doi:10.1016/j.jmb.2009.02.031 J. Mol. Biol. (2009) 388, 180–194

JMB

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Inferential Optimization for Simultaneous Fitting of Multiple Components into a CryoEM Map of Their Assembly

Keren Lasker[1,2]*, Maya Topf[3], Andrej Sali[2]* and Haim J. Wolfson[1]*

[1]*Blavatnik School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel-Aviv 69978, Israel*

[2]*Department of Bioengineering and Therapeutic Sciences, Department of Pharmaceutical Chemistry, and California Institute for Quantitative Biosciences (QB3), University of California at San Francisco, San Francisco 94158, CA, USA*

[3]*Institute of Structural and Molecular Biology, School of Crystallography, Birkbeck College, University of London, Malet Street, London WC1E 7HX, UK*

Received 7 September 2008; received in revised form 29 December 2008; accepted 12 February 2009
Available online 20 February 2009

Edited by M. Levitt

Models of macromolecular assemblies are essential for a mechanistic description of cellular processes. Such models are increasingly obtained by fitting atomic-resolution structures of components into a density map of the whole assembly. Yet, current density-fitting techniques are frequently insufficient for an unambiguous determination of the positions and orientations of all components. Here, we describe MultiFit, a method used for simultaneously fitting atomic structures of components into their assembly density map at resolutions as low as 25 Å. The component positions and orientations are optimized with respect to a scoring function that includes the quality-of-fit of components in the map, the protrusion of components from the map envelope, and the shape complementarity between pairs of components. The scoring function is optimized by our exact inference optimizer DOMINO (*D*iscrete *O*ptimization of *M*ultiple *IN*teracting *O*bjects) that efficiently finds the global minimum in a discrete sampling space. MultiFit was benchmarked on seven assemblies of known structure, consisting of up to seven proteins each. The input atomic structures of the components were obtained from the Protein Data Bank, as well as by comparative modeling based on a 16–99% sequence identity to a template structure. A near-native configuration was usually found as the top-scoring model. Therefore, MultiFit can provide initial configurations for further refinement of many multicomponent assembly structures described by electron microscopy.

Keywords: electron microscopy; protein structure modeling; docking; optimization; macromolecular assemblies

Introduction

Structural description of macromolecular assemblies is essential for a mechanistic understanding of the cell.[1] The scope of the problem is revealed by protein interaction studies: The yeast cell contains approximately 800 distinct core complexes of 4.9 proteins, on average,[2] most of which have not yet been structurally characterized.[3] The human proteome is likely to have an order of magnitude more distinct assemblies than the yeast cell. Therefore, there are thousands of biologically relevant assemblies whose structures still need to be determined.

Structural determination of macromolecular assemblies is a major challenge in structural biology. X-ray crystallography can provide structures of stable assemblies at atomic resolution.[4] However, there are many other assemblies that are refractory to crystallographic determination. A low-resolution structure of these assemblies can be determined by

**Corresponding authors.* K. Lasker and H.J. Wolfson are to be contacted at School of Computer Science, Tel Aviv University, Tel-Aviv 69978, Israel. A. Sali is to be contacted at UCSF MC 2552, Byers Hall at Mission Bay, Suite 503B, University of California at San Francisco, 1700 4th Street, San Francisco, CA 94158, USA. E-mail addresses: kerenl@tau.ac.il; sali@salilab.org; wolfson@tau.ac.il.

Abbreviations used: cryoEM, cryo-electron microscopy; PDB, Protein Data Bank.

cryo-electron microscopy (cryoEM).[5] The resolution usually ranges from 4 Å, where the backbone of the protein can be traced, to 30 Å, where only the outer envelope of the assembly is visible.[6]

The increasing numbers of atomic and cryoEM data sets[7,8] have stimulated the development of computational techniques for fitting atomic structures of assembly components into a cryoEM density map of the whole assembly. The result is a pseudo-atomic model of the assembly that can reveal significant insights into its structure, dynamics, function, and evolution.[9–13]

Here, we focus on determining the positions and orientations (i.e., placements) of multiple atomic component models within the assembly density. When the structure of a homologous assembly (template) is available, the placements of the components can be computed by fitting the template into the target assembly density, superposing the target component models on the corresponding template components, and refining the model.[14,15] Alternatively, the component positions can be determined experimentally by protein-labeling methods relying, for example, on gold-labeled antibodies.[16] However, when only a cryoEM map and component structures are available, a general method for solving the configuration problem is not yet available.

A sequential method for fitting multiple components into an assembly map has been described.[17] The method starts by fitting of the largest component into the map, followed by iterative fitting of the largest remaining component into the unoccupied density until all components have been fitted. The fitting of a component into a given map can be performed manually using interactive visualization tools.[18] More desirably, automated fitting methods that assess the placement of a component by a fit between the component and a segmented[6] or complete density of the assembly can also be used; the fit is optimized over the translational and rotational degrees of freedom of a rigid component relative to the map.[19] The sequential method is applicable if the remaining components can be unambiguously fitted into the unoccupied densities. Unfortunately, this condition is generally not satisfied, especially when the resolution is low, the number of components is large, and component models are inaccurate.[20] For example, sequential fitting is not expected to work for the 19S proteasome with 20 component proteins,[21] for the mammalian ribosome for which 30 of 80 proteins are not present in known archaeal or bacterial ribosomes,[14] or for the ryanodine receptor isoform 1 for which some domains are poorly modeled while no template is available for others.[22]

Here, we describe a method named MultiFit for determining the configuration of multiple high-resolution component structures based on the quality-of-fit of each component into the density map, the protrusion of each component from the map envelope, and the shape complementarity between pairs of components. The combination of these terms reduces the ambiguity of the final solution, compared to using any individual term on its own.

The task of sampling the configuration space is challenging because the placement of a component depends on the placement of other components. MultiFit tackles this combinatorial challenge by reformulating the problem as an inferential optimization over a discrete sampling space. In outline, a discrete set of possible placements for each component is first generated independently of other components. Next, the globally optimal combination of placements with respect to a scoring function is found by a combination of branch-and-bound search and the DOMINO (*D*iscrete *O*ptimization of *M*ultiple *IN*teracting *O*bjects) inferential optimizer. The relative translations and orientations of pairs of components in the best-ranking configurations are then refined; specifically, a refined discrete sampling space is generated by pairwise geometrical docking between interacting components, and the optimal refined combination of placements is again found using DOMINO. We successfully validated the method on a simulated benchmark of six assemblies consisting of up to seven proteins each. In addition, for a more realistic test, we determined the configuration of four domains in the subunit of GroES–ADP7–GroEL–ATP7 chaperonin from *Escherichia coli* based on an experimentally determined map at a resolution of 23.5 Å.[23] A near-native configuration scored best in four test cases, third best in two cases, and fourth best in the remaining case.

Below, we begin with a detailed description of general combinatorial optimization by DOMINO, followed by a formal definition of the component configuration problem and the MultiFit algorithm for a solution using DOMINO (Theory). We then demonstrate the performance of MultiFit on benchmark cases (Results). Finally, we discuss the implications of MultiFit and DOMINO for the structural characterization of large assemblies (Discussion).

Theory

Combinatorial optimization by DOMINO

DOMINO applies a divide-and-conquer approach to efficiently find solutions with the globally optimal score within a discrete sampling space (Fig. 1).[24,25] The idea is to decompose the set of variables into relatively uncoupled but potentially overlapping subsets that can be sampled independently of each other, and then to efficiently gather the subset solutions into the global minimum. The strength of this approach derives from the decomposition procedure that helps reduce the size of the search space from exponential in the number of components in the whole system to exponential in the number of components in the largest subset. Next, we describe DOMINO's application to the minimization of a scoring function F corresponding to a sum of single-body terms $\{\alpha_i\}$ and pairwise terms $\{\beta_{i,j}\}$:

$$F(y_1, \ldots, y_n) = \sum_i \alpha_i(y_i) + \sum_{i<j} \beta_{i,j}\left(y_i, y_j\right)$$

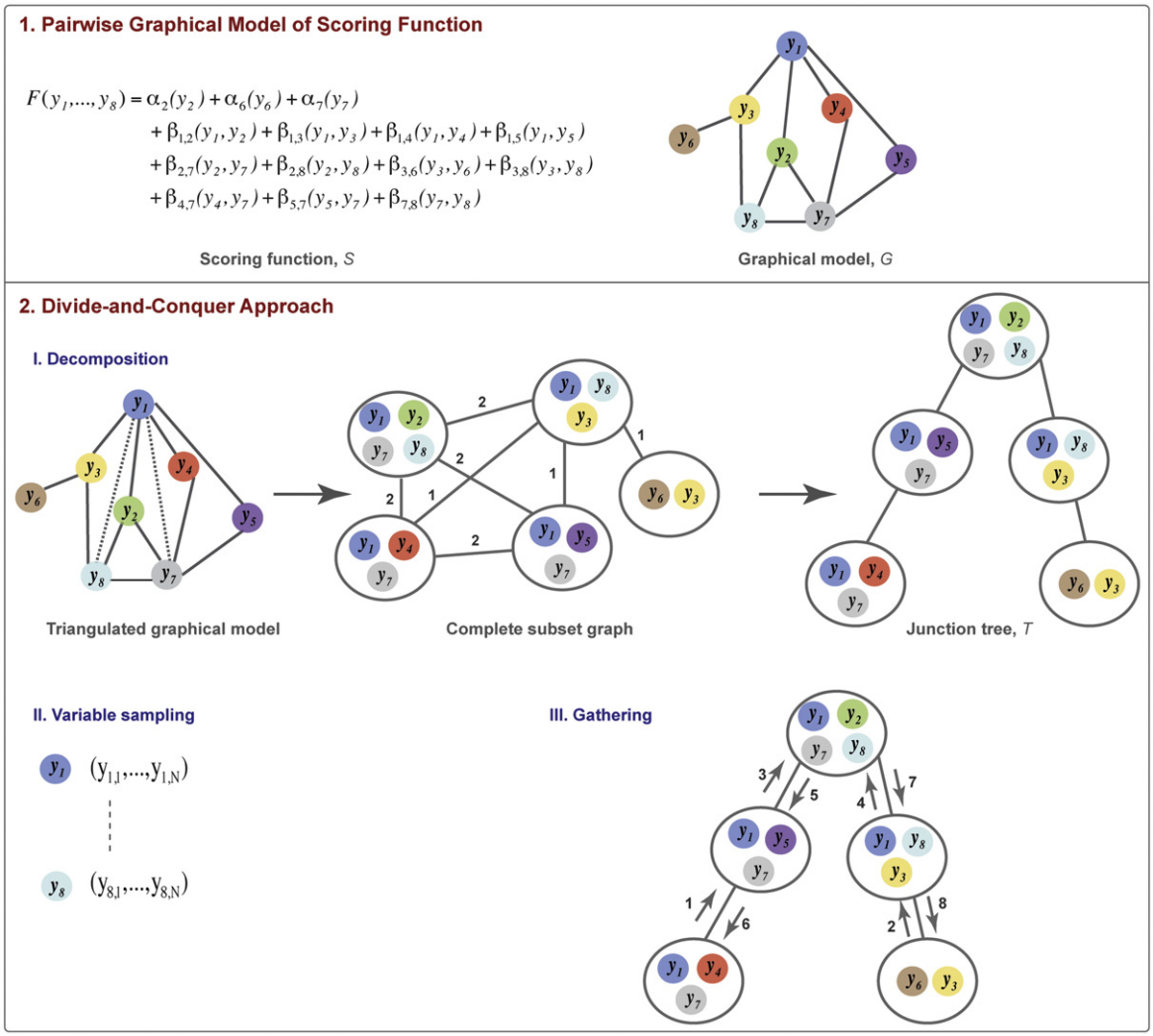

Fig. 1. DOMINO outline. (1) The DOMINO optimizer is illustrated with the scoring function F of 8 variables $\{y_i\}$ composed of a sum between 3 single-body terms $\{\alpha_i\}$ and 11 pairwise terms $\{\beta_{i,j}\}$. The scoring function is encoded in the graphical model G. (2) (I) Decomposition of the graphical model results in a junction tree T.[24,25] The graphical model is first triangulated; a graph is triangulated if there are no cycles with more than three edges without a chord (a chord is an edge connecting two nonadjacent nodes in a cycle). The triangulation procedure adds edges (dotted lines) to the graphical model until no cycle is chordless. The triangulated graphical model is then converted into a complete subset graph. The nodes of the complete subset graph are maximum cliques in the triangulated graphical model (gray circles); a maximum clique is a subgraph whose nodes are connected directly to each other and are not all part of another clique. The weight of an edge in the complete subset graph is the number of shared variables between the adjacent subsets, as indicated; edges of weight zero are not shown. Next, the junction tree is the maximum spanning tree of the complete subset graph; a maximal spanning tree of a graph spans all of the nodes without cycles, using a subset of the original edges with the maximal sum of their weights. (II) The sampling space of each variable is discretized. (III) Finally, the globally optimal solution of F is gathered from enumerated subset states by passing messages between subset nodes. The numbers on the edges indicate a valid sequence of message passing.

where $\{y_i\}$ are the variables being optimized (e.g., in MultiFit, these variables are the positions and orientations of the components). The scoring function F is represented by a graphical model $G=(V,E)$. The graphical model G of the scoring function F is a graph whose nodes V correspond to the variables $\{y_i\}$, and edges E connect pairs of nodes. The weight of a node corresponding to y_i is α_i, and the weight of an edge between nodes corresponding to y_i and y_j is $\beta_{i,j}$. Thus, the scoring function F is the sum of all node and edge weights.

The problem of finding the minimum of the scoring function F is equivalent to the maximum *a posteriori* problem in a graphical model. This problem is known to be NP-hard (nondeterministic polynomial—time hard) for an arbitrary graph G;[26] NP-hard is a class of decision, search, and optimization problems whose computing time increases at least exponentially with the number of optimized variables. When a graphical model has, at most, one path between any two given nodes (i.e., it does not contain cycles and thus is a singly connected

graphical model or a tree), it can be efficiently optimized by the belief-propagation algorithm.[27]

Unfortunately, the belief-propagation algorithm is not guaranteed to converge into the globally optimal solution for graphs with cycles, such as the graphical models used for the MultiFit application. Therefore, to ensure that we find the global minimum of G efficiently, we apply a divide-and-conquer approach. First, the variables to be optimized are decomposed into smaller relatively uncoupled but potentially overlapping subsets, using a junction tree construction algorithm (the decomposition step). Second, a discrete sampling space is generated for each variable (the variable sampling step; e.g., by uniform sampling). Third, the discrete states of the individual subsets are constructed and gathered into the globally optimal solution, using the belief-propagation algorithm (the gathering step). Graph theory provides

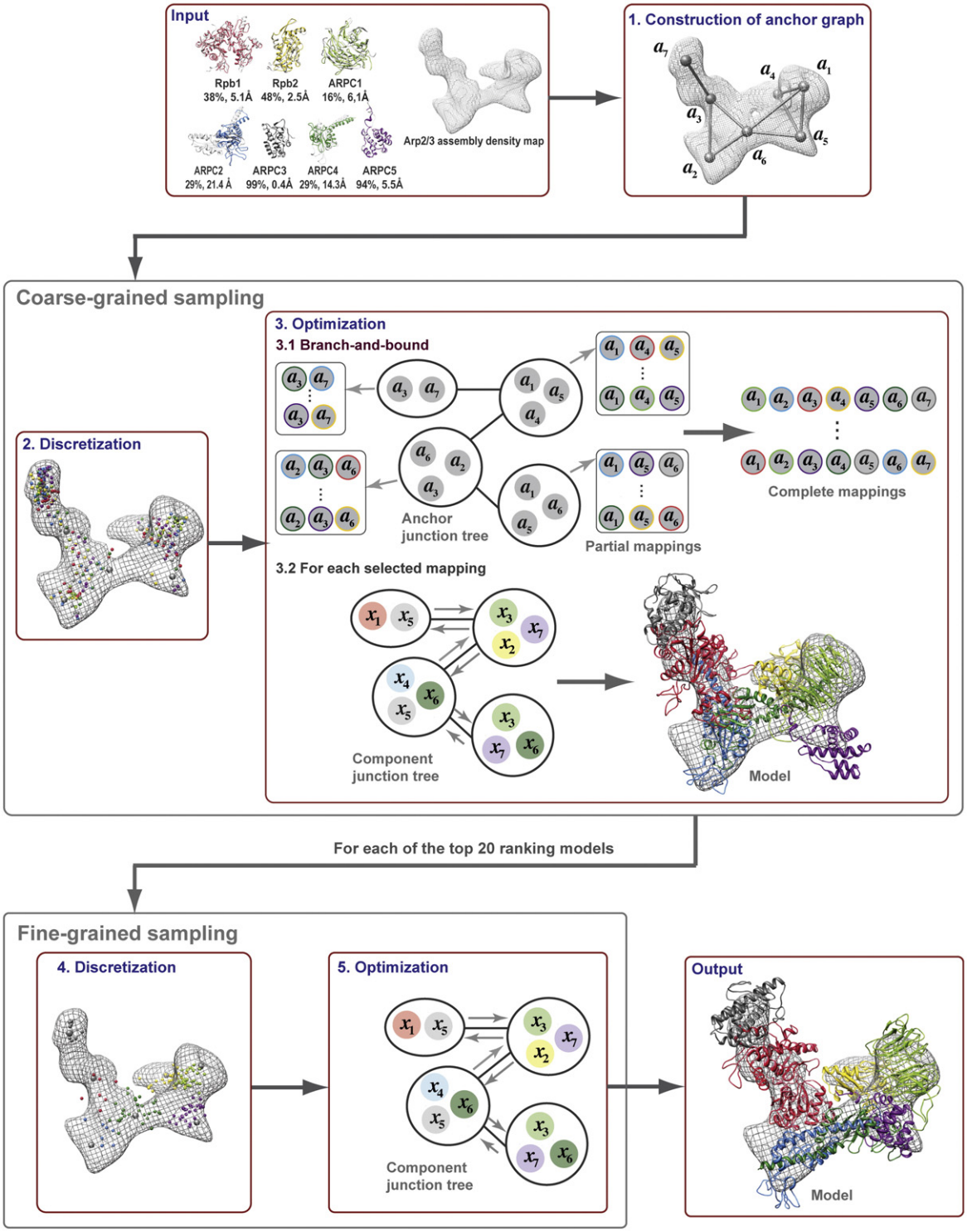

Fig. 2 (*legend on next page*)

efficient algorithms for decomposition (i.e., junction tree construction) and gathering (i.e., belief propagation). Next, we elaborate on each of the three steps.

In the decomposition step, the graphical model G is converted into a tree T whose nodes U are potentially overlapping subsets of variables $\{y_i\}$ (Fig. 1). Importantly, for any two nonadjacent subsets in T that share some variables, the subsets that connect them must also contain these variables (i.e., T is a junction tree). For any junction tree, it is possible to gather the discrete states of individual subsets into the globally optimal solution using the belief-propagation algorithm. For maximum efficiency, we aim toward decomposing the graphical model into the junction tree such that the size of the largest subset is minimal, which is an NP-hard problem. We use the minimum-degree method that was shown empirically to result in smallest subsets for sparse graphical models.[28]

In the variable sampling step, a discrete set of values for each variable is created. The details of this discretization may depend on the scoring function F. Most generally, uniform sampling over a relevant range of values can be used. A potentially better possibility is to use the union of the local minima of scoring functions spanned by the variables in the subsets containing the discretized variable.

In the final step (gathering step), the HUGIN version of the belief-propagation algorithm[29] is applied to the junction tree T to find the global minimum of F (Fig. 1). The computational complexity of the HUGIN algorithm is $O(|U|L^s)$, where s is the size of the largest subset of T, and L is the number of values of a variable y_i.

The belief-propagation algorithm is based on passing messages between the nodes (i.e., subsets of variables) of the junction tree. A subset is allowed to send a message to a neighbor subset if it has received messages from all of its remaining neighbor subsets. Thus, propagation of messages is initiated in subsets connected only to a single subset (i.e., the leaf subsets) and proceeds to the neighboring subsets until some subset receives messages from all of its neighbors (i.e., the root subset). The content of a message to a target subset is a vector of the minimal values of the partial scoring function F over the variables in all previously visited subsets and the target subset, for each combination of values of the remaining variables in the target subset; a partial scoring function over a subset of variables includes only those terms of F that involve these variables. Messages from the root subset are then sent back to the other subsets, completing the message-passing process when the leave subsets receive back the messages from the root subset. For messages from the root subset, the partial scoring function is the scoring function F (because all subsets have been already visited), and thus each subset that received a message from all other subsets can infer the values of its variables in the global minimum. The efficiency of message passing derives from enumerating combinations of values for only those variables that are shared between different subsets.

MultiFit: Simultaneous fitting of multiple components into a density map of their assembly

The goal is to find the positions and orientations (i.e., placements) of components (e.g., subcomplexes, proteins, domains, and secondary structure segments), represented at atomic resolution, within a cryoEM density map of their assembly. We express this structure characterization challenge as a combinatorial optimization problem. Next, we outline a representation of the modeled system, a scoring function, and an optimization algorithm.

Fig. 2. MultiFit outline. The MultiFit algorithm is illustrated using an assembly between models of Rpb1 (red), Rpb2 (yellow), ARPC1 (light green), ARPC2 (blue), ARPC3 (gray), ARPC4 (dark green), and ARPC5 (purple) (PDB entry 1tyq). The component–template sequence identities and C^{α} RMSDs are indicated. The input to MultiFit is the assembly density map (gray mesh) and the atomic structures of the individual components (top left). The output is a ranked list of assembly models that optimize the MultiFit scoring function (one model is shown on the bottom right). (1) The anchor points (the seven labeled nodes) are constructed for the input density map by k-cluster; the nine gray edges indicate pairs of anchor points that are sufficiently close to allow components placed in their vicinity to interact with each other. (2) The sampling space of component placements is discretized by fitting each of the seven components around each of the seven anchor points (regions) and by selecting a number of top-ranking placements for each component in each region; the small colored spheres indicate placement centroids. (3) The optimal combination of component placements is found by optimizing the scoring function S for each mapping of components to anchor points using DOMINO. (3.1) For efficiency, we replace the enumeration by a branch-and-bound procedure that eliminates some of the mappings and makes use of partial results. In the branch stage, we first decompose the anchor graph into an anchor junction tree using DOMINO's decomposition algorithm (Fig. 1). The top 60% mappings of components to anchor points for each subset of anchor points (partial mappings) are found and stored by iterating over all possible partial mappings; the color of the circle indicates which component is mapped to the anchor point. The partial mappings are scored by partial scoring function S, including only the terms involving the mapped components. Complete mappings consistent with the stored partial mappings are generated efficiently with a hashing procedure (not described). (3.2) Next, for each of these complete mappings, the optimal combination of placements for the seven components is found by DOMINO; the color of the solid circles in the component junction tree indicates the component mapped to the corresponding anchor point in the anchor junction tree. The molecular model shown has a mapping score of 0 and a rank of 10th. (4) The 20 top-scoring coarse models are further refined. A refined sampling space is generated for each coarse configuration by docking pairs of its interacting components and by selecting only those placements that are approximately consistent with the initial coarse configuration. (5) DOMINO is applied again to find the optimal combination of placements for the seven components; the molecular model shown has a mapping score of 0 and a rank of 4th.

Representation

The assembly density map is represented by a three-dimensional grid in which every voxel is assigned an estimated density value. The components are represented by their atoms and remain rigid throughout the entire optimization process (Fig. 2).

Scoring

We evaluate potential configurations based on the quality-of-fit of individual components into the density map, the protrusion of each component from the map envelope, and the shape complementarity between pairs of components.

Optimization

The component configuration that optimizes the scoring function is identified by a combinatorial optimization protocol consisting of three stages: (i) anchor graph construction, (ii) coarse-grained sampling, and (iii) fine-grained sampling (Fig. 2). In anchor graph construction, the density map is discretized into regions, and the connectivity between them is calculated. In coarse-grained sampling, the sampling space is first discretized by fitting each of the components into each of the map regions and selecting a number of top-ranking placements for each component in each region. Next, a branch-and-bound search through all mappings of components to regions combined with DOMINO finds 20 top-scoring configurations. In fine-grained sampling, each of these top configurations is refined by DOMINO; a refined sampling space is generated for each coarse configuration by docking pairs of its interacting components and selecting only those placements that are approximately consistent with the initial coarse configuration.

Scoring function for MultiFit

The score of placements of N components[30] in an assembly density map is:

$$S(x_1, \ldots, x_N) = \sum_i \{\varphi_1(x_i) + \varphi_2(x_i)\} + \sum_{i<j} \varphi_3\left(x_i, x_j\right)$$

$\varphi_1(x_i)$ is the quality-of-fit of x_i into the assembly density map D. In the extreme case, the configuration that optimizes $\sum_i \varphi_1(x_i)$ may occupy only the highest-density region in the assembly density map. To overcome this problem, we add two geometric terms (φ_2 and φ_3) to the scoring function. The component protrusion term $\varphi_2(x_i)$ scores how well x_i is placed inside the density envelope. The interaction term $\varphi_3(x_i,x_j)$ scores the pairwise shape complementarity between the structures x_i and x_j.

Quality-of-fit term

The fit of a given structure x_i into the assembly density map D is usually assessed by a cross-correlation measure between the densities of x_i and the assembly.[5,20] Here, we use the "normalized fitting score" C as implemented in Mod-EM (Eq. (2) in Topf *et al.*[31]); the density of x_i is simulated at the same resolution as the assembly density map D, using the uniform-sphere model. However, C is insufficient for comparing placements of different components because small domains have a better chance of higher cross-correlation with the map.[32] Thus, we calculate the quality-of-fit of a component into a map by expressing C as a Z-score $(C-m)/s$, where m and s are, respectively, the mean and standard deviation of a reference distribution of C. The reference distribution is generated by optimally fitting randomly selected, similarly sized protein structures into simulated maps of randomly selected, similarly sized protein structures (F. Davis, M. S. Madhusudan, N. Eswar, A. Sali, and M. Topf, unpublished results).

Interaction term

The pairwise shape complementarity score between main-chain atoms of the structures x_i and x_j is calculated as the weighted sum of a reward for interaction surfaces and a penalty for steric clashes between the components.[33,34] Specifically, the reward is the total number of surface atom pairs of x_i and x_j within a distance cutoff, and the penalty is a weighted sum of all clashing pairs of atoms of x_i and x_j. To speed the calculation of the reward, we first classify atoms as buried or exposed by placing each component structure on a grid and dividing the grid into a surface and four core shells according to the closest distance from the molecular surface (the surface shell contains all grid points that are, at most, half of the map resolution away from the surface).[33] The reward is calculated by indexing the surface atoms of x_i in a geometric hash table,[35,36] querying the hash table for each surface atom of x_j, and summing the number of hits to get the reward. To calculate the steric clash penalty, we determine the accessibility of each atom of x_i (and x_j) using the grid of x_j (x_i). If an atom in x_i (x_j) is located within the surface ($k=0$) or the k^{th} core shell of x_j (x_i), we add $(k+1)\times 27$ to the penalty. The sum of the penalty score of x_i with respect to x_j and the penalty score of x_j with respect to x_i is divided by 2 to obtain the steric clash penalty. Due to fitting and modeling errors, the correct configuration of components might include some minor clashes between interacting components. These clashes are not significantly penalized because of the thickness of the surface shell and because of the evaluation of the favorable and penalty terms using only main-chain atoms. The choice of shell thickness and weight of the penalty score was chosen by trial and error.

Component protrusion

The protrusion of a component from the assembly envelope is defined to be the negative value of the shape complementarity score between the component surface and the assembly envelope. The assembly envelope is calculated by representing each density

voxel above a threshold as an atom and calculating the Connolly surface[37] of this collection of atoms.

Optimization for MultiFit

Construction of anchor graph

The centroids of L approximately equally sized regions of density voxels are calculated from the density map D using a k-means clustering[38] (k-cluster) that is similar to the QVOL procedure of Situs;[39] a density voxel belongs to the region with the closest centroid. When L equals N, and the components are of similar sizes, the centroids of the regions correspond approximately to the centroids of the N assembly components. These points are the nodes of the anchor graph. We then calculate the connectivity between the anchor points (i.e., the edges of the anchor graph); a pair of anchor points (a_i,a_j) is connected if their regions are in contact.

Discretization step in coarse-grained sampling

We construct a discrete sampling space of component placements, represented by a set of M' placements (by default, 50) for each of the N components in each of the L regions. Thus, each set of placements for all components in region i (A_i) contains $M=M'N$ "local" placements around its anchor point a_i. Here, we set L to N, although L can also be larger than N.

In detail, for each component j, the discrete sampling space is constructed as follows. Placements around each anchor point a_i are sampled by optimizing the normalized fitting score C in a cube surrounding the anchor point (the edge length of the cube is half the resolution of the map). This optimization is performed by Mod-EM,[31] starting with a random starting orientation of the component centered at the anchor point. Next, the sampled placements for all anchor points are clustered based on their pairwise C^α RMSD values: The highest scored placement (by C) initiates the first cluster and is its pivot. The closest remaining placement either is joined with the first cluster for which its C^α RMSD with the cluster's pivot is less than the threshold (half the resolution of the map) or initiates its own cluster otherwise. The process is repeated with the best-scoring nonclustered placement until all placements have been clustered. The best-scoring placement from each cluster is assigned to the set of placements $A_{i,j}$ corresponding to the closest anchor point a_i; each anchor point is assigned, at most, M' placements.

Optimization step in coarse-grained sampling

We find the optimal combination of placements of components by optimizing the scoring function S within the discrete sampling space constructed in the previous step. The global minimum of S is the minimum of the optimal solutions for each of the $L!/N!$ mappings of components to anchor points $\Pi=\{\pi_k\}$, where π_k is a function that maps a component j to an anchor point i ($i=\pi_k\{j\}$); formally, we solve $min_{\pi_k\in\Pi}\, min_{\{(x_1,\ldots,x_N)\,|\,\pi_k\}}\, S(x_1,\ldots,x_N)$, where x_j are placements of component j in the set $A_{\pi_k(j),j}$, as constrained by mapping π_k.

Naively, this optimization could be achieved by a nested double loop in which the outer loop consists of enumerating the mappings and the inner loop consists of applying DOMINO to the scoring function S constrained by the given mapping. However, enumerating over all possible mappings becomes computationally expensive as the number of components increases. To improve the efficiency of MultiFit, we replace the enumeration by a branch-and-bound procedure that eliminates some of the mappings and makes use of partial results (Fig. 2).

The scoring function F optimized by DOMINO for each mapping $\left(\min_{\{(x_1,\ldots,x_N)\,|\,\pi_k\}} S(x_1,\ldots,x_N)\right)$ is a simplified S that does not contain uninformative interaction terms φ_3 corresponding to physically noninteracting components (Fig. 2); specifically, we eliminate interaction terms between pairs of components that are mapped to unconnected anchor points. Importantly, it is this simplification that results in a relatively "sparse" graphical model G, thus allowing it to be optimized efficiently by DOMINO.

Discretization step in fine-grained sampling

We construct a refined discrete sampling space for a coarse configuration found in coarse-grained sampling $(x_1^0,\ldots,x_N^0)$. The refined set of placements of component j is first initialized with the placements in $A_{\pi(j),j}$, as found in coarse-grained sampling. We then enrich this set of placements by sampling the binding of component j to neighboring components $\{w\}$ with PATCHDOCK.[33] A PATCHDOCK-produced binding mode of component j to component $w(x_j)$ is added to the refined set of placements of component j if (i) the distance between the centroid of x_j and the centroid of x_j^0 is below half of the resolution of the map and (ii) x_j is consistent with the density map boundaries [i.e., if $\varphi_2(x_j)$ is below a predefined threshold]. Finally, the refined set of placements of component j is reranked by the quality-of-fit score and clustered according to C^α RMSD (as described above).

Optimization step in fine-grained sampling

The optimal combination of component placements is found by DOMINO by optimizing the scoring function S within the refined discrete sampling space.

Results

Benchmark with simulated maps

Benchmark

We tested MultiFit on a benchmark of six simulated test cases. The assembly density maps

were simulated at 20 Å resolution using the PDB2VOL program of Situs[40] with voxel size of 3 Å. The input atomic structures of the components included native structures from the Protein Data Bank (PDB[41]), as well as models calculated by comparative modeling using MODELLER-9v3†[42] based on related template structures with sequence identity ranging from 16% to 99%. The accuracy of the individual comparative models is quantified using C^{α} RMSD and native overlap to the corresponding native structure. Native overlap (NO3.5) measures the percentage of C^{α} atoms of the model that are within 3.5 Å from the corresponding C^{α} atoms in the native structure. The native overlap was calculated by superposing the model on the corresponding native structure using rigid-body least-squares minimization, as implemented in the *model.superpose* command of MODELLER-9v3.

We use three scores to quantify the accuracy of modeled configurations at different levels of resolution. First, the mapping score is the number of substitutions needed to convert the assessed mapping of components to anchor points into the native mapping of components to anchor points (the Hamming distance); the native mapping has a mapping score of 0. Second, the configuration score is the fraction of the components positioned correctly; we define a component as positioned correctly if the distance between its centroid and the corresponding reference centroid is smaller than half of the map resolution. Third, the assembly placement score is the average of its component placement scores, each of which is composed of a distance and an angle to the reference placement; the distance is calculated between the centroids of the placements, and the angle is the axis angle of the rotation matrix between the two placements.[43] Because the components are kept rigid throughout the optimization process, the reference components used in the assessment of an assembly model are the component models superposed on the corresponding components in the native assembly (i.e., the reference placement). We chose not to use the C^{α} RMSD measure to assess assembly models because the significance of C^{α} RMSD values depends strongly on the number of assembly components and their sizes.[44]

Determining the configuration of Arp2/3

To illustrate MultiFit, we first describe in detail a challenging application to Arp2/3 (Table 1, Figs. 2 and 3). The Arp2/3 complex of seven proteins is crucial for regulating the initiation of actin polymerization and the organization of the resulting filaments.[45] A density map was simulated from the Arp2/3 crystal structure with ATP and Ca^{2+} (PDB entry 1TYQ[46]). The atomic structures of the Arp2/3 components (proteins) were modeled using templates with sequence identity ranging from 16% to

† http://salilab.org/modeller

Table 1. Determining the configuration of the Arp2/3 assembly

Component (name, chain, residue range)	Component modeling				Coarse-grained sampling		Fine-grained sampling	
	Template (PDB entry, residue range)	% Sequence ID[a]	C^{α} RMSD (Å)[b]	NO3.5 (%)[b]	Discretization (best placement score, fitting rank)[c]	Optimization (best-scoring placement score)[d]	Discretization (best placement score, fitting rank)[c]	Optimization (best-scoring placement score)[d]
Rpb1, A, 4–408	2q1nB, 4–370	40	5.1	74	(4.4,12), 22	(7.3, 179)	(4.4, 12), 4	(4.4, 12)
Rpb2, B, 143–349	1nwkA, 140–334	48	2.5	93	(2.4,30), 32	(2.5, 30)	(2.4, 30), 39	(9.2, 14)
ARPC1, C, 5–361	1erjC, 342–708	16	6.1	52	(3.6, 20), 42	(12.1, 115)	(3.6, 20), 101	(1.4, 44)
ARPC2, D, 1–274	1u2vF, 3–168	29	21.4	42	(14.9, 52), 38	(19.9, 172)	(9.1, 25), 5	(9.1, 25)
ARPC3, E, 1–169	2p9nE, 2–173	99	0.4	100	(1.2, 23), 31	(3.9, 163)	(1.2, 23), 5	(1.2, 23)
ARPC4, F, 3–186	1u2vD, 137–279	29	14.3	38	(21.1, 84), 1	(23.0, 177)	(11.8, 46), 36	(11.8, 46)
ARPC5, G, 11–150	2p9nG, 11–150	94	5.5	88	(6.7, 117), 50	(6.7, 117)	(6.7, 117), 61	(12.6, 9)

[a] The percentage of sequence identity between the template and the component, as calculated from their alignment used for comparative modeling.
[b] The C^{α} RMSD and native overlap NO3.5 between the modeled component superposed on its native structure.
[c] The placement score and rank of the best placement, calculated by C^{α} RMSD to the reference. The placements were ranked by the normalized fitting score *C*.
[d] The placement score of the placement found in the top-ranking assembly configuration.

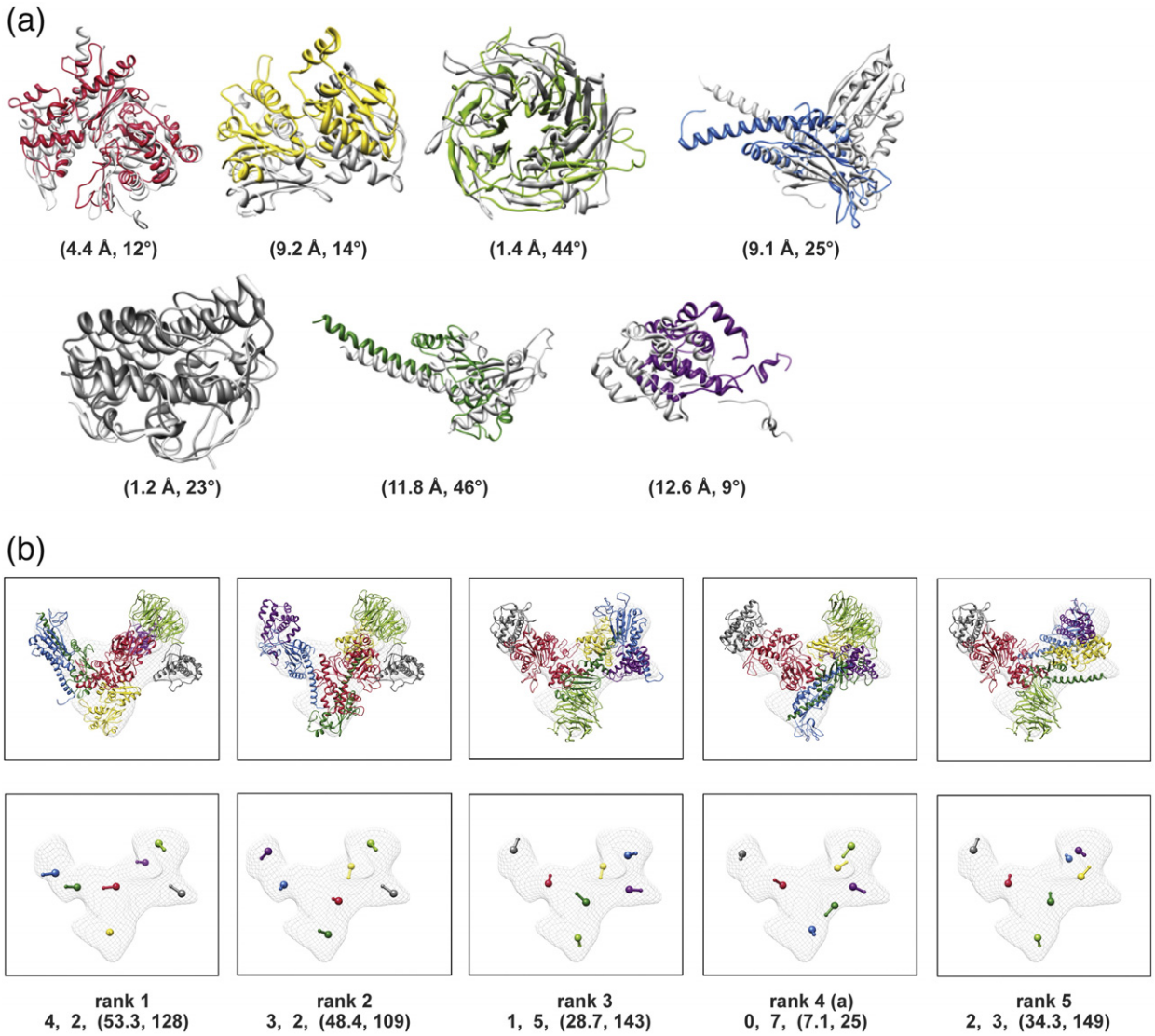

Fig. 3. MultiFit results for Arp2/3. (a) An assessment of the final model with a mapping score of 0. The model has the fourth smallest value of the scoring function S (the fourth model in (b)). The modeled (color) and reference placements (gray) of the individual components are compared (Results); the corresponding placement scores are indicated below each comparison. (b) Five top-ranked models for Arp2/3. The atomic representations of the models are displayed on the top row. The bottom row shows the centroid and the rotation axis for each component; the corresponding rank, mapping score, configuration score, and assembly placement score are indicated below each model.

99%: the C^{α} RMSD error for these models varied from 0.4 Å to 21.4 Å, and their native overlap varied between 38% and 100%; we intentionally used inaccurate comparative models to benchmark the robustness of our method with respect to errors in the component conformations.

In the final output of MultiFit, the near-native model with an assembly placement score of (7.1 Å, 25°) was ranked 4th among all the sampled configurations. In coarse-grained sampling, this model was ranked 10th, with a configuration score of 4/7 and an assembly placement score of (10.8 Å, 136°). The centroids of the individual components were positioned in the vicinity of their native centroids; however, the orientations of some components were incorrect, resulting in steric clashes between components. In fine-grained sampling, the 20 top-scoring models were refined. The refinement procedure was able to resolve many of the clashes in the model, which in turn improved its global score, resulting into the final rank of 4th. Next, we elaborate on the individual steps of the optimization protocol.

In anchor graph construction, we computed seven anchor points from the density map. The average distance between the anchor points and the centroids of the corresponding reference components was 7.2 Å. We then identified pairs of anchor points that are sufficiently close to allow components placed in their vicinity to interact with each other. The procedure pruned 12 of the possible 21 pairs (i.e., $7 \cdot 6/2$). The remaining 9 pairs allowed identification of 9 of the 12 native contacts between the 7 components.

In the discretization step of coarse-grained sampling, we fitted each component into the neighborhood of each anchor point using MODEM. We assessed the accuracy of the discretization by the placement score of the best placement of each component (i.e., the placement with the lowest C^{α} RMSD corresponding to the reference). These best placement scores ranged from (2.5 Å, 30°) to (23.0 Å, 177°). As expected, as the model accuracy measured by C^{α} RMSD and native overlap decreases, so does the rank of the best placement. The most accurate placement was ranked within the top 50 solutions for each component by the normalized fitting score *C*.

In the optimization step of coarse-grained sampling, we first represented the scoring function as a graphical model. The globally optimal component configuration was then found by a branch-and-bound search, in conjunction with the DOMINO optimizer. We utilized DOMINO for decomposing the simplified graphical model into an anchor junction tree of subsets of anchor points. The anchor junction tree contained four subsets of 2, 3, 3, and 3 anchor points. The branch-and-bound procedure resulted in 486 complete mappings for the 7 components (out of 7! = 5040 possible mappings). For each of these 486 mappings, the optimal placements of the 7 components were inferred by the gathering algorithm of DOMINO. A configuration with a mapping score of 0, a configuration score of 4/7, and an assembly placement score of (10.8 Å, 136°) was ranked 10th. The total running time with precomputed scoring terms was approximately 70 min on a single central processing unit; it takes approximately 2 h to precompute the scoring function terms.

This prediction demonstrates some of the benefits of and problems associated with coarse-grained sampling. For example, an accurate placement of Rpb2 and ARPC5 could not have been obtained solely based on the quality-of-fit due to nonnative conformations of their models (Table 1). Nevertheless, global optimization of the scoring function for the entire assembly did result in the correct placement for these two components. However, global optimization can also make a prediction less accurate. For example, ARPC4 was placed inaccurately because of the need for shape complementarity with inaccurately modeled neighbors Rpb1, ARPC1, ARPC2, and ARPC5. Such problems can be partly resolved by finer discretization of the sampling space (i.e., the fine-grained sampling; see the text below), in addition to flexible fitting (not attempted here).

In fine-grained sampling of a given model, we repopulated the sampling space for the corresponding complete mapping with pairwise docking solutions between the interacting components. Specifically, we enriched the set of placements by sampling binding modes of a component to the corresponding placed components of its neighboring anchor points using PATCHDOCK.[33] We then ran DOMINO again to find the optimally refined configuration. The assembly placement score of the refined configuration is (7.1 Å, 25°), which clearly demonstrates improvement in the accuracy of the relative orientations. For example, the placement accuracy of ARPC4 improved from (23.0 Å, 177°) to (11.8 Å, 46°). The improved placement was ranked only 499th in the pairwise docking between ARPC4 and ARPC1. However, global optimization relying on restraints derived from coarse-grained sampling (i.e., shape complementarity between interacting components and protrusion from the map envelope) resulted in this placement occurring in the best-scoring assembly configuration.

To validate the contribution of the shape complementarity score, we optimized a scoring function lacking this term (φ_3 in the scoring function *S*; Theory). The top-ranking configuration had a mapping score of 3, a configuration score of 3/7, and an assembly placement score of (42.5 Å, 94°). A model with a mapping score of 0 was not found in the top 50 solutions. This comparison demonstrates the positive contribution of the shape complementarity score to the accuracy of the generated assembly models.

Benchmark

To assess MultiFit more comprehensively than is possible by a single example, we also applied it to a benchmark that included five additional simulated test cases. In all six simulated tests, a model with a mapping score of 0 was found within the top four solutions (Table 2); in fact, the model with the mapping score of 0 was the best-scoring model in all cases for which the structures of the individual components were modeled based on templates with sequence identities higher than 60%. The assembly placement score of the model with the mapping score of 0 ranged between (2.6 Å, 4°) and (7.1 Å, 25°). These results demonstrate the utility of MultiFit in predicting the configuration of atomic components in a low-resolution density map of their assembly. Next, we report the benchmark results at each of the five steps of the algorithm.

In anchor graph construction, the average distance between the predicted anchor point and the centroid of the corresponding reference component in the near-native configuration was between 4 Å and 7 Å.

In the discretization step of coarse-grained sampling, a near-native configuration was sampled within the discrete sampling space in all test cases. However, this configuration was not necessarily ranked highly according to our scoring function due to steric clashes between interacting components.

In the optimization step of coarse-grained sampling, a model with a mapping score of 0 was found in the top 10 solutions in all test cases; in four of the six cases, it was the best-scoring solution. The assembly placement score of the model with a mapping score of 0 ranged from (2.6 Å, 4°) to (10.8 Å, 136°). The prediction accuracy depended on the

Table 2. Benchmark

Assembly		Component modeling			Coarse-grained sampling		Fine-grained sampling	
Assembly (name, PDB entry)	Number of components	% Sequence ID [average (range)][a]	C^{α} RMSD (Å) [average (range)][b]	NO3.5 (%) [average (range)][b]	Rank[c]	Optimization (assembly placement score)	Rank[c]	Optimization (assembly placement score)
Chaperonin GroEL, 1oel	Three domains	65 (60–72)	0.9 (0.2–1.0)	96 (92–100)	1	(2.6, 13)	1	(2.6, 13)
SUMO-RanGAP1-Ubc9-Nup358/RanBP2 complex, 1z5s	Four proteins	100 (100–100)	0.0 (0.0–0.0)	100 (100–100)	1	(5.9, 113)	1	(5.0, 67)
SUMO-RanGAP1-Ubc9-Nup358/AmBP2 complex, 1z5s	Four proteins	37 (18–56)	7.9 (1.2–15.0)	52 (13–98)	3	(7.7, 92)	3	(6.4, 62)
Dihydropyrimidine dehydrogenase, 1gte	Five domains	100 (100–100)	0.0 (0.0–0.0)	100 (100–100)	1	(2.6,4)	1	(2.6, 4)
Archaeon *Methanopyrus kandleri*, 1e6v	Six proteins	61 (57–68)	0.0 (0.0–0.0)	100 (100–100)	1	(2.5, 8)	1	(2.5, 8)
Arp2/3 complex, 1tyq	Seven proteins	51 (16–99)	7.9 (0.4–21.4)	70 (38–100)	10	(10.8, 136)	4	(7.1, 25)

[a] The average, minimum, and maximum percentages of sequence identity between the assembly components and their templates.
[b] The average, minimum, and maximum C^{α} RMSDs and native overlap NO3.5 between the modeled components superposed on their corresponding component in the native assembly.
[c] The rank of a model with a mapping score of 0.

component accuracy (Table 2). As the accuracy of the component models is decreased, the rank of the correct configuration and its placement score also become worse. The benchmark shows that coarse-grained sampling is able to determine component positions quite accurately, but frequently fails to result in accurate relative orientations. The main reason is the coarseness of the discrete sampling space, as demonstrated by the Arp2/3 and 1z5s examples. In the latter case, we obtained the near-native assembly [i.e., (5.9 Å, 113°)] with the native components and a less accurate configuration [i.e., (7.7 Å, 92°)] with distorted components.

In the discretization step of fine-grained sampling, the PATCHDOCK docking program[33] was able to sample near-native interaction modes between pairs of components. However, these interactions were generally not ranked highly by PATCHDOCK. For example, in the 1z5s case with distorted components, the most accurate docking prediction of chains C and D against chain A ranked 405th and 138th, respectively.

In the optimization step of fine-grained sampling, the refined models were at least as accurate as the most accurate models generated in coarse-grained sampling, sometimes much more so. In particular, the accuracy of the relative orientations between pairs of interacting components improved. For example, in the 1z5s case with distorted components, the assembly placement score improved from (7.7 Å, 92°) to (6.4 Å, 62°). The refined model contained placements derived from the docking prediction of chains C and D against chain A. These placements were ranked 405th and 138th by PATCHDOCK; reweighing the placements by the normalized fitting score C increased their ranks to 78th and 43rd, respectively. In the end, DOMINO correctly selected these placements for the final best-scoring configuration.

Benchmark with an experimentally determined map

To test the method in a realistic setting, we benchmarked it again by modeling the component configuration for an assembly with an experimentally determined cryoEM map.

GroEL–GroES domains

GroEL–GroES is a chaperonin that aids protein folding in *E. coli*. GroEL consists of two back-to-back rings of seven identical subunits, each of which contains three domains (i.e., the equatorial, intermediate, and apical domains). GroES is a ring of seven identical single-domain proteins that cap GroEL. We applied MultiFit to model the configuration of the four domains in an interacting pair of the GroEL and GroES subunits. Atomic coordinates for the four domains were obtained from a crystal structure of the GroEL–ADP–GroES complex (ADP state; PDB entry 1aon[47]). The corresponding density was segmented from a cryoEM map of the bacterial GroES–ADP7–GroEL–ATP7 chaperonin determined

at 23.5 Å resolution (ATP state; EMDB ID 1046[23]). The crystal structure of the ADP state was fitted to the density (as one rigid body) and used as reference for assessment. The main structural differences between the ATP and ADP states are the downward rotation of the intermediate domain and the counter-clockwise twist of the apical domain.[23]

The configuration with a mapping score 0 was ranked third, with an assembly placement score of (13.9 Å, 160°). A sampling space of approximately 14 million combinations was searched within 16 min of central processing unit time. The fine-grained sampling was able to generate a more accurate model with an assembly placement score of (11.0 Å, 84°). We note in passing that fitting all 49 domains (i.e., $3\times7\times2+7$) into the density of both rings would presumably benefit from the added information in the subunit–subunit interactions within and across rings; however, to test MultiFit in a more challenging setting, we deliberately modeled only a single symmetry unit consisting of three GroEL domains and one GroES domain.

Discussion

We described MultiFit, a computational method used for determining the positions and orientations (i.e., placements) of multiple atomic components in a cryoEM density map of their assembly. The problem is formulated in terms of combinatorial optimization, solved by our inferential optimizer DOMINO that guarantees the finding of the global minimum within a given discrete sampling space. The input is a density map and a set of atomic components that are kept rigid throughout the optimization process. For a given configuration of components, the scoring function measures the quality-of-fit of the atomic structures in the map, the protrusion from the map envelope, and the shape complementarity between pairs of components. The optimization process consists of coarse-grained and fine-grained sampling stages. Each sampling stage starts with a discretization step achieved, respectively, by fitting and docking, followed by an optimization step that relies on DOMINO. Both DOMINO and MultiFit are available as part of Integrative Modeling Platform‡.[48,49]

Accurate MultiFit predictions for seven test cases demonstrated its utility (Table 2). Specifically, our benchmark demonstrated the utility of MultiFit in predicting the configuration of components with known folds within a density map at resolutions between 20 Å and 23.5 Å; the average assembly placement score for the near-native configurations was (5.3 Å, 38°). MultiFit was able to determine the assembly configuration even in cases where the fitting scores were ambiguous. Examples include Arp2/3 (Table 1) and the 1z5s test case with distorted components (Table 2).

‡ http://salilab.org/imp

Next, we discuss (i) the benefits of simultaneous multiple component fitting, (ii) inaccuracies resulting from the discrete sampling space, and (iii) broad utility of combinatorial optimization.

Benefits of simultaneous fitting

Most methods for modeling assemblies in the context of a cryoEM map rely on a segmented assembly map and/or a model of the whole assembly. In the absence of such information, sampling the configuration space is computationally challenging, as the placement of each component may depend on the placements of other components. For example, the configuration of the Arp2/3 assembly with modeled components could not have been solved by iteratively fitting the largest remaining component in the unoccupied region using ModEM.[31] Moreover, the configuration cannot be modeled accurately without the component protrusion and the interaction terms in the scoring function used by MultiFit. However, by considering the placements of all components simultaneously, the protrusion of a component from the assembly envelope, and the shape complementary between the interacting components, we were able to determine the assembly configuration with an assembly placement score of (7.1 Å, 25°).

Inaccuracies resulting from discrete sampling space

MultiFit prediction will be accurate when a near-native configuration exists in the discrete sampling space and corresponds to the global minimum of the scoring function. These two conditions depend, in turn, on the accuracy of the atomic models of the individual components and the choice of anchor points. Next, we elaborate on these two dependencies.

Accuracy of component models

The atomic models of the individual components might be inaccurate due to modeling errors, induced fit, and conformational selection.[50] As the accuracy of the component models decreases, the discretized sampling space (either by fitting or by docking) is less likely to contain near-native placements (i.e., the sampling problem), and the global minimum is less likely to correspond to the most accurate sampled configuration (i.e., the scoring problem). In other words, these errors may affect the accuracy of the predicted assembly configuration due to scoring and sampling inaccuracies. One such example is the pair of 1z5s test cases (Table 2): The inputs to the first test case were the native components and the assembly density. The discretization steps of coarse-grained and fine-grained samplings resulted in near-native placements, and the top-ranked configuration detected by DOMINO had a relatively accurate assembly placement score of (5.0 Å, 67°). The inputs to the second test case were models with an average C^{α} RMSD error of 6.3 Å. The discrete sampling spaces

generated in the coarse-grained and fine-grained samplings contained less accurate placements. As a result, the utility of the scoring terms (especially the protrusion from the map envelope and the shape complementarity) decreased. The assembly placement score of the final assembly model with distorted component models was significantly worse (6.4 Å, 62°) than the assembly placement score of the assembly model with the native components. More accurate assembly models may be obtained by using a shape complementarity score that is less sensitive to component model errors and/or by an explicit treatment of the component conformations. To this end, techniques might be adopted from flexible fitting of a component into a density map[42,51] and from flexible molecular docking.[52,53]

Accuracy of anchor points

Given the k-cluster algorithm, the utility of the anchor points is affected by the variances in the size and shape of the components (data not shown). The utility of the anchor points is also affected by the resolution of the map (data not shown). To obtain a discrete sampling space that contains a near-native configuration, we sample candidate placements of each component in the neighborhood of each anchor point. However, there are many assemblies for which the variation in component sizes is too large for reasonable neighborhood sizes. We intend to improve the utility of anchor point calculation by considering component sizes and density map segmentation.[54,55]

Combinatorial optimization in structural biology

Modeling challenges in structural biology can generally be expressed as optimization problems.[48] These optimization problems often fall into a general class of NP-complete problems (Theory).[56] Combinatorial optimization is a type of optimization in which the set of feasible solutions is discrete, and the goal is to find the best possible solution within this discrete set. Combinatorial optimizers have been suggested for various modeling tasks such as side-chain packing,[57–59] threading,[28] ab initio RNA folding,[60] and prediction of quaternary structures of multiprotein complexes.[61] These methods can, in principle, be reformulated as a combinatorial optimization of a scoring function represented by a graphical model, benefiting from graph theory techniques.[24,25] Such a formulation has already been proposed for the side-chain packing problem.[59]

Our DOMINO method can, in principle, be applied to many problems in structural modeling, from low-resolution assembly modeling to side-chain refinement. Its strength derives from the junction tree algorithm that helps reduce the size of the search space from exponential in the number of components in the whole system to exponential in the number of components in the largest subset. More specifically, the computational complexity is $O(|U|L^s)$, where $|U|$ is the number of subsets in the junction tree, L is the size of the largest subset, and s is the number of discrete values of a single variable in the graphical model. Fortunately, at the granularity level used in MultiFit's application to protein assemblies in our benchmark, the theoretical complexity of the junction tree algorithm has not been a limiting factor. Nevertheless, in other applications that involve a dense graphical model of the scoring function and extensively sampled variable values, incomplete sampling of a discrete space may have to be accepted.

In conclusion, MultiFit and DOMINO can help bridge the gap between the atomic structures of the individual proteins and the cryoEM maps of their assemblies. In particular, they can provide initial configurations for further refinement of many multicomponent assembly structures described by electron microscopy.[42,51,62,63]

Acknowledgements

We thank Frank Alber for stimulating discussions, Ben Webb for help with the Integrative Modeling Platform software, and Dina Schneidman-Duhovny for help with the PATCHDOCK software. The research of K.L. was supported by a fellowship from the Edmond J. Safra Bioinformatics Program at Tel-Aviv University and the Clore Foundation Ph.D Scholars program and was carried out in partial fulfillment of the requirements for the Ph.D. degree at TAU. M.T. was funded by an MRC Career Development Award (G0600084). A.S. was supported by the Sandler Family Supporting Foundation, National Institutes of Health (R01 GM54762, U54 RR022220, PN2 EY016525, and R01 GM083960), National Science Foundation (IIS-0705196), Hewlett-Packard, NetApp, IBM, and Intel. H.J.W. acknowledges support by the Binational US–Israel Science Foundation, Israel Science Foundation (281/05) and the Hermann Minkowski-Minerva Center for Geometry at Tel Aviv University.

References

1. Robinson, C. V., Sali, A. & Baumeister, W. (2007). Molecular sociology of the cell. *Nature*, **450**, 973–982.
2. Gavin, A. C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M. *et al.* (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636.
3. Abbott, A. (2002). Proteomics: the society of proteins. *Nature*, **417**, 894–896.
4. Drenth, J. (1999). Principles of Protein X-ray Crystallography Springer Verlag, New York.
5. Frank, J. (2006). Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State, Oxford University Press, New York.
6. Chiu, W., Baker, M., Jiang, W., Dougherty, M. & Schmid, M. (2005). Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure*, **13**, 363–372.

7. Berman, H. M. (2008). The Protein Data Bank: a historical perspective. *Acta Crystallogr. Sect. A*, **64**, 88–95.
8. Henrick, K., Newman, R., Tagari, M. & Chagoyen, M. (2003). EMDep: a web-based system for the deposition and validation of high-resolution electron microscopy macromolecular structural information. *J. Struct. Biol.* **144**, 228–237.
9. Davis, J. A., Takagi, Y., Kornberg, R. D. & Asturias, F. A. (2002). Structure of the yeast RNA polymerase II holoenzyme: mediator conformation and polymerase interaction. *Mol. Cell*, **20**, 409–415.
10. Marlovits, T. C., Kubori, T., Lara-Tejero, M., Thomas, D., Unger, V. M. & Galan, J. E. (2006). Assembly of the inner rod determines needle length in the type III secretion injectisome. *Nature*, **441**, 637–640.
11. Mitra, K., Schaffitzel, C., Shaikh, T., Tama, F., Jenni, S., Brooks, C. L. *et al.* (2005). Structure of the *E. coli* protein-conducting channel bound to a translating ribosome. *Nature*, **438**, 318–324.
12. Schaffitzel, C., Oswald, M., Berger, I., Ishikawa, T., Abrahams, J. P., Koerten, H. K. *et al.* (2006). Structure of the *E. coli* signal recognition particle bound to a translating ribosome. *Nature*, **444**, 503–506.
13. Schmid, M. F., Sherman, M. B., Matsudaira, P. & Chiu, W. (2004). Structure of the acrosomal bundle. *Nature*, **431**, 104–107.
14. Chandramouli, P., Topf, M., Ménétret, J., Eswar, N., Cannone, J., Gutell, R. *et al.* (2008). Structure of the mammalian 80S ribosome at 8.7 Å resolution. *Structure*, **16**, 535–548.
15. Kostek, S., Grob, P., De Carlo, S., Lipscomb, J. S., Garczarek, F. & Nogales, E. (2006). Molecular architecture and conformational flexibility of human RNA polymerase II. *Structure*, **14**, 1691–1700.
16. Hainfeld, J. & Powell, R. (2000). New frontiers in gold labeling. *J. Histochem. Cytochem.* **48**, 471–480.
17. Rossmann, M. G., Bernal, R. & Pletnev, S. V. (2001). Combining electron microscopic with X-ray crystallographic structures. *J. Struct. Biol.* **136**, 190–200.
18. Goddard, T., Huang, C. & Ferrin, T. (2007). Visualizing density maps with UCSF Chimera. *J. Struct. Biol.* **157**, 281–287.
19. Ceulemans, H. & Russell, R. B. (2004). Fast fitting of atomic structures to low-resolution electron density maps by surface overlap maximization. *J. Mol. Biol.* **338**, 783–793.
20. Fabiola, F. & Chapman, M. S. (2005). Fitting of high-resolution structures into electron microscopy reconstruction images. *Structure*, **13**, 389–400.
21. Baumeister, W., Walz, J., Zühl, F. & Seemüller, E. (1998). The proteasome: paradigm of a self-compartmentalizing protease. *Cell*, **93**, 367–380.
22. Serysheva, I., Ludtke, S., Baker, M., Cong, Y., Topf, M., Eramian, D. *et al.* (2008). Subnanometer-resolution electron cryomicroscopy-based domain models for the cytoplasmic region of skeletal muscle RyR channel. *Proc. Natl Acad. Sci. USA*, **105**, 9610–9615.
23. Ranson, N. A., Farr, G. W., Roseman, A. M., Gowen, B., Fenton, W. A., Horwich, A. L. & Saibil, H. R. (2001). ATP-bound states of GroEL captured by cryo-electron microscopy. *Cell*, **107**, 869–879.
24. Jordan, M. I. (2004). Graphical models. *Stat. Sci.* **19**, 140–155.
25. Lauritzen, S. (1996). Graphical Models Oxford University Press, New York, NY.
26. Shimony, S. E. (1994). Finding MAPs for belief networks is NP-hard. *Artif. Intell.* **68**, 399–410.
27. Pearl, J. (1998). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers Inc., San Francisco.
28. Xu, J., Jiao, F. & Berger, B. (2005). A tree-decomposition approach to protein structure prediction. *Proc. IEEE Comput. Syst. Bioinf. Conf.*, 247–256.
29. Andersen, S., Olesen, K. & Jensen, F. (1990). HUGIN—a shell for building Bayesian belief universes for expert systems. pp. 332–337, Morgan Kaufmann Publishers Inc., San Francisco.
30. Krogan, N., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A. *et al.* (2006). Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, **440**, 637–643.
31. Topf, M., Baker, M. L., John, B., Chiu, W. & Sali, A. (2005). Structural characterization of components of protein assemblies by comparative modeling and electron cryo-microscopy. *J. Struct. Biol.* **149**, 191–203.
32. Lasker, K., Dror, O., Shatsky, M., Nussinov, R. & Wolfson, H. J. (2007). EMatch: discovery of high resolution structural homologues of protein domains in intermediate resolution cryo-EM maps. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **4**, 28–39.
33. Duhovny, D., Nussinov, R. & Wolfson, H. J. (2002). Efficient Unbound Docking of Rigid Molecules. *WABI 2002, Springer Lecture Notes in Computer Science*, **2452**, 185–200.
34. Chen, R. & Weng, Z. (2003). A novel shape complementarirty scoring function for protein–protein docking. *Proteins*, **51**, 397–408.
35. Wolfson, H. & Rigoutsos, I. (1997). Geometric hashing: an overview. *IEEE Comput. Sci. Eng.* **11**, 263–278.
36. Lamdan, Y. & Wolfson, H. J. (1988). Geometric hashing: a general and efficient model-based recognition scheme. Proceedings of the International Conference on Computer Vision, 2nd Int. Conf. on Computer Vision (ICCV), pp. 238–249, IEEE Computer Society Press, Los Alamitos, CA.
37. Connolly, M. (1983). Analytical molecular surface calculation. *J. Appl. Crystallogr.* **16**, 548–558.
38. Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C.D., Silverman, R. & Wu, A. Y. (2002). An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 881–892.
39. Wriggers, W., Milligan, R. A., Schulten, K. & McCammon, J. A. (1998). Self-organizing neural networks bridge the biomolecular resolution gap. *J. Mol. Biol.* **284**, 1247–1254.
40. Wriggers, W., Milligan, R. A. & McCammon, J. A. (1999). Situs: a package for docking crystal structures into low-resolution maps from electron microscopy. *J. Struct. Biol.* **125**, 185–195.
41. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H. *et al.* (2000). The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–242.
42. Sali, A. & Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779–815.
43. Topf, M., Lasker, K., Webb, B., Wolfson, H. J., Chiu, W. & Sali, A. (2008). Protein structure fitting and refinement guided by cryoEM density. *Structure*, **16**, 295–307.
44. Cohen, F. & Sternberg, M. (1980). On the prediction of protein structure: the significance of the root-mean-square deviation. *J. Mol. Biol.* **138**, 321–333.
45. Goley, E. D. & Welch, M. D. (2006). The ARP2/3 complex: an actin nucleator comes of age. *Nat. Rev. Mol. Cell Biol.* **7**, 713–726.
46. Nolen, B. J., Littlefield, R. S. & Pollard, T. D. (2004). Crystal structures of actin-related protein 2/3

complex with bound ATP or ADP. *Proc. Natl Acad. Sci. USA*, **101**, 15627–15632.
47. Xu, Z., Horwich, A. L. & Sigler, P. B. (1997). The crystal structure of the asymmetric GroEL–GroES–(ADP)7 chaperonin complex. *Nature*, **388**, 741–750.
48. Alber, F., Forster, F., Korkin, D., Topf, M. & Sali, A. (2008). Integrating diverse data for structure determination of macromolecular assemblies. *Annu. Rev. Biochem.* 77.
49. Alber, F., Dokudovskaya, S., Veenhoff, L., Zhang, W., Kipper, J., Devos, D. *et al.* (2007). Determining the architectures of macromolecular assemblies. *Nature*, **450**, 683–694.
50. Boehr, D. D. & Wright, P. E. (2008). How do proteins interact? *Science*, **320**, 1429–1430.
51. Schröder, G., Brunger, A. & Levitt, M. (2007). Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure*, **15**, 1630–1641.
52. Bahar, I. & Rader, A. (2005). Coarse-grained normal mode analysis in structural biology. *Curr. Opin. Struct. Biol.* **15**, 586–592.
53. Bonvin, A. (2006). Flexible protein–protein docking. *Curr. Opin. Struct. Biol.* **16**, 194–200.
54. Birmanns, S. & Wriggers, W. (2007). Multi-resolution anchor-point registration of biomolecular assemblies and their components. *J. Struct. Biol.* **157**, 271–280.
55. Kawabata, T. (2008). Multiple subunit fitting into a low-resolution density map of a macromolecular complex using a Gaussian mixture model. *Biophys. J.* **95**, 4643–4658.
56. Wales, D. & Scheraga, H. (1999). Global optimization of clusters, crystals, and biomolecules. *Science*, **285**, 1368–1372.
57. Canutescu, A., Shelenkov, A. & Dunbrack, R. J. (2003). A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.* **12**, 2001–2014.
58. Xu, J. & Berger, B. (2006). Fast and accurate algorithms for protein side-chain packing. *J. ACM*, **53**, 533–557.
59. Yanover, C., Schueler-Furman, O. & Weiss, Y. (2008). Minimizing and learning energy functions for side-chain prediction. *J. Comput. Biol.* **15**, 899–911.
60. Zhao, J., Malmberg, R. L. & Cai, L. (2008). Rapid ab initio prediction of RNA pseudoknots via graph tree decomposition. *J. Math. Biol.* **56**, 145–159.
61. Inbar, Y., Benyamini, H., Nussinov, R. & Wolfson, H. J. (2005). Prediction of multimolecular assemblies by multiple dockin. *J. Mol. Biol.* **349**, 435–447.
62. Trabuco, L., Villa, E., Mitra, K., Frank, J. & Schulten, K. (2008). Flexible fitting of atomic structures into electron microscopy maps using molecular dynamics. *Structure*, **16**, 673–683.
63. Orzechowski, M. & Tama, F. (2008). Flexible fitting of high-resolution X-ray structures into cryoelectron microscopy maps using biased molecular dynamics simulations. *Biophys. J.* **95**, 5692–5705.