}$ for thousands of sampled pairs of residues (i, i'), (j, j') from real, error-containing pairwise sequence alignments generated with HHALIGN [15]. The two-component Gaussian mixture distribution predicted by the mixture density network in Fig 3B is plotted in red. From (**A**) to (**C**), the reliability of the alignments at (i, i') and (j, j') (as measured by pp and sim values) decreases. Consequently, the weight of the background component increases at the expense of the signal component. (**D**) Same as (C) but showing the distribution of N – O distances instead of C_α – C_α distances.

doi:10.1371/journal.pcbi.1004343.g002

density distribution over $\log(d) - \log(d_t)$ has a single peak and is well fitted by a single Gaussian. However, when the alignment quality deteriorates, as shown in Fig 2B and 2C, a second component in the distribution manifests itself. It stems from residues (i, i') and (j, j') for which either (i, i') or (j, j') or both are not homologous. These data points thus contribute a background distribution that does not depend on the distance d_t in the template.

These observations motivated us to model the restraint function $p(\log d|\ \log d_t, \text{pp}, \text{sim}) = p(\log d|\theta)$ using a two-component Gaussian mixture distribution (see Fig 3A) whose means, standard deviations and mixture weight w depend on $\theta = (\log d_t, \text{pp}, \text{sim})$ or $\theta' = (\text{pp}, \text{sim})$:

$$p(\log d|\theta) = \underbrace{w(\theta)\mathcal{N}(\log d|\mu(\theta), \sigma^2(\theta))}_{\text{correctly aligned}} + \underbrace{(1 - w(\theta))\mathcal{N}(\log d|\mu_{\text{bg}}(\theta'), \sigma^2_{\text{bg}}(\theta'))}_{\text{alignment errors}} \quad (1)$$

The mixture weight $w(\theta)$ can be regarded as the probability that both (i, i') and (j, j') are correctly aligned. Locally unreliable alignments will lead to a stronger background component and hence to softer distance restraints. Note that, because distances cannot be negative, they are not

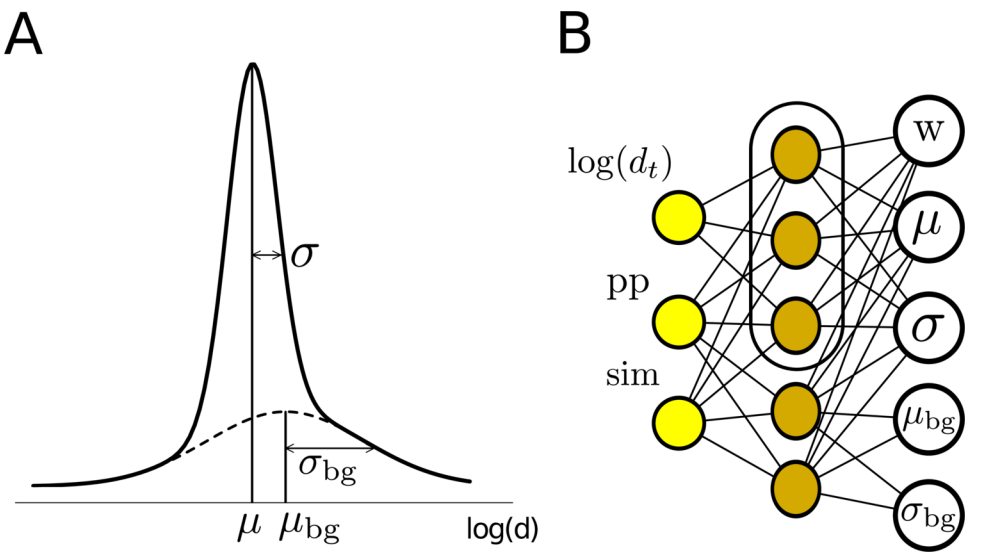

Fig 3. (**A**) Illustration of the two-component Gaussians mixture distribution in Eq (1). (**B**) Mixture density network to predict the parameters (w, μ, σ, μ_{bg}, σ_{bg}) of the Gaussian mixture distribution given the three variables θ = (log d_t, pp, sim) (d_t: distance in template, pp: posterior probability for both aligned residue pairs to be correctly aligned, sim: sequence similarity). Since the background component does not depend on d_t, the nodes for μ_{bg} and σ_{bg} are only connected to the two lowest hidden nodes that are not connected to log d_t.

doi:10.1371/journal.pcbi.1004343.g003

well modelled by Gaussian distributions, whose left tail can penetrate into the negative domain. We therefore modeled the distribution of log d instead of d.

Mixture density networks. To predict the five parameters of the Gaussians mixture distribution in Eq (1), we trained four mixture density networks [16], one for each combination of atom types listed above. A mixture density network is a special kind of neural network that learns the optimum adaptive functions for predicting the parameters of a Gaussian mixture distribution. It is trained by maximizing the likelihood of a set of training data that consists of the input features together with the value log d whose distribution should be learned. We used the R package `netlabR` to implement a mixture density network with five hidden nodes as illustrated in Fig 3(B). As input features we used θ = (log d_t, pp, sim). The local alignment quality pp(i, j) and the global BLOSUM62 sequence similarity sim are parsed from the output of HHALIGN in the hh-suite package [15], a widely used software for remote homology detection and sequence alignment (see Fig 8, green points). The set of three features was obtained by starting from a more redundant set of alignment features described in Table B in S1 Text and successively eliminating features whose omission did not significantly deteriorate the likelihood on the training set (in particular probability and raw score).

Combining restraints from multiple templates. When several templates cover residues i and j of the query, the restraints on the distance d of atoms in residues i and j from those templates have to be combined. Multiplying the restraint functions as probability theory would suggest (see below) would not work in MODELLER's case. When one of the restraints is wrong due to an alignment error, for instance, the restraint function of the incorrect restraint would severely distort the model structure, because the probability density of its single-component Gaussian falls off very fast for increasing distance from its mean, which effectively forbids any gross violation of the restraint. Therefore, MODELLER resorts to a heuristic to estimate the probability density $p(d|d_1, d_2)$ resulting from the restraints of two templates t_1, t_2 with corresponding distances d_1 and d_2: It adds both probability densities $p(d|d_1)$ and $p(d|d_2)$ (Fig 4A) using some

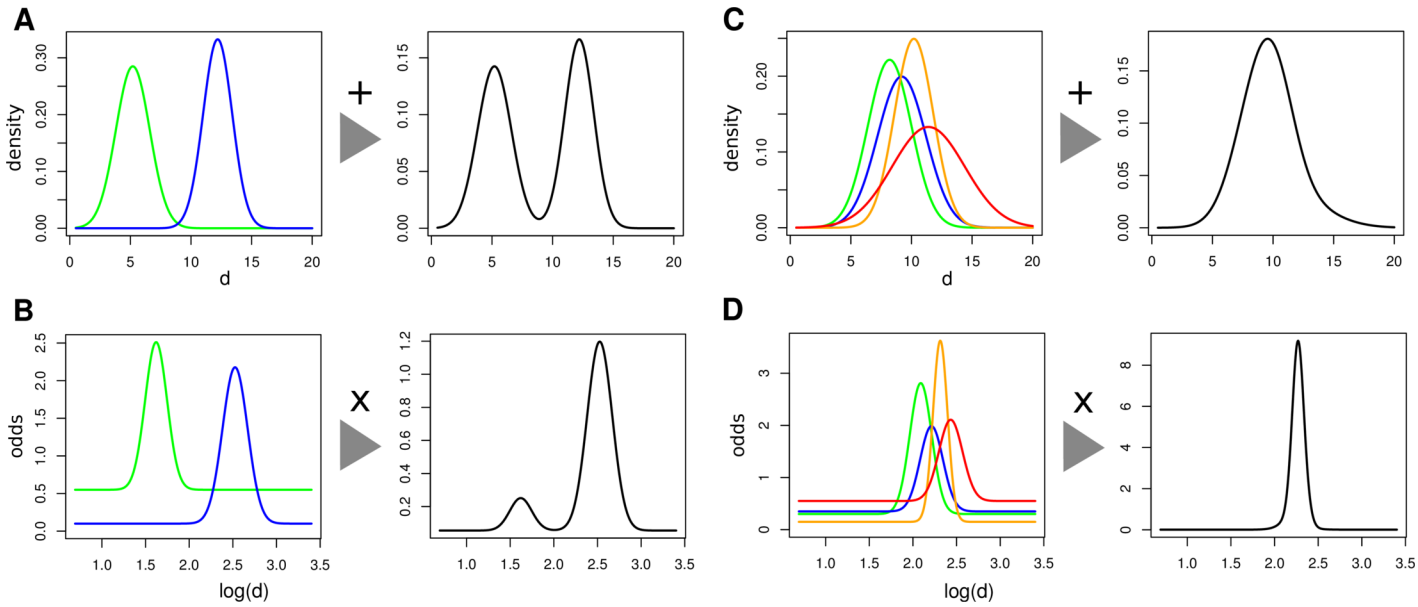

Fig 4. Comparison of how restraints from multiple templates are combined in MODELLER (top row) and in our new approach (bottom row). (A) In MODELLER, two restraints functions (green and blue) are additively mixed with mixing weights that have to be learned on a set of triples of aligned protein structures. **(B)** Our new restraints are multiplied instead of being added. The background component ensures that the restraint function becomes constant and the restraint thus becomes inactive (i.e. ignored) when the distance d is far from the distance in the template. **(C)** MODELLER's additive mixing leads to a total restraint function that is wider than any of the single-template restraints, not narrower as it should. **(D)** The multiplication of restraints functions according to probability theory leads to the desired behaviour of the total restraint function becoming more pointed with each restraint. Note that our new restraints are expressed as odds instead of densities (see also Eq 6).

doi:10.1371/journal.pcbi.1004343.g004

weights:

$$p(d|d_1, d_2, s_1, s_2) \approx \alpha(s_1)\, p(d|d_1) + \alpha(s_2)\, p(d|d_2)\,. \tag{2}$$

Here s_1 and s_2 measure the average sequence similarity in the sequence neighbourhoods around the two pairs of aligned residues from q and t_1 and from q and t_2, respectively. The optimum functions $\alpha(s_1)$, $\alpha(s_2)$ were found by training on a large number of structurally aligned triplets of proteins q, t_1, t_2 [12].

This heuristic approach leads to undesirable behaviour, as illustrated in Fig 4A and 4C. According to elementary statistical principles, a restraint function for a distance d based on restraints from multiple templates should contain more information and be more sharply resolved than any single-template restraint function. However, the additive mixture density restraint in Eq (2) is wider, not narrower, than any single restraint.

The new two-component distance restraints allow us to apply the rules of probability to combine the information from the two templates. By Bayes' theorem we obtain

$$p(d|d_1, d_2) = \frac{p(d_1, d_2|d)\, p(d)}{p(d_1, d_2)}\,. \tag{3}$$

If the information in the templates was approximately conditionally independent given d, i.e., $p(d_1, d_2|d) \approx p(d_1|d)\, p(d_2|d)$ we would obtain

$$\frac{p(d|d_1, d_2)}{p(d)} \approx \frac{p(d_1|d)}{p(d_1)}\,\frac{p(d_2|d)}{p(d_2)} = \frac{p(d|d_1)}{p(d)}\,\frac{p(d|d_2)}{p(d)}\,, \tag{4}$$

where Bayes' theorem was applied to each factor in the second step.

In practice, the query and templates are related to each other through evolution along phylogenetic trees, and conditional independence cannot be assumed. We therefore approximate the dependence among the templates by weighting their odds ratios, with weights $w_k \in [0, 1]$. This method is analogous to weighting sequences according to their similarity with other sequences in a multiple sequence alignment in order to compute a sequence profile [17] or some other family-dependent features [18]. We will derive a method to determine optimal template-specific weights w_k in the following subsection. The previous formula can then be generalised to K templates, giving

$$\frac{p(d|d_1,\ldots,d_K)}{p(d)} \approx \prod_{k=1}^{K} \left(\frac{p(d|d_k)}{p(d)}\right)^{w_k}. \quad (5)$$

Here, $p(d)$ is the probability independent of any template, i.e., the background distribution $\mathcal{N}(d|\mu_{\mathrm{bg}}, \sigma^2_{\mathrm{bg}})$. According to Eq (1), the restraint functions are now (for the sake of brevity we omit θ and θ')

$$\frac{p(d|d_k)}{p(d)} = \frac{(1-w)\mathcal{N}(\log d|\mu_{\mathrm{bg}}, \sigma^2_{\mathrm{bg}}) + w\mathcal{N}(\log d|\mu, \sigma^2)}{\mathcal{N}(\log d|\mu_{\mathrm{bg}}, \sigma^2_{\mathrm{bg}})} = 1 - w + w\frac{\mathcal{N}(\log d|\mu, \sigma^2)}{\mathcal{N}(\log d|\mu_{\mathrm{bg}}, \sigma^2_{\mathrm{bg}})}. \quad (6)$$

Note that the ratio of the two Gaussians is again a Gaussian, because subtracting two quadratic functions of d again yields a quadratic function. Fig 4B and 4D illustrate how restraints from multiple templates are combined under our new statistical approach and that this leads to the expected desirable behaviour of the total restraint restraining more strongly than the one-component restraints.

Dividing by the background has two effects: first, it prevents the background to become dominant when the individual background components of all $P(d|d_k)$ are multiplied. Second, the negative logarithm of Modeller's distance restraint is quadratic in d, and hence unsatisfiable restraints can lead to extreme values during optimization. Dividing by the background avoids this quadratic increase because $P(d|d_k)/P(d)$ has flat tails where it approaches a constant $(1 - w)$. In cases of incorrect alignments with a wrong distance d_t in the template, the restraint will not disrupt the query's model structure as d will be pulled away from d_t into the flat region of the restraint. Combining two component distance restraints as shown in Fig 4D thus reinforces consistent restraints while avoiding distortions from incorrect restraints.

Running Modeller with the new distance restraints. After having picked a set of templates, we run the Modeller (version 9.10) `automodel.homcsr(0)` command that generates a file with the list of restraints from the query-template alignment. We parse the list of restraints and replace each template-dependent distance constraint (which is either a Gaussian function for a single-template restraint or a Gaussian mixture for a multi-template restraint) with a set of our own distance restraints, one for each template. For this purpose, we added a restraint function that computes the logarithm of Eq (6) to Modeller. All template-independent restraints such as main chain and side chain dihedral angle restraints, bond lengths etc. are left unchanged. We run Modeller with the modified restraints list to generate a 3D model.

Template weighting

Motivation. As a motivation for the template weighting scheme, consider the case shown in Fig 5A. Giving all three templates the same weight ignores the dependencies described by the tree [18]. Template t_3 should get a weight of 1, since conditioned on q it is independent of the other two templates. But templates t_1 and t_2 should get weights clearly smaller than 1, since they do not contribute independent information to d. On the other hand, they are not identical

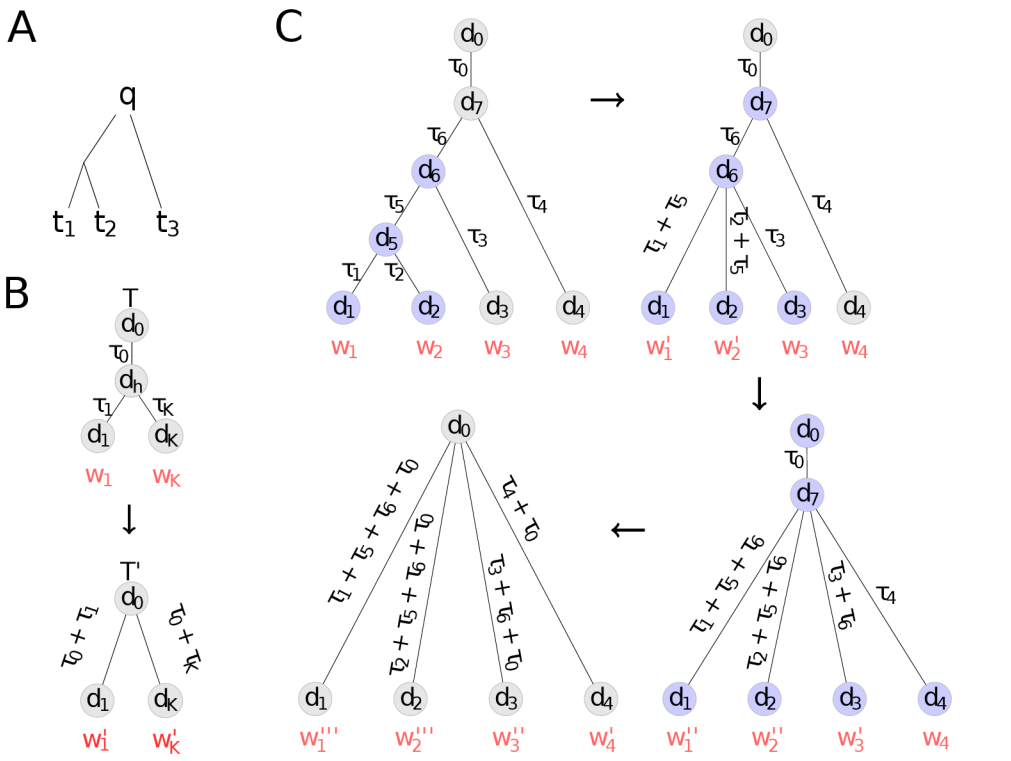

Fig 5. Iterative scheme for computing weights for templates by transforming the phylogenetic tree connecting them and the query protein into an equivalent tree with star-like topology with the query in the center. (A) Templates t_1 and t_2 are closely related and should be down-weighted with respect to t_3. **(B)** Any tree $\mathcal{T}$ with a structure at an internal node with unknown distance d_h to which all templates are connected in a star-like topology (top) can be transformed into an equivalent tree $\mathcal{T}'$ (bottom) with star-like topology, where equivalence means that the restraint on the distance d_0 of the top node is the same for both trees. τ_1, ... τ_K indicate evolutionary distances. **(C)** Iterative restructuring of a phylogenetic tree. In each step, the basic transformation from Fig 5B is applied to the subtree colored in blue. Weights and edge lengths get updated until all templates are directly connected to the query.

doi:10.1371/journal.pcbi.1004343.g005

and hence should receive a weight clearly larger than 0.5. But how do we compute the exact optimum weights w_k for templates 1, . . ., K given a phylogenetic tree with known edge lengths?

Iterative restructuring. We begin by rooting the phylogenetic tree at the query, and giving its leaf nodes initial weights of 1. By iteratively applying the elementary step in Fig 5B to subtrees, we can transform a tree with arbitrary topology into a tree with a star-like topology, as shown in Fig 5C. At each step, one inner node is removed and the procedure continues until all template leaves are directly connected to the query. At each step, we simply need to update the template weights to obtain the final weights w_k for the star-like tree. In the star-like tree which we finally obtain, all template distances d_k are conditionally independent, and hence we obtain for the odds ratio the result in Eq 5, using the final weights w_k from this iterative process.

Elementary step. For the elementary step, we will show that the upper (sub)tree $\mathcal{T}$ in Fig 5B yields exactly the same odds ratio for d_0 as the transformed, star-topology tree $\mathcal{T}'$ below,

$$\frac{p(d_0|d_1 \dots d_K, w_1 \dots, w_K, \mathcal{T})}{p(d_0)} = \frac{p(d_0|d_1 \dots d_K, w'_1 \dots, w'_K, \mathcal{T}')}{p(d_0)}, \tag{7}$$

if the new weights w'_k are chosen according to

$$w'_k = \frac{1/\tau_0 + 1/\tau_k}{1/\tau_0 + \sum_{l=1}^{K} w_l/\tau_l}\, w_k. \tag{8}$$

The updated weights are proportional to the old w_k with a proportionality factor approaching 1 for $\tau_0 \ll \tau_k$. The sum of weights over all K templates is $(\sum_{k=1}^{K} w_k/\tau_0 + \sum_{k=1}^{K} w_k/\tau_k)/(1/\tau_0 + \sum_{k=1}^{K} w_k/\tau_k)$, which goes to 1 for $\tau_0 \gg \max\{\tau_k\}$, signifying that in this case the information in the templates is completely redundant.

To show that the odds ratio in Eq (7) is conserved when transforming the tree $\mathcal{T}$ into $\mathcal{T}'$ in Fig 5B, we integrate over the unknown, hidden distance d_h,

$$p(d_0|d_1 \ldots d_K, w_1 \ldots, w_K, \mathcal{T}) = \int p(d_0|d_h, w_0) p(d_h|d1 \ldots d_K, \mathcal{T})\, d(d_h)\,, \tag{9}$$

and apply Eq (5) to the second term in the integrand,

$$p(d_0|d_1 \ldots d_K, w_1 \ldots w_K, \mathcal{T}) = \int p(d_0|d_h, w_0) \prod_{k=1}^{K} \left(\frac{p(d_h|d_k, \tau_k)}{p(d_h)}\right)^{w_k} d(d_h). \tag{10}$$

We now make the very reasonable assumption that the evolution of the distance between pairs of atoms manifests diffusive behaviour. This behaviour results if the change in distance can be modelled by many small, independent changes, each change being the consequence of a sequence mutation that will slightly change the protein structure. Concretely, this means the probability of observing a distance d_l after an evolutionary time τ_{kl}, when in the ancestor the distance was d_k, is given by

$$p(d_l|d_k, \tau_{kl}) = \mathcal{N}(d_l|d_k, \gamma\tau_{kl}) \tag{11}$$

with some rate constant γ. Note that at time $\tau_{kl} = 0$ the standard deviation vanishes and the right hand-side becomes equal to the delta functional, as it should. Substituting the conditional probabilities in the integral with these expressions, we see that the integral is over a product of Gaussians and can be solved analytically by the method of completing the square (see Suppl. Material). This results in a Gaussian distribution which is shown in the Supporting Information to be equivalent to the tree $\mathcal{T}'$ with transformed weights w'_k given by Eq (8).

For simplicity, we use the UPGMA algorithm [19] to construct the initial tree $\mathcal{T}$. The distances are computed as $\mathrm{dist}(t_k, t_l) = -\log(\mathrm{TMscore}_{\mathrm{pred}}(t_k, t_l))$, where $\mathrm{TMscore}_{\mathrm{pred}}$ is the TMscore [20] predicted by a neural network similar to the one in the next subsection (Supplemental Fig. S1), but without the experimental resolution as input feature. The tree constructed in this way is subsequently rearranged so that the query q is at its root.

Note that by its construction the final tree with star-like topology has the same edge lengths between the query and any template as the real tree. This is important, since the restraint function for template t_k from the mixture density network depends on the similarity between q and t_k. In order for the new star-like tree to be equivalent to the real one, it has to represent the same pairwise $q - t_k$ similarities as the real tree.

Template selection

Single template selection. HHsearch ranks templates by the probability P_{hom} for the template to be homologous to the query protein. To pick the template best-suited for homology modeling, we trained a simple neural network with three hidden nodes (Supplemental Fig. S1) on the training set (see Results). The network predicts the TMscore [20] of the model built

with the query-template alignment, given various alignment features described in Table B in S1 Text. The idea is similar to [21], who proposed a neural network (NN) for picking the first template. We tried several feature combinations and, similar to previous work described in [22], found that the following features yielded the best results: HHSEARCH raw score, secondary structure similarity score divided by query length, expected number of correctly aligned target residues divided by query length, resolution of template structure in Angstroms. For each query, we picked the protein with highest predicted TMSCORE among all proteins found by HHSEARCH as the first template.

Multiple template selection. Picking the right set of templates for homology modeling is a difficult problem. The main beneficial effect of adding more templates is to increase the number of residues for which distance restraints can be generated [7]. However, picking too many templates can decrease the model quality because, as we discussed in the context of how MODELLER's restraints work, even a single bad template that gives rise to wrong distance restraints can severely distort the resulting 3D model.

To our knowledge, no theoretically well founded strategy for multi-template protein homology modeling has been developed so far, which contrasts with its widespread use in virtually every successful prediction pipeline. Contrary to single template selection, picking further templates is fundamentally complicated by complex dependencies between all selected structures. Current methods are therefore based on heuristics [23–25]. Some methods [26, 27] build a set of models based on several different template lists and then post-select a final model according to some quality measure [28].

As a simple baseline approach to multiple template selection, we employ the network of the previous section to select the first template. Further templates are added if 1) their predicted TMSCORE is at least 90% of the first template, 2) they are structurally similar to the first template (TMALIGN score > 0.7) and 3) all selected templates are structurally similar to each other (pairwise TMALIGN score > 0.8).

Next, we propose here a heuristic method which aims to optimise the trade-off between increasing the query sequence coverage and decreasing the restraint quality of already covered residues due to adding more diverged templates with less reliable alignments.

We select the set of templates from among the top 100 found by HHSEARCH in the following way (Fig 6). The first template t_1 is selected by the neural network that predicts the TMSCORE. For each template in the template list $\mathcal{L}$ (lower dashed box in the figure) a score $S(t)$ in (see Eq 14) is (re)calculated that rewards a high coverage while penalising the addition of templates whose alignment quality is worse than that of already selected templates. The template with highest score (t_4 in Fig 6) is added to the selected set if its score is still positive. The process is iterated until no template is left in $\mathcal{L}$ that has a positive score.

To calculate the score $S(t)$ (see Eq 14 below), we first define the local quality score,

$$s(i,t) = P_{hom}(t)\, p(i \diamond i' | q, t), \tag{12}$$

which is simply the product of the probability P_{hom} that template t is homologous to q times the probability $p(i \diamond i' | q, t)$ that residue i from q is homologous (i.e. correctly aligned) to residue i' in t. The latter probability is estimated by HHALIGN and HHSEARCH by a Forward-Backward algorithm. The local improvement (or impairment) of $s(i, t)$ with respect to the best local score $s(i, t')$ among already selected templates $t' \in \mathcal{T}_{\text{acc}}$ is

$$\Delta s(i,t) = s(i,t) - \max_{t' \in \mathcal{T}_{\text{acc}}} \{s(i,t')\}. \tag{13}$$

To weight the positive values more strongly than the negative ones, we apply the exponential function to $\Delta s(i, t)$, subtract a per-residue threshold β and sum over all aligned residue pairs

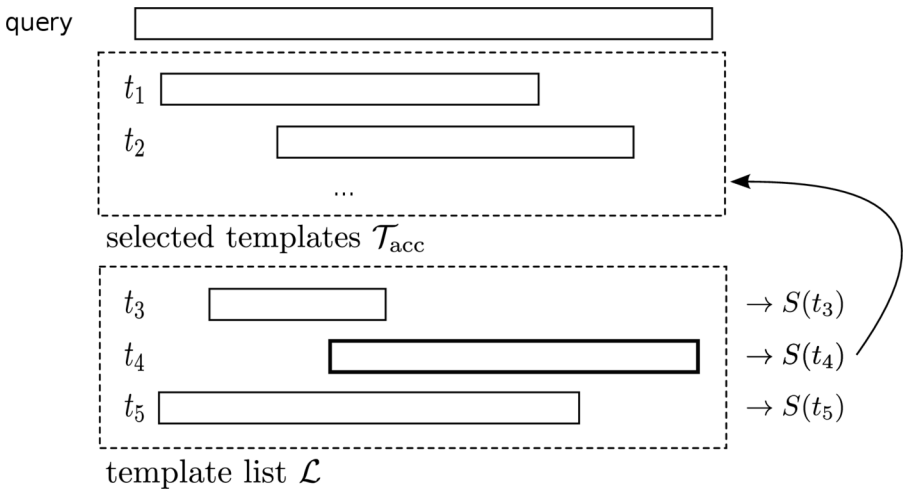

Fig 6. Selection of multiple templates. $\mathcal{T}_{acc}$ is the set of accepted templates, $\mathcal{L}$ is the set of template candidates. For each template in $\mathcal{L}$, its score is calculated according to Eq (14) and the template with the highest score (t_4) is added to $\mathcal{T}_{acc}$. This process is iterated until there is no more template with a positive score, or $\mathcal{T}_{acc}$ contains more than 8 templates.

doi:10.1371/journal.pcbi.1004343.g006

(i, i') in the alignment $A(q, t)$ between q and t:

$$S(t) = \sum_{(i,i')\in A(q,t)} [\, e^{\alpha \Delta s(i,t)} - \beta \,]. \tag{14}$$

The parameters α and β influence the degree of non-linearity and greediness of the selection, respectively. They were optimised with a simple grid search on the optimisation set as explained in the Results section.

Results

Benchmark sets

We filtered the sequences from the PDB database of protein structures (May 2010) down to 20% and 70% maximum pairwise sequence identity and a minimal pairwise E-Value of 0.1 (using scripts `pdb2fasta.pl` and `pdbfilter.pl` in the HHsuite package v2.0.16). For all sequences in the resulting pdb20 and pdb70 databases, we built multiple sequence alignments (MSAs) with our sensitive, iterative sequence search tool HHblits (v2.0.16) that is based on the pairwise alignment of profile hidden Markov models (Hmms) [15]. We used standard HHblits parameters with three search iterations against the uniprot20 database to get sufficiently diverse MSAs that are well suited to detect even remotely homologous proteins. The query sequences were picked from among the pdb20, and the template database was obtained from the pdb70 as explained below.

We extracted three disjoint query sets from the pdb20, a test, a training and an optimization set, with 1000, 1000, and 500 proteins, respectively. To achieve a good balance of easier and more challenging queries for modeling, we aimed to obtain the same distribution of query-template sequence identities as for the 108 queries in the CASP7 experiment shown in Supplemental Fig. S2 (which is similar to the distribution in CASP11, see Fig. S2). We computed the total amount of queries needed in each sequence identity bin (0%–5%, 5%–10%, . . ., 95%–100%). We then randomly picked query sequences from the pdb20 without replacement. For each picked query, we searched for possible templates in the pdb70 database and found the template

most structurally similar to q according to TMALIGN (excluding the query itself) and recorded the sequence identity given by TMALIGN. q was then put into one of the three sets if the sequence identity bin for that set was not yet filled up. Otherwise, q was rejected. Finally, for each of the three query sets we constructed a template set by removing the sequences in the query set from the pdb70.

We then searched with each query sequence q in one of the three sets through the corresponding template database using HHSEARCH, a slower and slightly more sensitive version of HHBLITS, resulting in a list tlist(q) of potential templates.

Distance restraints

Mixture density network. As training data for the mixture density networks for two-component distance restraints, we used the 3D models generated with single templates picked by the neural network. The alignment features for the network were again parsed from the HHSEARCH results. We fitted distributions of $\log(d)$ with the mixture of two Gaussians. MODELLER includes four different classes of distances depending on the atom types involved: between two C_α atoms (C_α–C_α), N-O atoms, side chain—main chain and side chain—side chain. We generated four sets of training data with 3 million training cases for C_α–C_α and N–O pairs, 1 million for SC–MC and 300k for SC–SC. Optimizing the log-likelihood of the mixture density network was done by conjugate gradient ascent until convergence was reached. Bad local optima were avoided by picking the run with maximum likelihood from among 50 random initializations.

Two-component distance restraints. We replaced all of MODELLER's template based distance restraints with our new two-component Gaussian mixture restraints. The optimization schedule was kept unchanged. The new restraints improved single-template modeling by 0.8% from a GDT-HA model quality score (GDT-HA is a high accuracy version of GDT-TS, see [29]) of 0.447 to 0.450, even though they were developed with multiple template modeling in mind. We then investigated the influence of replacing the new restraints when using our new multi-template selection strategy. We obtained an improvement of the average GDT-HA score over the 1000 queries in the test set by 2.5%, from 0.480 to 0.492 (Table 1 and Fig 7A), which is highly significant according to a paired t-test (P-value: $< 2.2 \times 10^{-16}$).

Template selection

Single template neural network. We selected the first template based on the query-template alignment features produced by HHSEARCH using the neural network with three hidden nodes shown in Fig. S1 (see Methods). To train the network, we built 3D models with MODELLER (version 9.10) for each query in the training set and each of the maximal 10 best-ranked HHSEARCH hits in tlist(q) as templates. This yielded 9212 models (since some queries had less than 10 database matches), whose model quality we evaluated using TMSCORE. To learn the network parameters we ran a standard back-propagation procedure. In order to avoid local optima, training was started from several random initializations, which all turned out to optimise to a similar likelihood on the training-set. The correlation between the network predictions and the true TMSCORE values was 0.89. Compared to selecting the first hit in the HHSEARCH results list for single template modeling, the neural network-based template selection led to a 0.9% increase of the average GDT-HA from 0.443 to 0.447 (Table 1).

Multiple template selection. Choosing multiple templates increases both the coverage and the probability to detect a correct template. However, a higher number of templates leads to accumulation of noise and wrong templates which decreases the model quality. As described in the Methods section, our template selection heuristic has two parameters, α and β. They

Table 1. Average model quality scores for different variations of template selection strategies and restraints used with MODELLER on a test set of 1000 single- and multi-domain proteins in the pdb20 database. The GDC-all score is similar to GDT-HA but also includes side-chain atoms in its assessment. Percent improvements are with respect to the first line. P-values are calculated based on a paired t-test with respect to the GDT-HA score in the previous line.

Method				GDT-HA	P-value	GDC-all
Name	Templates	Template selection	Restraints			
s.1st.old	Single	first hit	MODELLER	0.443 (+0%)	-	51.9 (+0%)
s.NN.old	Single	neural network	MODELLER	0.447 (+0.9%)	1.5E-6	52.4 (+1.0%)
s.NN.new	Single	neural network	new	0.450 (+1.5%)	8E-6	52.8 (+1.7%)
m.ss.old	Multiple	simple selection	MODELLER	0.462 (+4.3%)	1E-10	53.5 (+3.1%)
m.mt.old	Multiple	new multi-template	MODELLER	0.480 (+8.4%)	2E-16	55.1 (+6.2%)
m.mt.new	Multiple	new multi-template	new	0.492 (+11.1%)	2E-16	56.3 (+8.5%)

doi:10.1371/journal.pcbi.1004343.t001

were optimized on a grid $(\alpha, \beta) \in \{0.9, 0.95, 1, 1.05, 1.1\} \otimes \{0.8, 0.9, 1, 1.1, 1.2\}$ using all sequences in the optimization set as queries. For each parameter combination, templates were selected according to the score in formula (14). The alignment features between the query and all templates then served as input for MODELLER and 3*D* structures were generated. We found $\alpha = 0.95$ and $\beta = 1$ to maximize the cumulative TMSCORE of all models.

The new multi-template selection strategy picked on average 4.6 templates per query (Supplemental Fig. S3), which resulted in a mean coverage of 94% of the query residues (i.e. 94% of query residues where aligned to at least one template residue). The new selection strategy leads to an improvement of 3.9% from an average GDT-HA score of 0.462 to 0.480 (Fig 7B) compared to the following baseline selection strategy: We sorted all templates with respect to their predicted TMSCORE given by the single template neural network. The first template in this list is always selected and up to 10 templates along this ranking are chosen as long as their predicted TMSCORE is at least 90% of the very first one (Table 1).

Compared to the single template modeling approach, the improvement of multiple-template modeling without any further refinements (using the simple selection strategy and MODELLER restraints) was 4.3%, from average GDT-HA 0.443 to 0.462 (Table 1 and Fig 7C). The total improvement from the baseline, single template modelling to the most refined multi-template modelling strategy sums up to 11.1% (Fig 7D, fist and last line in Table 1).

Note that the improvement in modelling quality of multiple vs. single template modelling does not show a dependence on GDT-HA scores or sequence identities. In other words, difficult targets profit to the same degree as simpler targets from using multiple templates. This is consistent with the observation that both for single- and multi-domain targets, the average number of selected templates was similar across the entire range of sequence identities tested, from 0% to 80% (Supplemental Fig. S3).

Evaluation on cores

Most model quality assessment scores, such as the GDT-HA, do not penalize incorrect regions and thus reward adding more templates to increase the fraction of the query structure for which restraints can be derived. [30] assessed the effect of using a single or multiple templates on model quality and concluded that most of the gains are due to increased coverage of query residues by template residues. We wanted to discriminate between improvements in model quality due simply to increased coverage and improvements owed to reducing statistical noise by increasing the number of distance restraints on "core residues", conveniently defined here as residues covered by the alignment to the first, top-ranked template. We remove all non-core

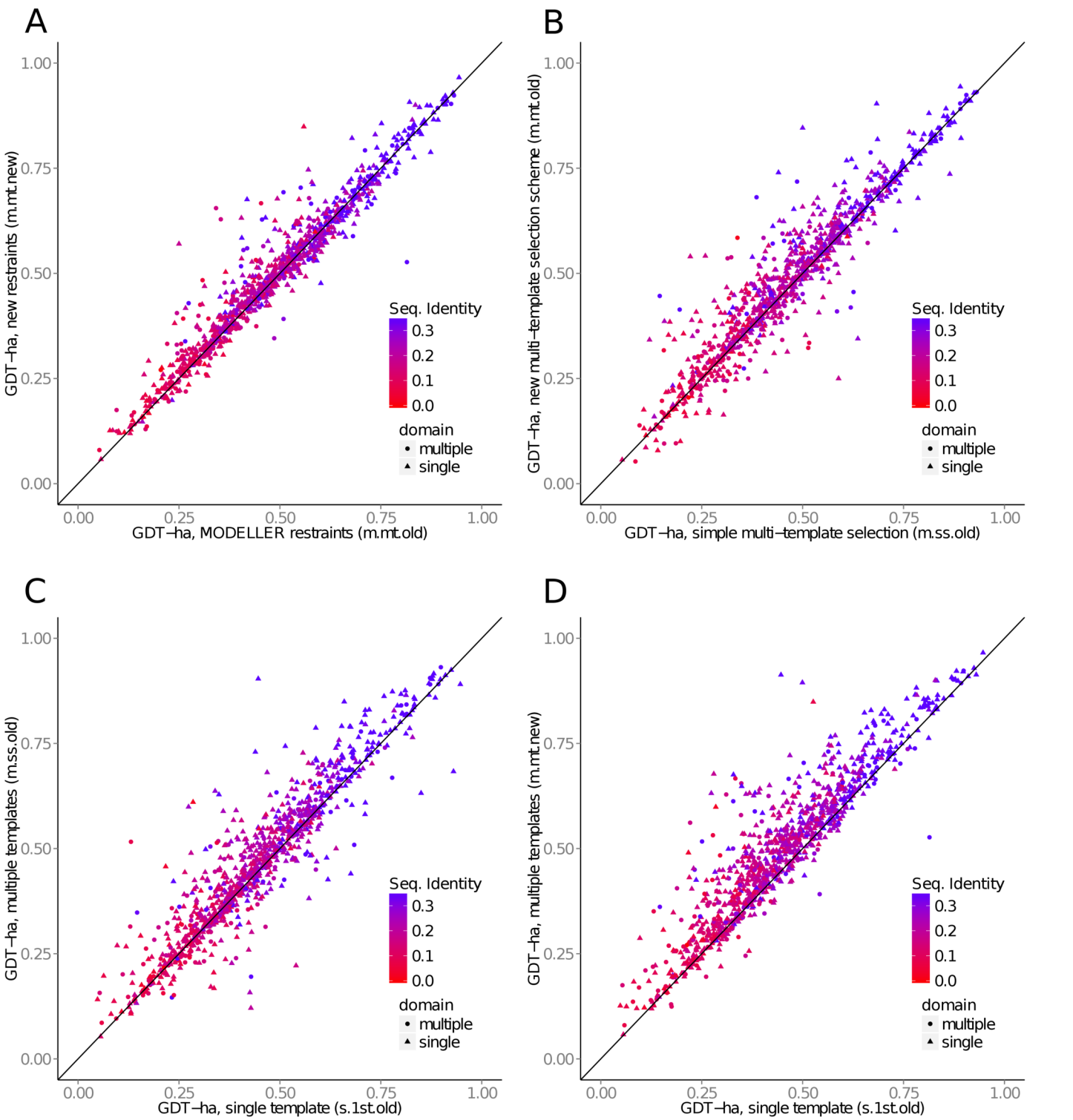

Fig 7. (A) Our two-component mixture restraints improve GDT-HA model quality over MODELLER's default restraints in multi-template modelling by 2.5% on average. (B) Our multi-template selection strategy improves GDT-HA scores over the simple multi-template selection strategy by 3.9% on average. (C) Multi-template modeling improves GDT-HA scores over single-template modelling (using MODELLER restraints) by 4.3% on average. (D) Overall improvements through new restraints, template weights, and the new multiple template selection over the baseline, single-template version (s.1st.old in Table 1) is 11.1%.

doi:10.1371/journal.pcbi.1004343.g007

Table 2. Multi-template homology modeling and the new restraints improve models within core regions independent of increased query sequence coverage. Mean GDT-HAS on query protein core regions, defined as the residues that are covered by the first template. Percent improvement with respect to the previous line.

Templates	Selection	Restraints	GDT-HA score overall	GDT-HA score cores only
Single	first hit	MODELLER	0.443 (-)	0.464 (-)
Multiple	new multi-template	MODELLER	0.480 (+8.4%)	0.504 (+8.6%)
Multiple	new multi-template	new	0.492 (+2.5%)	0.514 (+2.0%)

doi:10.1371/journal.pcbi.1004343.t002

residues in the input alignment to MODELLER. In that way, distance constraints can only be generated on cores. Then we evaluate the resulting models on core residues only and we compare the GDT-HAS with the general case.

Table 2 shows that, first, using multiple templates leads to a clear improvement over single templates both in the core regions and overall. This shows that the effect of adding further templates to the first selected template does indeed improve model quality to a similar extent in the core and non-core regions. Similarly, the improvements due to our new two-component restraints are of the same order in the core regions (+2.0%) as overall (+2.5%), leading to a similar conclusion, that the new restraints improve the model to the same extent in the core and non-core regions.

Robustness with respect to wrong restraints

Our probabilistic multi-template modeling approach should have the advantage over the MODELLER restraints of being more robust towards wrong restraints, because the new distance restraints become flat when $\log d$ deviates strongly from $\log d_t$, i.e., when the restraint cannot be satisfied at all. Therefore, completely wrong restraints practically get ignored in the new approach. Note that this was not a design target of our method but it is simply a consequence of a correct statistical treatment. To test our hypothesis on the robustness of the new restraints, we modified the template selection as follows.

For each query in the test set, we constructed three different template sets (Table 3). The three sets contained two good templates each, and 0, 1 or 2 bad templates, respectively. The good templates were the top two templates according to the TMSCORES predicted by the neural network in Fig. S1 that also attained a true TMSCORE of > 0.5. The bad templates were the lowest ranked templates with a true TMSCORE < 0.3. The average model quality obtained with these three selections are shown in Table 3. As expected, the models built with the new restraints proved to be considerably more robust than the models built with the standard MODELLER pipeline.

CASP assessment

CASP (Critical Assessment of Structure Prediction) is a community wide, double-blind experiment that takes place every second year to objectively test the performance of various

Table 3. The probabilistic multi-template modeling approach is less negatively affected by bad templates. Mean GDT-HA scores of 1000 models built with templates sets containing 0, 1 and 2 bad templates (TMSCORE < 0.3) along with two good templates (TMSCORE > 0.5).

Good templates	Bad templates	MODELLER restraints	New restraints
2	0	0.474 (-)	0.480 (-)
2	1	0.466 (–1.7%)	0.475 (–1.0%)
2	2	0.458 (–3.4%)	0.471 (–1.9%)

doi:10.1371/journal.pcbi.1004343.t003

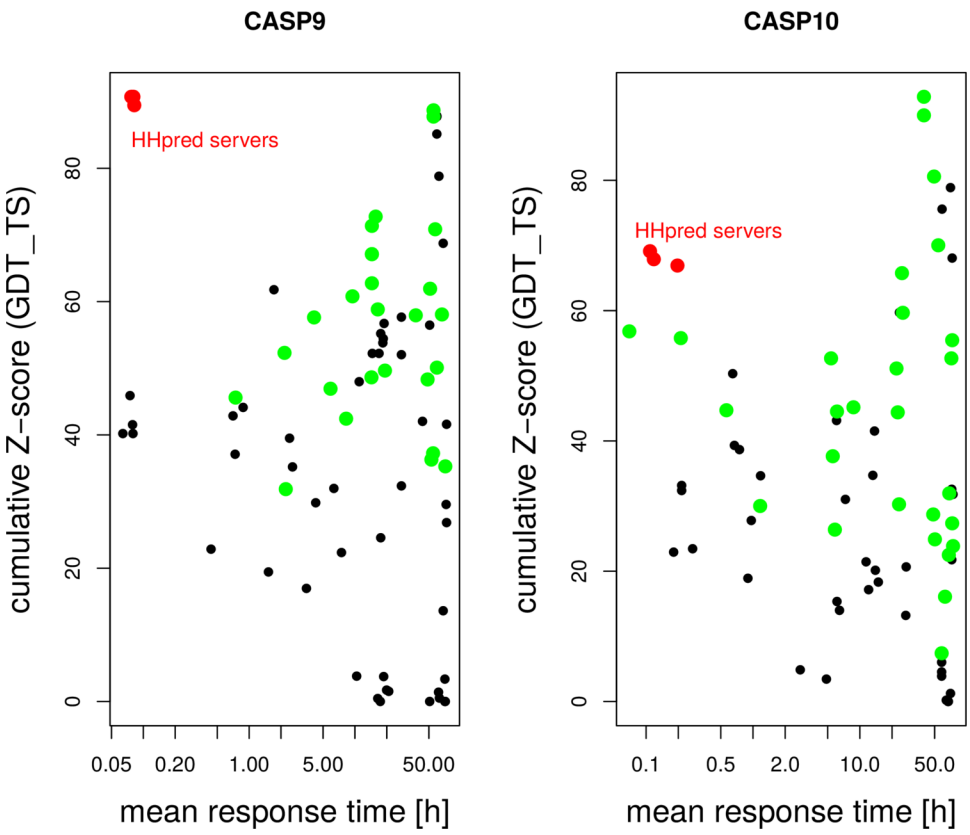

Fig 8. Cumulative Z-score of all server predictions in the template-based modeling category of the CASP9 and CASP10 community-wide assessment of techniques for protein structure prediction [1, 3]. HHpred servers are red, other servers using our HHsuite software are shown in green.

doi:10.1371/journal.pcbi.1004343.g008

predictors. HHpred regularly participates in the server based structure prediction category competing with 70–80 other servers. For CASP9 and CASP10, we integrated all methods described above into the HHpred pipeline.

Depending on whether there existed a suitable template in the databases, all queries are subdivided into two categories: template based (TBM) and free (FM). Due to the ever increasing database sizes, most of the queries are TBM (121 vs. 26 in CASP9 and 111 vs. 15 in CASP10). As Fig 8 shows, for TBM HHpred is among the most accurate servers (top 1 in CASP9 and top 7 in CASP10 according to the official CASP ranking—all three servers differ only in minor technical details, see [31, 32]). At the same time HHpred is faster by a factor of $\sim$ 350 compared with the other leading groups. Fig 8 summarizes the official results in the TBM category from two community-wide assessments of techniques for protein structure prediction, CASP9 (121 query proteins) and CASP10 (111 query proteins) [1, 3]. The values used in the figure were downloaded from the official CASP website (http://predictioncenter.org/). For detailed results, see Supporting Information. When replacing the new restraints with Modeller's default restraints for the CASP10 set on the same selection of templates, the gdtts-score decreased by 3%.

When considering HHpred's performance in CASP9 and CASP10, note that assessors filtered out targets that will be too simple to predict by eliminating targets for which a high-confidence homologous template could be found using HHsearch. This procedure thus selectively

biases the targets at the detriment of HHpred by eliminating targets that would be easy for HHpred to predict.

Discussion

Protein structure prediction is a mature field, in which the best methods differ only by a few percent in performance according to recent CASP benchmarks. Even so, great progress has been made in the last 10 to 15 years in template-based protein structure prediction, fuelled by advances in techniques for remote homology detection and alignment [6] and techniques for model quality assessment [3]. In contrast, most successful servers in CASP employ MODELLER to build their 3D homology models, a software whose core has changed very little since its publication 22 years ago. This speaks to the enormous success of MODELLER's statistical approach to homology modeling. In this study we have shown how to generalize the statistical approach by taking account of alignment errors and treating restraints from multiple templates in a probabilistically satisfactory way.

These theoretical insights have led to improvements in average model quality (around 6.5%) that are somewhat smaller than what we expected initially. In hindsight, MODELLER's heuristic to derive multi-template restraints works surprisingly well. Also, since MODELLER's internal workings (e.g. the stochastic optimization) are optimized together with its own restraints, it might well be possible to improve on the presented results by specifically optimizing MODELLER's model building procedure with our new restraints. We note, however, that an average model score improvement of 4.4% (m.ss.old versus m.mt.new in GDT-TS, see Table A in S1 Text) corresponds to the difference in GDT-TS scores between the 3rd best and 14th best server in CASP10 [5]. This is a considerable success in particular because our theoretical approach is quite general and can be transferred to other homology modelling methods and to the up-and-coming field of modeling large protein complexes from heterogeneous experimental data [33].

We noted during our tests that the positive impact of the new restraints on model quality is strongest when evaluated with the strictest score, GDT-HA, as compared to the less strict GDT-TS or TMSCORE (Table A in S1 Text). Here, strictness refers to how severely already small deviations of the model from the true structure are penalized. This observation shows that the improvements of our new restraints are to a substantial degree in the high-precision regime, i.e., below 1 Å, by further improving regions of the model that are already fairly well modeled. Since the best-modeled regions are expected to largely coincide with the highly conserved and hence functionally most important parts of the protein, we expect the new restraints to have the strongest impact on the functionally most important regions of the model.

We are convinced of the power of probability theory in describing quantitative phenomena under uncertainty. MODELLER is an excellent case in point. An interesting idea is to carry the probabilistic view further by probabilistically integrating structural and sequence information. All approaches so far start from a fixed query-template alignment (or from a set of alternative alignments) and try to find the 3D model that is best compatible with the alignment. To allow information from the 3D modelling to be fed back to the alignment stage and vice versa, it seems promising to explore the joint posterior probability distribution of alignment and 3D structure. One way to do this would be by Markov Chain Monte Carlo Gibbs sampling of the alignment and the model structure from appropriate conditional distributions.

Supporting Information

S1 Text. Contains supplemental figures (single template neural network, sequence identity distributions), tables (model quality scores, alignment features, CASP results) and methods

(details of template weighting).
(PDF)

S1 Fasta. Training, optimization and test set. These correspond to the supplemental files S2_training_set, S2_test_set.fasta and S2_optimization_set.fasta, respectively.
(ZIP)

Acknowledgments

We would like to thank Andrei Sali (UCSF), Markus Meier, and Jessica Andreani for discussions, and Milot Mirdita for bioinformatics support.

Author Contributions

Conceived and designed the experiments: JS AM. Analyzed the data: AM. Wrote the paper: JS AM.

References

1. Mariani V, Kiefer F, Schmidt T, Haas J, Schwede T (2011) Assessment of template based protein structure predictions in CASP9. Proteins 79 Suppl 1: 37–58. PMID: 22002823
2. Kinch L, Yong Shi S, Cong Q, Cheng H, Liao Y, et al. (2011) CASP9 assessment of free modeling target predictions. Proteins 79 Suppl 10: 59–73. doi: 10.1002/prot.23181 PMID: 21997521
3. Huang Yea (2013) Assessment of template-based protein structure predictions in CASP10. Proteins 2: 43–56.
4. Yan R, Xu D, Yang J, Walker S, Zhang Y (2013) A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. Scientific reports 3: 2619. doi: 10.1038/srep02619 PMID: 24018415
5. Kryshtafovych Aea (2014) CASP10 results compared to those of previous CASP experiments. Proteins 82: 164–174. doi: 10.1002/prot.24448
6. Söding J, Remmert M (2011) Protein sequence comparison and fold recognition: progress and good-practice benchmarking. Current opinion in structural biology 21: 404–11. doi: 10.1016/j.sbi.2011.03.005 PMID: 21458982
7. Wallner B, Elofsson A (2005) All are not equal: A benchmark of different homology modeling programs. Protein Science 15: 1315–1327. doi: 10.1110/ps.041253405
8. Dalton JaR, Jackson RM (2007) An evaluation of automated homology modelling methods at low target template sequence similarity. Bioinformatics 23: 1901–8. doi: 10.1093/bioinformatics/btm262 PMID: 17510171
9. Levitt M (1996) Accurate modeling of protein conformation by automatic segment matching. J Mol Biol 226: 507–533. doi: 10.1016/0022-2836(92)90964-L
10. Schwede T, Kopp J, Guex N, Peitsch M (2003) Swiss-model: an automated protein homology-modeling server. Nuc Acid Res 31: 3381–85. doi: 10.1093/nar/gkg520
11. Petrey D, Xiang Z, Tang CL, Xie L, Gimpelev M, et al. (2003) Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. Proteins 53 Suppl 6: 430–435. doi: 10.1002/prot.10550 PMID: 14579332
12. Sali A, Blundell T (1993) Comparative Protein Modelling by Satisfaction of Spatial Restraints. J Mol Biol 234: 779–815. doi: 10.1006/jmbi.1993.1626 PMID: 8254673
13. Joo K, Lee J, Seo JH, Lee K, Kim BG, et al. (2009) All-atom chain-building by optimizing MODELLER energy function using conformational space annealing. Proteins 75: 1010–23. doi: 10.1002/prot.22312 PMID: 19089941
14. Fiser A, Do RK, Sali A (2000) Modeling of loops in protein structures. Protein science: a publication of the Protein Society 9: 1753–73. doi: 10.1110/ps.9.9.1753
15. Remmert M, Biegert A, Hauser A, Söding J (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nature methods 9: 173–5. doi: 10.1038/nmeth.1818
16. Bishop C (1994) Mixture Density Networks. Technical report, Aston University.
17. Durbin R, Eddy S, Krogh A, Mitchison G (1998) Biological sequence analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

18. Altschul S, Carroll R, DJ L (1989) Weights for data related by a tree. J Mol Biol 207: 647–653. doi: 10.1016/0022-2836(89)90234-9 PMID: 2760928

19. Sokal R, Michener C (1958) A statistical method for evaluating systematic relationships. University of Kansas Science Bulletin 38: 1409–38.

20. Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic acids research 33: 2302–9. doi: 10.1093/nar/gki524 PMID: 15849316

21. Xu J (2005) Fold recognition by predicted alignment accuracy. IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM 2: 157–65. doi: 10.1109/TCBB.2005.24 PMID: 17044180

22. Hildebrand A, Remmert M, Biegert A, Söding J (2009) Fast and accurate automatic structure prediction with HHpred. Proteins 77 Suppl 9: 128–32. doi: 10.1002/prot.22499 PMID: 19626712

23. Peng J, Xu J (2011) RaptorX: exploiting structure information for protein alignment by statistical inference. Proteins 79 Suppl 1: 161–71. doi: 10.1002/prot.23175 PMID: 21987485

24. Cheng J (2008) A multi-template combination algorithm for protein comparative modeling. BMC structural biology 8: 18. doi: 10.1186/1472-6807-8-18 PMID: 18366648

25. Fernandez-Fuentes N, Rai BK, Madrid-Aliste CJ, Fajardo JE, Fiser A (2007) Comparative protein structure modeling by combining multiple templates and optimizing sequence-to-structure alignments. Bioinformatics 23: 2558–65. doi: 10.1093/bioinformatics/btm377 PMID: 17823132

26. Wang Z, Eickholt J, Cheng J (2010) MULTICOM: a multi-level combination approach to protein structure prediction and its assessments in CASP8. Bioinformatics 26: 882–8. doi: 10.1093/bioinformatics/btq058 PMID: 20150411

27. Zhang Y (2008) I-TASSER server for protein 3D structure prediction. BMC bioinformatics 9: 40. doi: 10.1186/1471-2105-9-40 PMID: 18215316

28. Zhang Y, Skolnick J (2004) SPICKER: A Clustering Approach to Identify Near-Native. J Comput Chem 25: 865–871. doi: 10.1002/jcc.20011 PMID: 15011258

29. Read RJ, Chavali G (2007) Assessment of CASP7 predictions in the high accuracy template-based modeling category. Proteins 69: 27–37. doi: 10.1002/prot.21662 PMID: 17894351

30. Larsson P, Wallner B, Lindahl E, Elofsson A (2008) Using multiple templates to improve quality of homology models in automated homology modeling. Protein Science: 990–1002. doi: 10.1110/ps.073344908 PMID: 18441233

31. Casp9 abstract book: the embo conference (2010). URL http://predictioncenter.org/casp9/doc/Abstracts.pdf.

32. Casp10 abstract book: the embo conference (2012). URL http://predictioncenter.org/casp10/doc/CASP10_Abstracts.pdf.

33. Russel D, Lasker K, Webb B, Velazquez-Muriel J, Tjioe E, et al. (2012) Putting the pieces together: Integrative modeling platform software for strucutre determination of macromolecular assemblies. PLoS Biol 10. doi: 10.1371/journal.pbio.1001244 PMID: 22272186