

Macromolecular assembly structures by comparative modeling and electron microscopy

Keren Lasker^{1,2,+,*}, Javier A. Velázquez-Muriel^{1,+}, Benjamin M. Webb¹, Zheng Yang³, Thomas E.
Ferrin³, and Andrej Sali^{1*}

¹Department of Bioengineering and Therapeutic Sciences, Department of Pharmaceutical Chemistry, and California Institute for Quantitative Biosciences (QB3), University of California, San Francisco, 1700 4th Street, San Francisco, CA 94158-2330, USA.

²Blavatnik School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel-Aviv 69978, Israel.

³ Resource for Biocomputing, Visualization, and Informatics, Department of Pharmaceutical Chemistry, University of California, San Francisco, 600 16th Street, San Francisco, CA 94143-2240, USA.

+ Equal contribution

*Corresponding authors:

Andrej Sali, Email: sali@salilab.org

Keren Lasker, Email: kerenl@salilab.org

Running title: Comparative modeling guided by EM

Total number of pages, figures, tables: 24, 4, 0

Submitted to: *Methods in Molecular Biology*

Date: February 5, 2011

Abstract

Advances in electron microscopy allow for structure determination of large biological machines at increasingly higher resolutions. A key step in this process is fitting component structures into the electron microscopy-derived density map of their assembly. Comparative modeling can contribute by providing atomic models of the components, *via* fold assignment, sequence-structure alignment, model building, and model assessment. All four stages of comparative modeling can also benefit from consideration of the density map. In this chapter, we describe numerous types of modeling problems restrained by a density map and available protocols for finding solutions. In particular we provide detailed instructions for density map-guided modeling using Integrative Modeling Platform (IMP), MODELLER, and UCSF Chimera.

Key words: macromolecular complexes, electron microscopy, fitting, homology modeling, comparative modeling, integrative modeling, visualization, Chimera, MODELLER, IMP.

1. Introduction

Structural description of macromolecular complexes is required for studying their assembly, function, and evolution (1, 2). Although numerous assembly structures have been determined by X-ray crystallography (3) and NMR spectroscopy (4, 5), these techniques are not always applicable. Recent advances established electron microscopy (EM) as a central technique for studying the structures of macromolecular assemblies in different functional states *in vitro* and *in vivo*. EM approaches include electron crystallography, single-particle EM, and electron tomography (6-8). EM generally produces a three-dimensional (3D) grid specifying the observed electron density of the system (*i.e.*, the density map). The resolution of this map is typically better than 25 Å, and can be as high as approximately 4 Å for highly symmetric structures (9, 10). In most cases, however, the resolution of a density map is insufficient to provide a full atomic description of a protein complex. To this end, computational integration of atomic resolution structures with EM density maps is essential. In particular, the resolution of the density map is often adequate for accurate rigid fitting of atomic structures of the subunits into the density map, resulting in an atomic model of the entire assembly (11-22). Given sufficient resolution, flexible fitting can be used to further refine the model by fitting into the density map while maintaining correct stereochemistry (23-27).

A key requirement for such density-guided structural modeling techniques is the availability of atomic resolution structures of the assembly components. However, these structures are frequently not available from X-ray crystallography or NMR spectroscopy. Fortunately, it may be possible to construct useful component models by comparative (homology) modeling.

Comparative modeling techniques are routinely used to model the structure of a given protein sequence (target) based primarily on its alignment to one or more proteins of known structure (templates) (28-30). The target structure is predicted by identifying one or more related proteins

of known structure, aligning the target sequence to the template structures, building a model, and assessing it. Comparative modeling approaches have become frequently applicable in part due to the success of structural genomics initiatives that aim to solve representative structures of most protein families by X-ray crystallography or NMR spectroscopy, such that most of the remaining proteins can be modeled with useful accuracy based on their similarity to a known structure. In fact, at least two orders of magnitude more sequences can be modeled by comparative modeling than have been determined by experiment (31). Therefore, methods for improving fitting into a density map by considering errors in comparative models have been developed (19, 32, 33). Moreover, the availability of a density map opens a possibility of improving the corresponding comparative model, by helping with fold assignment, sequence-structure alignment, model building, and model assessment (14, 20, 22, 34).

In this chapter, we describe various types of density-guided modeling problems and available solutions within Integrative Modeling Platform (IMP) (35), MODELLER (28), and UCSF Chimera visualization software (36). This description is followed by a Notes section that highlights several practical issues in density-guided modeling.

2. Materials

To follow the examples, IMP, MODELLER, Chimera, and a set of input files are required. The IMP software can be downloaded from <http://salilab.org/imp/download.html>, MODELLER from <http://salilab.org/modeller>, and Chimera from <http://www.cgl.ucsf.edu/chimera>. All programs are available in a binary form for most common machine types and operating systems. IMP can also be rebuilt from the source code. The example files are found in `biological_systems/groel` directory in IMP.

3. Methods

Selecting a protocol for density-guided structural modeling depends on the resolution of the

density map and the available atomic information. Interpretation of the density map usually begins by identifying the different structural units (*e.g.*, entire protein chains, domains, secondary structure elements, or nucleic acids) by means of segmentation techniques (6, 37). Independently, the availability of atomic structures of the components is determined; when necessary, comparative models are built (29, 38), if a template can be found. Then, an appropriate integrative modeling protocol is selected (Figure 1).

We describe in detail the modeling of the bacterial molecular chaperone GroEL (39, 40, 41). GroEL promotes protein folding in bacterial cells in conjunction with its lid-like co-chaperonin protein complex GroES. GroEL is composed of two heptameric rings of identical 57 kDa subunits stacked back-to-back. The GroEL structure was extensively studied by X-ray crystallography (42-44) and EM (45-48) across different species, and thus provides a good illustration of approaches that integrate EM data into assembly modeling (49).

The inputs for the GroEL example (Figure 2) are the sequence of the *E. coli* GroEL chaperone monomeric unit (UniProt id: P0A6F5, file: *data/sequences/groel_ecoli.ali*) and an EM density map of the naked groEL at 11.5 Å resolution (45) (EMDB id: 1081, file: *data/em_maps/groel-11.5A.mrc*) consisting of 14 subunits. We start by searching for known structures homologous to the GroEL monomeric unit (Sections 3.1 and 3.2) and independently segment the density map (Section 3.3). We then use the density map to assess the choice of the template(s) (Section 3.4). Next, we build a comparative model of the GroEL monomeric unit based on the selected template(s) (Sections 3.5 and 3.6) and model the entire GroEL complex by simultaneously fitting 14 rigid copies of the monomer model into the complete density map (Section 3.7). Finally, we improve the accuracy of the model by refining it to better fit into its density map (Section 3.8).

3.1. Template identification

Template identification is achieved by scanning the sequence of a monomeric unit of the GroEL against a library of sequences for the known protein structures in the Protein Data Bank (PDB)

(www.pdb.org, (**50**)). We use the *profile.build()* command of MODELLER. The *profile.build()* algorithm uses a local dynamic programming procedure to identify templates with sequences related to the target. In the simplest case, *profile.build()* takes as input the target sequence (file: *data/sequences/groel_ecoli.ali*) and a database of sequences with known structures (file: *data/datasets/pdb_95.pir*), and returns a set of statistically significant alignments (file: *build_profile.prf*). The script and further details can be found in file *scripts/script1_build_profile.py* and **Note 1**.

3.2. Template(s) selection by sequence

Selection of candidate template(s) from known structures found to be homologous to the target is generally a subjective process. Frequently, the selected template(s) share the highest sequence identity to the target. However, additional assessment may be used; in Section 3.4, we demonstrate the use of an EM density map for selecting the most appropriate templates.

The output file *build_profile.prf* (see **Note 2**) identifies 13 potential templates, all with high confidence according to their E-values, some covering the entire target sequence and others only parts of it. We remove structures matching only a fraction of the target sequence (PDB codes: 1dk7A, 1kidA, 1la1A, and 1srvA), as there is a sufficient number of templates with high confidence covering the entire sequence. To analyze the relationships between the 9 remaining structures, we use the *alignment.compare_structures()* command in MODELLER to assess structural and sequence similarity between the structures. This command compares the structures according to the alignment constructed by the *malign3d()* command and produces a clustering tree from the input matrix of pairwise C α root mean standard deviation (RMSD) distances, helping to visualize differences among the template candidates. The script and further details can be found in file *script2_compare_templates.py* and **Notes 2-3**.

3.3. Density map segmentation

Interpretable structural features depend on the resolution of the map and their size. At low resolutions (20-25 Å), the overall shape of the assembly and boundaries of sub-complexes or large proteins can be detected. As the resolution improves, boundaries of smaller proteins or domains can be identified (51-53). At a medium resolution (6-10 Å), secondary structure elements are apparent (37). At a higher resolution, backbone tracing and even side chain conformation may be possible to define (54). Segmentation is in many cases performed in a semi-manually manner using visualization tools such as Chimera (21), Amira (<http://www.amira.com>), Gorgon (<http://gorgon.wustl.edu>), and Sculptor (<http://sculptor.biomachina.org>). Notably, a watershed segmentation procedure has been integrated into Chimera (52); secondary structures segmentation and annotation can be performed *via* the Gorgon visualization software.

Here, we apply a Gaussian mixture model-based segmentation of the density map into 14 regions using the *IMP.multifit.density2anchors* program (55). The resulting segmented regions correspond to the density regions occupied by the subunits. A complete list of commands and further details can be found in file *script3_density_segmentation.py* and **Notes 4-5**.

3.4. Template selection by fitting to a density map

The density map of the target can aid the process of template selection, by assessing the optimal overlap between a template structure and the density map (14, 19, 20, 34, 56). Such assessment is particularly useful when the templates do not share high sequence similarity with the target or when the conformations of the target and template structures differ (Section 3.6). We score the 9 remaining candidate templates by fitting each of them into the density map and reporting the EM quality-of-fit score (*see Note 6*) (25). The score ranges from 0 to 1, with 0 indicting a perfect fit. Here, the density map is a segmented region corresponding to a monomeric subunit of the GroEL complex density map (file: *groel_subunit_11.mrc*).

Fitting of a component structure into the density map usually optimizes a similarity score between

the component and the density map (*e.g.*, the cross-correlation coefficient) as a function of the component's translation and rotation relative to the density map (rigid fitting) (**49, 57**). IMP provides four different methods for performing rigid fitting, based on: (i) anchor points matching by geometric hashing (*IMP.multifit.anchor_points_based_rigid_fitting()*) (**55**), (ii) fast Fourier transform (**58**) (*IMP.multifit.fft_based_rigid_fitting()*), (iii) principal component analysis (PCA) (**55**) (*IMP.multifit.pca_based_rigid_fitting()*), and (iv) local Monte Carlo/conjugate gradient search (**25**) (*IMP.em.local_rigid_fitting()*). Here, we read the profile output into IMP and fit each of the candidate templates into the density map, employing the PCA-based fitting, followed by a local fitting (*see* **Notes 8-9**). The resulting quality-of-fit scores range from 0.18 to 0.33, indicating that despite the high sequence identity of the target sequence to some of the structures (60% for 1sjpA; 63% for 1we3A), the target structure is in a different conformational state than the templates. Interestingly, some templates with high quality-of-fit scores had lower sequence identity than templates with high sequence identity (*e.g.*, 3kfeA with 27% sequence identity and EM quality-of-fit of 0.3 *versus* 1we3A with 63% sequence identity and EM quality-of-fit of 0.32), illustrating the potential utility of a density map for improving comparative models. To exemplify advanced flexible fitting techniques, we chose 1iokA as the template. The script and further details can be found in file `scripts/script4_score_templates_by_cc.py`, **Notes 6-9**, and Figures 2,3.

3.5. Template alignment

Once template(s) have been selected, the next step of a comparative modeling procedure is aligning the chosen template(s) to the target sequence. Here, sequence-structure alignments are calculated using the *align2d()* command of MODELLER (**59**). Although *align2d()* relies on a global dynamic programming algorithm (**60**), it is different from standard sequence-sequence alignment methods because it incorporates structural information from the template when constructing the alignment. This goal is achieved through a variable gap penalty function that

tends to place gaps in solvent exposed and curved regions, outside secondary structure segments, and between two positions that are close in space (**61**). The resulting alignment is written into the file *groel-liokA.ali* in the PIR format. The script and further details can be found in file *scripts/script5_template_alignment.py*.

In addition, templates and their alignments to the target sequence can be explored using UCSF Chimera. Chimera uses BLAST to search the PDB for potential templates, which are displayed in the Multalign Viewer tool (Figure 4, top) (**62**). The Viewer allows for alignment editing, for example to remove gaps within an element of regular secondary structure in the template, which frequently contribute to model error. Additional sequences can be added to the alignment, either by typing or extracting from other structures in Chimera.

3.6. Modeling building and assessment

We perform automated comparative model building using the *automodel()* command in MODELLER, generating 10 comparative models based on the input target-template alignment (file: *scripts/script6_model_building_and_assessment.py*). Comparison between these 10 models reveals structural differences (C α RMSD between pairs of models range from 4.6 Å to 8.2 Å, file: *scripts/script7_pairwise_rmsd.py*). To select the most accurate model, we assess the quality of the models according to the normalized Discrete Optimized Protein Energy (zDOPE, *see Note 10*) (**63**), TSVMMod (**64**), and the EM quality-of-fit (**25**) scores. We remove the c-terminus region of each model (residues 524 to 548) prior to the assessment procedure, as it was not covered by the template. The first assessment measure is the normalized DOPE score (MODELLER command *assess_normalized_dope()*); a value of less than -1 indicates that the distribution of atom pair distances in the model resembles that found in a large sample of known protein structures. The model with the minimum zDOPE score value is model 1 (score of 0.19). However, none of the truncated models got a zDOPE score lower than -0.06, despite the relatively low zDOPE score of the template (-0.6), indicating inaccuracies in the modeling

procedure and/or an unusually unfavorable zDOPE score value of the (correct) template structure (see **Note 11**). The second assessment measure is the TSVMMod score that predicts the native overlap (defined as the fraction of C α atoms within 3.5 Å of the native structure) of a comparative model in the absence of a solved structure using support vector machine learning (**64**) (<http://modbase.compbio.ucsf.edu/evaluation>). The predicted C α RMSD errors are between 5.3 and 8.6 Å for the full models and 3.4 to 3.9 for the truncated models (file: *tsvmod.server.results.txt*). The third assessment measure is the EM quality-of-fit score that measures the fit of a model to the density map. All 10 truncated models got comparable scores around 0.2. As according to these criteria all models are of comparable accuracy, we selected model 1 as the starting model for refinement because it scored the best according to zDOPE and EM quality-of-fit scores. A complete list of commands and further details can be found in *scripts/script6_model_building_and_assessment.py*, *scripts/script7_pairwise_rmsd.py*, and **Notes 10-11**.

Alternatively, MODELLER can be called from within Chimera, either as a process run on the user's computer or as a process run remotely via a web service. From the Chimera-MODELLER interface, the user can choose the target sequence, template structure(s), and specify advanced options, e.g. number of output models (Figure 4, middle left). If the user chooses to run MODELER locally, the MODELLER script file generated by Chimera is accessible and customizable. The MODELLER modeling process is run in the background and can be monitored via Chimera's task manager. For the single chain of GroEL, it took about 20 minutes running via the web service to generate 10 models. When the results become available, the models are displayed in Chimera and their scores shown in a table (Figure 4, bottom left). The results table lists the GA341 (**65**), zDOPE and DOPE scores. Clicking the *Fetch Scores* option, triggers a call to TSVMMod for calculating estimated RMSD and overlaps.

3.7. Multiple fitting into a density map

So far we have modeled the structure of the monomeric unit. However, the density map was determined for the entire complex. As a template of the entire complex is not known (for the purpose of this example), we model the whole assembly by fitting 14 copies of the monomeric unit model into the map. We use the symmetric version of the MultiFit program designed to efficiently sample ring complexes. We first split the density into two rings long the Z axis (file: *scripts/script8_split_density.py*). We then run MultiFit separately for each ring (file: *scripts/script9_symmetric_multiple_fitting.py*). The procedure outputs a list of assembly models ranked by their EM quality-of-fit score (files: *multifit.top.output* and *multifit.bottom.output*, see **Note 13**). The two top ranking models, one from each ring (files: *model.top.0.pdb* and *model.bottom.0.pdb*), are joined to create a complete model of the assembly with an EM quality-of-fit score of 0.08. A complete list of commands and further details can be found in *scripts/script8_split_map.py*, *scripts/script9_symmetric_multiple_fitting.py*, and **Notes 12-13**.

Alternatively, MultiFit can be called from within Chimera, either as process run on the user's computer or run remotely via a web service. From the Chimera-MultiFit interface, user can choose the monomeric unit model, EM density map and specify the map resolution. When MultiFit finishes its calculation in the background, the solutions are displayed and their geometric complementarity scores and EM quality-of-fit scores are shown in a table.

3.8. Flexible fitting into a density map

The comparative model generated for the monomeric subunit of GroEL complex is in a different conformational state than the one determined by EM, as indicated by the EM quality-of-fit score (0.2). Conformational differences between a comparative model and its density map can originate from different conditions (*e.g.*, crystallization *versus* freezing) under which the isolated components and assembly structures were determined, as well as errors in modeling methods (such as mis-assignment of secondary structure elements and their shifts in space caused by

target-template misalignment). Flexible fitting can help by refining the conformation of the component, together with its position and orientation. Here, we use the FlexEM method in MODELLER (25) for refining the model to better fit its density. The procedure first adjusts the positions and orientations of its secondary structure segments followed by a full atomic refinement. The increased accuracy of the model is reflected by the EM quality-of-fit score that improved from 0.43 to 0.36. A complete list of commands and further details can be found in file *scripts/script10_flexible_fitting.py* and **Notes 14-15**.

4. Conclusions

EM techniques are becoming increasingly useful for structural characterization of macromolecular assemblies (66). In most cases, however, the resolution of a density map is insufficient to provide a complete atomic description of a protein complex with high confidence. To this end, computational integration of atomic resolution structures with EM density maps is essential. Here, we demonstrate how MODELLER, IMP and Chimera can be used for modeling structures of such assemblies by a combination of homology modeling, fitting, and refinement techniques. These steps are now combined within the Chimera software allowing the user to visualize and control the modeling process. We expect such integrative modeling protocols to become increasingly useful and facilitate maximizing the coverage, accuracy, resolution and efficiency of the structural characterization of macromolecular assemblies.

5. Notes

1. Below we provide a detailed description of *script1_build_profile.py*:
 - *log.verbose()* sets the amount of information that is written out to the log file.

- *environ()* initializes the 'environment' for the current modeling procedure, by creating a new environ object, called env. Almost all MODELLER scripts require this step, as the *environ()* object is needed to build most other objects.
- *sequence_db()* creates a sequence database object, calling it sdb, which is used to contain large databases of protein sequences.
- *sdb.read()* reads a text file ,containing non-redundant PDB sequences, into the sdb database.
The input options to this command specify the name of the database (seq_database_file: 'pdb_95.pir'), the format (seq_database_format='pir'), whether to read all sequences from the file (chains_list='all'), upper and lower bounds for the lengths of the sequences to be read (minmax_db_seq_len=(30,4000)), and whether to remove non-standard residues from the sequences (clean_sequences=True).
- *sdb.write()* writes a binary machine-independent file (seq_database_format='binary') with the specified name (seq_database_file:'pdb_95.bin'), containing all sequences read in the previous step.
- The second call to *sdb.read()* reads the binary format file back in for faster execution.
- *alignment()* creates a new 'alignment' object (aln).
- *aln.append()* reads the target sequence groel from the file groel.ali and *aln.to_profile()* converts it to a profile object (prf). Profiles contain similar information as alignments, but are more compact and better suited for sequence database searching.
- *prf.build()* searches the sequence database (sdb) using the target profile stored in the prf object as the query. Several options, such as the parameters for the alignment algorithm (matrix_offset, rr_file, gap_penalties etc.), are specified to override the default settings. max_aln_evalue specifies the threshold value to use when reporting statistically significant alignments.

- *prf.write()* writes a new profile containing the target sequence and its homologs into the specified output file (file:*build_profile.prf*).
- The profile is converted back to the standard alignment format and written out using *aln.write()*.

2. The results of the *build_profile()* command are stored in the output file *output/build_profile.prf*. The first six lines of this file list the input parameters used to create the alignments between the identified templates and the target sequence. Subsequent lines contain several columns of data, one for each template. For the purposes of this example, the relevant columns are (i) the second column, containing the PDB code of the related template sequences; (ii) the tenth column, indicating length of the matched alignment between the GroEL subunit and the template; (iii) the eleventh column, containing the percentage sequence identity of the alignment; and (iv) the twelfth column, containing E-values for the statistical significance of the alignments.

3. After a list of all related protein structures and their alignments with the target sequence has been obtained, template structures are usually prioritized depending on the purpose of the comparative model. Template structures may be chosen based purely on the target-template sequence identity or a combination of several other criteria, such as the experimental accuracy of the structures (resolution of x-ray structures, number of restraints per residue for NMR structures), conservation of active-site residues, holo-structures that have bound ligands of interest, and fit to other experimental data such as density maps and small angle X-ray scattering curves (67).

4. A segmentation of the EM density map is performed by an adaptation of the Gaussian mixture model (GMM) clustering technique (55, 68). Geometrically, an assembly of globular proteins can be viewed as a spatial configuration of ellipsoidal components. Each such component can be approximated by a 3D Gaussian, represented by a 3D mean (*i.e.*, its centroid) and a 3D variance (*i.e.*, the square lengths of its principal axes). Thus, a segmentation of an

assembly density that corresponds to its molecular configuration can be formulated as finding the most likely linear combination of Gaussian components from which the assembly density was sampled.

5. The script *script3_density_segmentation.py* sets a call to the *IMP.multifit.density2anchors* program; for more options, call the executable directly. *density2anchors* requires specifying of the number of Gaussians (K). It is recommended to set K to the number of proteins (domains) of the assembly for segmenting a low-resolution (an intermediate resolution) density map, however different K s should be tested. To visually inspect of segmentation results, add the *seg* option to *density2anchors* run; with this option *density2anchors* writes each segment into a separate MRC file and provides a *load_configurations.cmd* script to load all segments into Chimera.

6. The EM quality-of-fit of a probe (ρ^P) to its density (ρ^{EM}) is defined as 1 minus the cross-correlation coefficient (CCF) between them. Specifically, CCF is defined as:

$$CCF = \frac{\sum_{i \in Vox(\rho^P)} \rho_i^{EM} \left(\sum_{j=1}^N \rho_{i,j}^P \right)}{\sqrt{\sum_{i \in Vox(\rho^P)} \left(\rho_i^{EM} \right)^2 \sum_{i \in Vox(\rho^P)} \left(\sum_{j=1}^N \rho_{i,j}^P \right)^2}}, \text{ where } Vox(\rho^P) \text{ represents all voxels in the density}$$

grid that are within two times the map resolution from any of the atoms of the protein; and where

the total density of P at grid point i is $\sum_{j=1}^N \rho_{i,j}^P$. The values of the EM quality-of-fit score range

from 0 to 1, where 0 indicates a perfect fit.

7. Below we highlight key commands in *script4_score_templates_by_cc.py*:

- First few lines parse the *build_profile.prf* file and extract the names of the templates.
- *IMP.em.read_map()* reads the density map. The command gets as input a density map filename and an appropriate reader, which is in this case a *MRCReaderWriter*. IMP supports MRC, Xplor, Spider, and EM formats.

- The resolution of the density map is not saved in the map, and needs to be set using the *set_resolution()* command.
 - *IMP.Model()* initializes an IMP model, which is going to store all templates.
 - *IMP.atom.read_pdb()* reads the structure of the template. The function requires a file in PDB format, and a model object that is going to store the molecule. In addition, the function can get as input a *Selector* that specify which atom types are should be read (eg *CAlphaPDBSelector* and *NonWaterPDBSelector*).
 - *IMP.atom.setup_as_rigid_body()* sets the molecule to a rigid body. The function returns a *IMP.core.RigidBody* decorator. To learn more on the decorator concept in IMP see <http://salilab.org/imp>.
 - The rigid fitting procedure is performed in two stages. First, coarse fits are explored using the *IMP.multifit.pca_based_rigid_fitting()* command. These fits are then refined by a local Monte Carlo/conjugate gradient (MC/CG) minimization using the *IMP.em.local_rigid_fitting()* command.
 - We write the fitted templates using the *IMP.atom.write_pdb()* command. Notice that we used *IMP.core.transform()* to transform the rigid body to its fitted position prior to the writing command.
8. The *IMP.multifit.pca_based_rigid_fitting()* command fits a protein to its density map by aligning their principal components. The principle components of the density are calculated according to all voxels above a density threshold (specified by the user) while the principle components of the density map are calculated according to all atoms. The function returns a list of fits. Each fit is represented by a transformations and a quality-of-fit score.
9. The *IMP.em.local_rigid_fitting()* command locally refines the current fit of a rigid body in a density map by a local MC/CG sampling. At each MC iteration of the rigid body is randomly

locally transformed followed by a CG minimization. The user can specify the number of MC iterations and the maximum number of CG steps allowed at each iteration.

10. The DOPE score is a pairwise atomic distance statistical potential that assesses atomic distances in a model relative to those observed in many known protein structures. The DOPE potential was derived by comparing the distance statistics from a non-redundant PDB subset of 1,472 high-resolution protein structures with the distance distribution function of the reference state. By default, the DOPE score is not included in the model building routine, and thus can be used as an independent assessment of the accuracy of the output models. In its normalized version (zDOPE), a score below -1.0 indicates a relatively accurate model, with more than 80% of its C α atoms within 3.5 Å of their correct positions. However, it might be that the template does not follow a typical shapes found in the PDB, which will result in a high zDOPE for the experimentally determined template. Thus, it is advised to compare the zDOPE profiles of both target and template.

11. The 10 models of the groEL subunit based of 1iok template achieve low zDOPE score (*i.e.* all models achieved a zDOPE score higher than 0). Visual inspection of the generated models reveals that the c-terminal fragment of the subunit was not covered by the alignment and thus not modeled. After removing this fragment from the models the zDOPE score dropped below 0.

12. MultiFit (**55, 69**) is a method for modeling the structure of a multi-subunit complex by simultaneously optimizing the fit of the model into its EM density map and the shape complementarity between its interacting subunits (<http://www.salilab.org/MultiFit>). It has been shown that the accuracy of both scoring terms is sensitive to errors in comparative modeling (**19, 70**). Thus, if the target(s) share high sequence identity to their template(s), it is advised to model the assembly based on the template structure(s) and then superpose the target models structure on the corresponding templates. For example, here the accuracy of the subunit homology models were low (as indicated by zDOPE and TSVMMod), especially in the loop regions. Thus, we run MultiFit with the template as input and then replaced by template with the subunit model using a

series of transformations commands. A refinement procedure (such as FlexEM) should be next used to fix clashes and improve the fit to the density.

13. Below we highlight key commands in *script9_symmetric_multiple_fitting.py*:

- *runMSPoints.pl* is a perl script for generating Connolly surface (**71**) from the subunit to be fitted.
- *build_cn_multifit_params.py* generated the parameters file being using by MultiFit. The script initialize the algorithm parameters with its defaults. The user can manually adjust these parameters to allow for an enhanced sampling. Example for one such parameter is the *pca_matching_threshold* parameter. MultiFit filters out ring complexes whose pca dimensions do not match the ones of the density map. The acceptable match size is set by the *pca_matching_threshold* parameter with default value of $\frac{3}{4}$ of the EM density map resolution.
- *symmetric_multifit* is the executable that runs MultiFit given the parameters file. The user can control the number of output models by the *-n* option. The results are written into a text file consisting, among others, of the following three key fields: (i) The transformation used to build a symmetric complex is written to the *dock rotation* and *dock translation* fields, (ii) The transformation used to fit the ring into the density is written to the *fit rotation* and *fit translation* fields, and (iii) The cross correlation score (one minus the EM quality of fit score) is written to the *fitting score* field.

14. A FlexEM refinement procedure is composed of two stages. In the CG stage, the positions and orientations of predefined rigid bodies are resolved via a MC/CG minimization; the rigid bodies usually correspond to secondary structure elements. In the MD stage, positions of all atoms are resolved via a fully atomistic molecular dynamics minimization. A FlexEM tutorial can be found at <http://sailab.org/Flex-EM>.

15. Below we highlight key commands in *script10_flexible_fitting.py*:

- Input parameters should be set: (i) *input_pdb_file*, the name of the comparative model file, already rigidly fitted to the density, (ii) *em_map_file* map, the name of the density map file, (iii) *apix*, the density map voxel size, and (iv) *res*, the resolution of the density map.
- The optimization procedure is controlled by few parameters: (i) *rigid_filename*, the name of the file holding the definition of the rigid bodies (see file *rigid_sses.txt* for the format), (ii) *optimization*, which optimization stage to run (CG or MD), (iii) *num_of_runs*, the number of models to produce, and (iv) *initial_dir*, the initial number for the output directories.
- This MD optimization stage is controlled by *md_parameters* (ie, temperatures and number of steps for the simulated annealing algorithm).
- The *md_return* parameter controls the output model reported as final for each run (*final_mdcg.pdb*). The model can be either the last one sampled (FINAL) or the best scoring one (OPTIMAL).
- In our example model #2 got the lowest EM-quality-of-fit score.

Acknowledgments

We are grateful to our colleagues Maya Topf, Friedrich Foerster, Jeremy Phillips, and Daniel Russel for their help with EM fitting, MODELLER, and IMP. We also thank Tom Goddard for help with the IMP/Chimera interface. The research of KL was supported by continuous mentorship from Haim J. Wolfson and by the Clore Foundation PhD Scholars program, and carried out her research in partial fulfillment of the requirements for the Ph.D. degree at TAU. This work was also supported by grants from National Institutes of Health [R01 GM54762, U54 GM074945, U54 GM074929, U01 GM61390, P01 GM71790 (AS), P41 RR01081 (TEF)]; the National Science Foundation [0732065 (AS)], and the Sandler Family Supporting Foundation (AS). We are also grateful for computing hardware gifts from Mike Homer, Ron Conway, NetApp, IBM, Hewlett Packard, and Intel.

References

1. Sali A, Glaeser R, Earnest T et al (2003) From words to literature in structural proteomics. *Nature* 422:216-225
2. Robinson C, Sali A, and Baumeister W (2007) The molecular sociology of the cell. *Nature* 450:973-982
3. Drenth J (2006) *Principles of Protein X-ray Crystallography*, 3rd edn. Springer, New York
4. Bonvin AM, Boelens R, and Kaptein R (2005) NMR analysis of protein interactions. *Current opinion in chemical biology* 9:501-508
5. Neudecker P, Lundstrom P, and Kay LE (2009) Relaxation dispersion NMR spectroscopy as a tool for detailed studies of protein folding. *Biophys J* 96:2045-2054
6. Frank J (2006) *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State* 2nd edn. Oxford University Press, New York
7. Stahlberg H, and Walz T (2008) Molecular electron microscopy: state of the art and current challenges. *Acs Chemical Biology* 3: 268–281
8. Lucic V, Leis A, and Baumeister W (2008) Cryo-electron tomography of cells: connecting structure and function. *Histochem Cell Biol* 130:185-196
9. Zhang J, Baker ML, Schroder GF et al (2010) Mechanism of folding chamber closure in a group II chaperonin. *Nature* 463:379-383
10. Chen JZ, Settembre EC, Aoki ST et al (2009) Molecular interactions in rotavirus assembly and uncoating seen by high-resolution cryo-EM. *Proc Natl Acad Sci U S A* 106:10644-10648
11. Volkman N, and Hanein D (1999) Quantitative fitting of atomic models into observed densities derived by electron microscopy. *J Struct Biol* 125:176-184
12. Roseman AM (2000) Docking structures of domains into maps from cryo-electron microscopy using local correlation. *Acta Crystallogr D Biol Crystallogr* 56:1332-1340
13. Rossmann MG, Bernal R, and Pletnev SV (2001) Combining electron microscopic with x-ray crystallographic structures. *J Struct Biol* 136:190-200
14. Jiang W, Baker ML, Ludtke SJ et al (2001) Bridging the information gap: computational tools for intermediate resolution structure interpretation. *J Mol Biol* 308:1033-1044
15. Chacon P, and Wriggers W (2002) Multi-resolution contour-based fitting of macromolecular structures. *J Mol Biol* 317:375-384
16. Suhre K, Navaza J, and Sanejouand YH (2006) NORMA: a tool for flexible fitting of high-resolution protein structures into low-resolution electron-microscopy-derived density maps. *Acta Crystallogr D Biol Crystallogr* 62:1098-1100
17. Birmanns S, and Wriggers W (2007) Multi-resolution anchor-point registration of biomolecular assemblies and their components. *J Struct Biol* 157:271-280
18. Navaza J, Lepault J, Rey FA et al (2002) On the fitting of model electron densities into EM reconstructions: a reciprocal-space formulation. *Acta Crystallogr D Biol Crystallogr* 58:1820-1825
19. Topf M, Baker M, John B et al (2005) Structural characterization of components of protein assemblies by comparative modeling and electron cryo-microscopy. *J Struct Biol* 149:191-203
20. Lasker K, Dror O, Shatsky M et al (2007) EMatch: discovery of high resolution structural homologues of protein domains in intermediate resolution cryo-EM maps. *IEEE/ACM Trans Comput Biol Bioinform* 4:28-39
21. Goddard TD, Huang CC, and Ferrin TE (2007) Visualizing density maps with UCSF Chimera. *J Struct Biol* 157:281-287

22. Lindert S, Staritzbichler R, Wotzel N et al (2009) EM-fold: De novo folding of alpha-helical proteins guided by intermediate-resolution electron microscopy density maps. *Structure* 17:990-1003
23. Hinsen K, Beaumont E, Fournier B et al (2010) From electron microscopy maps to atomic structures using normal mode-based fitting. *Methods Mol Biol* 654:237-258
24. Orzechowski M, and 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
25. Topf M, Lasker K, Webb B et al (2008) Protein structure fitting and refinement guided by cryo-EM density. *Structure* 16:295-307
26. Trabuco LG, Villa E, Mitra K et al (2008) Flexible fitting of atomic structures into electron microscopy maps using molecular dynamics. *Structure* 16:673-683
27. Schroder GF, Brunger AT, and Levitt M (2007) Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure* 15:1630-1641
28. Sali A, and Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol* 234:779-815
29. Marti-Renom MA, Stuart AC, Fiser A et al (2000) Comparative protein structure modeling of genes and genomes. *Annu Rev Biophys Biomol Struct* 29:291-325
30. Ginalski K (2006) Comparative modeling for protein structure prediction. *Curr Opin Struct Biol* 16:172-177
31. Pieper U, Eswar N, Webb B et al (2009) MODBASE, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res* 37:D347-354
32. Zhu J, Cheng L, Fang Q et al (2010) Building and refining protein models within cryo-electron microscopy density maps based on homology modeling and multiscale structure refinement. *J Mol Biol* 397:835-851
33. Shacham E, Sheehan B, and Volkman N (2007) Density-based score for selecting near-native atomic models of unknown structures. *J Struct Biol* 158:188-195
34. Velazquez-Muriel JA, Sorzano CO, Scheres SH et al (2005) SPI-EM: towards a tool for predicting CATH superfamilies in 3D-EM maps. *J Mol Biol* 345:759-771
35. Alber F, Dokudovskaya S, Veenhoff L et al (2007) Determining the architectures of macromolecular assemblies. *Nature* 450:683-694
36. Pettersen EF, Goddard TD, Huang CC et al (2004) UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem* 25:1605-1612
37. Chiu W, Baker ML, Jiang W et al (2005) Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure* 13:363-372
38. Baker D, and Sali A (2001) Protein structure prediction and structural genomics. *Science* 294:93-96
39. Horwich AL, Farr GW, and Fenton WA (2006) GroEL-GroES-mediated protein folding. *Chem Rev* 106:1917-1930
40. Frydman J (2001) Folding of newly translated proteins in vivo: the role of molecular chaperones. *Annu Rev Biochem* 70:603-647
41. Sigler PB, Xu Z, Rye HS et al (1998) Structure and function in GroEL-mediated protein folding. *Annu Rev Biochem* 67:581-608
42. Xu Z, Horwich AL, and Sigler PB (1997) The crystal structure of the asymmetric GroEL-GroES-(ADP)₇ chaperonin complex. *Nature* 388:741-750
43. Braig K, Adams PD, and Brunger AT (1995) Conformational variability in the refined structure of the chaperonin GroEL at 2.8 Å resolution. *Nat Struct Biol* 2:1083-1094
44. Braig K, Otwinowski Z, Hegde R et al (1994) The crystal structure of the bacterial chaperonin GroEL at 2.8 Å. *Nature* 371:578-586

45. Ludtke SJ, Jakana J, Song JL et al (2001) A 11.5 Å single particle reconstruction of GroEL using EMAN. *J Mol Biol* 314:253-262
46. Clare DK, Bakkes PJ, van Heerikhuizen H et al (2009) Chaperonin complex with a newly folded protein encapsulated in the folding chamber. *Nature* 457:107-110
47. Ludtke SJ, Baker ML, Chen DH et al (2008) De novo backbone trace of GroEL from single particle electron cryomicroscopy. *Structure* 16:441-448
48. Ranson NA, Farr GW, Roseman AM et al (2001) ATP-bound states of GroEL captured by cryo-electron microscopy. *Cell* 107:869-879
49. Alber F, Forster F, Korkin D et al (2008) Integrating diverse data for structure determination of macromolecular assemblies. *Annu Rev Biochem* 77:443-477
50. Berman H, Henrick K, Nakamura H et al (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic Acids Res* 35:D301-303
51. Baker ML, Ju T, and Chiu W (2007) Identification of secondary structure elements in intermediate-resolution density maps. *Structure* 15:7-19
52. Pintilie GD, Zhang J, Goddard TD et al (2010) Quantitative analysis of cryo-EM density map segmentation by watershed and scale-space filtering, and fitting of structures by alignment to regions. *J Struct Biol* 170:427-438
53. Volkman N (2002) A novel three-dimensional variant of the watershed transform for segmentation of electron density maps. *J Struct Biol* 138:123-129
54. Baker ML, Baker MR, Hryc CF et al (2010) Analyses of subnanometer resolution cryo-EM density maps. *Methods Enzymol* 483:1-29
55. Lasker K, Sali A, and Wolfson HJ (2010) Determining macromolecular assembly structures by molecular docking and fitting into an electron density map. *Proteins* 78:3205-3211
56. Khayat R, Lander GC, and Johnson JE (2010) An automated procedure for detecting protein folds from sub-nanometer resolution electron density. *J Struct Biol* 170:513-521
57. Wriggers W, and Chacon P (2001) Modeling tricks and fitting techniques for multiresolution structures. *Structure* 9:779-788
58. Frigo M, and Johnson SG (2005) The Design and Implementation of FFTW3. *Proceedings of the IEEE* 93:216-231
59. Madhusudhan MS, Webb BM, Marti-Renom MA et al (2009) Alignment of multiple protein structures based on sequence and structure features. *Protein Eng Des Sel* 22:569-574
60. Needleman SB, and Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 48:443-453
61. Madhusudhan MS, Marti-Renom MA, Sanchez R et al (2006) Variable gap penalty for protein sequence-structure alignment. *Protein Engineering, Design & Selection* 19:129-133
62. Meng EC, Pettersen EF, Couch GS et al (2006) Tools for integrated sequence-structure analysis with UCSF Chimera. *BMC Bioinformatics* 7:339
63. Shen MY, and Sali A (2006) Statistical potential for assessment and prediction of protein structures. *Protein Sci* 15:2507-2524
64. Eramian D, Eswar N, Shen M et al (2008) How well can the accuracy of comparative protein structure models be predicted? *Protein Sci* 17:1881-1893
65. Melo F, Sanchez R, and Sali A (2002) Statistical potentials for fold assessment. *Protein Sci* 11:430-448
66. Henrick K, Newman R, Tagari M et al (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

67. Putnam CD, Hammel M, Hura GL et al (2007) X-ray solution scattering (SAXS) combined with crystallography and computation: defining accurate macromolecular structures, conformations and assemblies in solution. *Q Rev Biophys* 40:191-285
68. Bishop CM (2007) *Pattern Recognition and Machine Learning* (Information Science and Statistics), 1 edn. Springer, New York
69. Lasker K, Topf M, Sali A et al (2009) Inferential optimization for simultaneous fitting of multiple components into a cryoEM map of their assembly. *J Mol Biol* 388:180-194
70. Ferrara P, and Jacoby E (2007) Evaluation of the utility of homology models in high throughput docking. *J Mol Model* 13:897-905
71. Connolly ML (1983) Solvent-accessible surfaces of proteins and nucleic acids. *Science* 221:709-713

Legends to figures

Figure 1

A flowchart illustrating the steps for modeling a protein complex by comparative modeling and density map fitting.

Figure 2

The steps of EM guided modeling as applied to the GroEL example. (Segmentation) The density map at 11.5 Å resolution is segmented into 14 regions corresponding to the regions occupied by the individual monomers of the assembly. The segments are shown in alternating shades of gray; (Fold detection) Candidate templates are found by scanning the GroEL subunit sequence against the sequences of PDB structures and fitting each of them to the density map. Four of the templates (1we3A, 3kfbA, 1ioaA, 1a6dA), the sequence identity to the target and the fit into the density map of each of them are shown. The selected template is highlighted in green; (Template alignment & model building) Sequence alignment between the target and the selected sequence is generated using a variable gap penalty method. Ten models are constructed and the best model is chosen using the zDOPE, TSVmod, and quality-of-fit scores. A zDOPE profile for the selected model and a superposition of the selected model (green) to the reference structure (gray) are shown; (Multiple fitting) 14 copies of the target model are simultaneously fitted into the density

using the MultiFit method. A model of the complete assembly as generated by MultiFit is shown in green; (Flexible fitting) FlexEM is used to refine the one of the complex subunits to fit the density map. The starting and refined models (green) superposed on the reference structure are shown.

Figure 3

The Python script used for scoring templates by their fit to a segment of a density map.

Figure 4

The Chimera – MODELLER interface. The sequence alignment is displayed in Chimera's Multalign Viewer tool (top). In the dialog for running MODELLER (middle left), one of the sequences in the alignment is designated as the target (sequence: P0A6F5), and at least one structure (associated with another sequence in the alignment) is designated as the template (structure: 1iok). Structure information is shown to help guide the choice of template. After the run, the resulting models are listed along with various model scores from MODELLER in a table (bottom left) and their structures are loaded into Chimera. In this example, the main Chimera window (right) shows the template as an outline and one of the model structures as a ribbon colored by error profile.

Figures

Figure 1

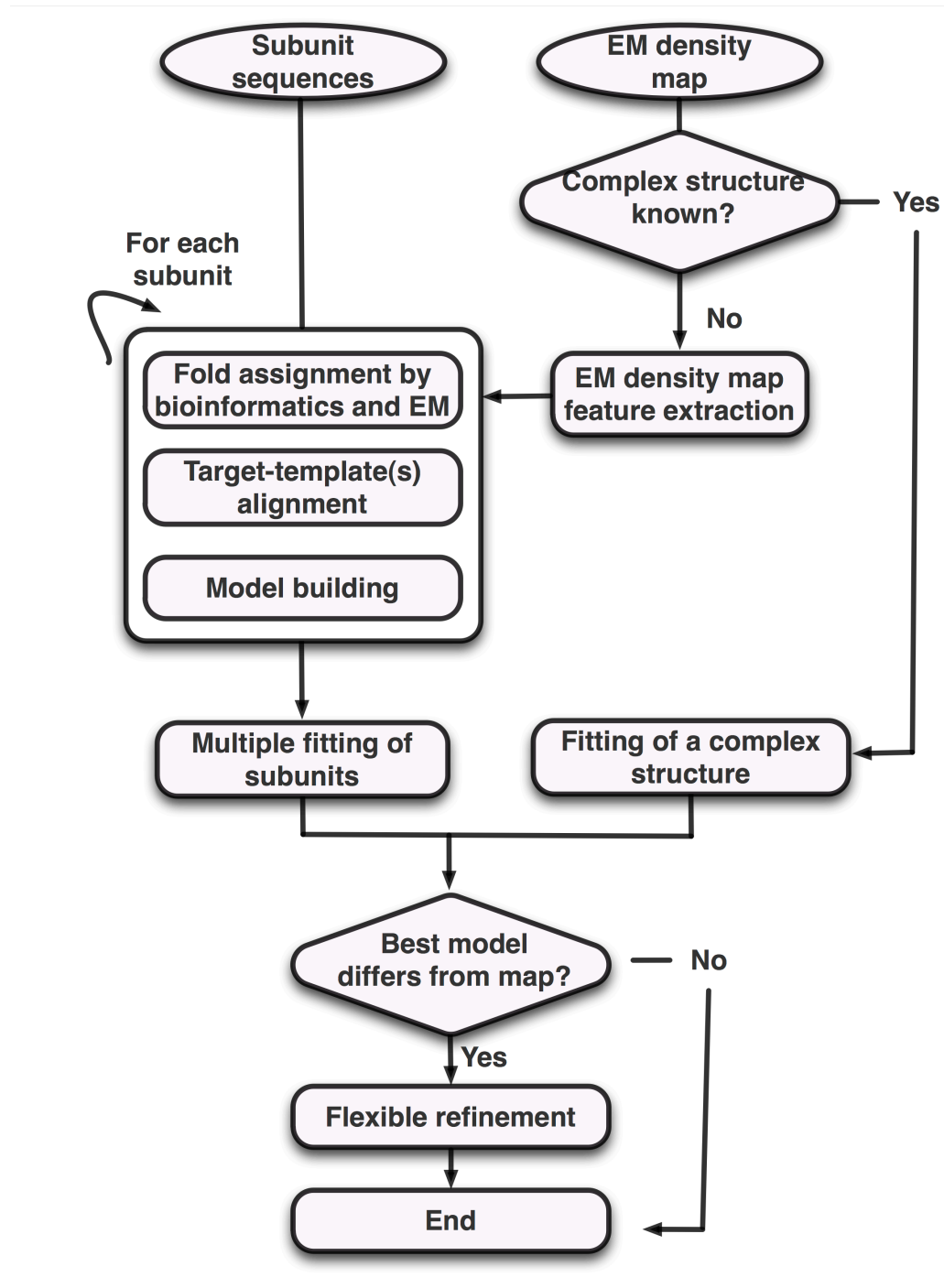


Figure 2

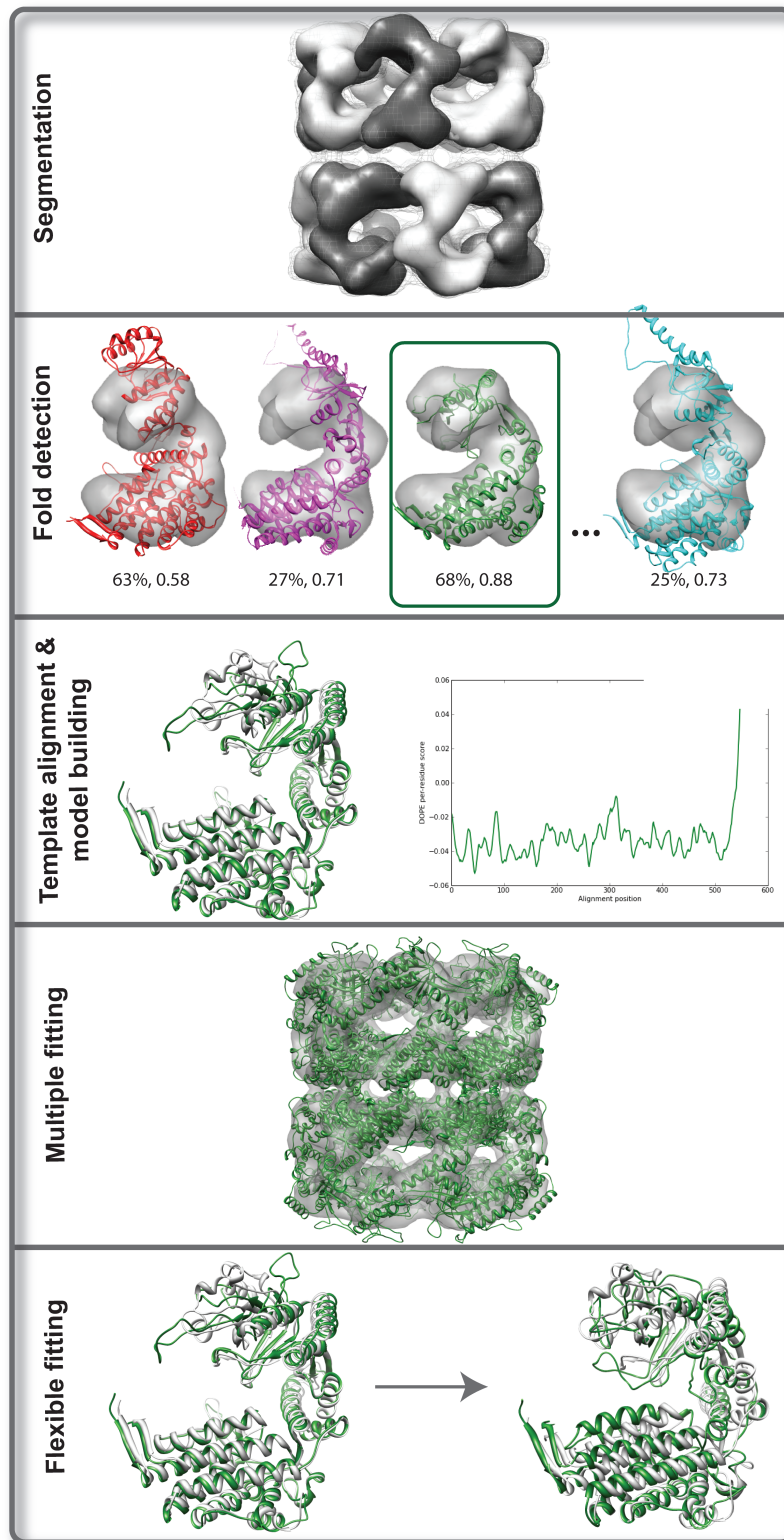


Figure 3

```
import IMP.em
import IMP.atom
import IMP.multifit
IMP.set_log_level(IMP.NONE)
#--- parse the templates file
templates_file="build_profile.prf"
templates=[];seq_ids=[]
for line in open(templates_file):
    if line[0]=="#":
        continue
    s=line.split()
    if int(s[0])!=1:
        continue
    if not(int(s[4])-int(s[3])>480):
        print "Not including:",s[0],s[1]
        templates.append([s[1][:4],s[1][-1]])
        seq_ids.append(s[10])
templates_dir="templates/"
#--- load the target density map
dnmap=IMP.em.read_map("groel_subunit_8.mrc",IMP.em.MRCReaderWriter())
dnmap.get_header_writable().set_resolution(10)
#--- load IMP model
mdl=IMP.Model()
rb_refiner=IMP.core.LeavesRefiner(IMP.atom.Hierarchy.get_traits())
template_fit_sols=[];best_temp=[]
best_fit=1.
dens_threshold=0.02
#--- iterate over the templates and fit each of them
for i,t in enumerate(templates):
    print "fitting template "+t[0]+t[1]
    #load the template
    mh=IMP.atom.read_pdb(templates_dir+t[0]+".pdb",mdl)
    IMP.atom.add_radii(mh)
    #get the right chain
    mh_chain=IMP.atom.get_by_type(mh,IMP.atom.CHAIN_TYPE)[ord(t[1])-ord('A')]
    rb=IMP.atom.setup_as_rigid_body(mh_chain)
    #fit the template to the density map
    sols=IMP.multifit.pca_based_rigid_fitting(rb,rb_refiner,dnmap,dens_threshold)
    IMP.core.transform(rb,sols.get_transformation(0))
    #refine the best scoring fit
    mhs=IMP.atom.Hierarchies()
    mhs.append(mh_chain)
    pdb_opt_state=None #IMP.atom.WritePDBOptimizerState(mhs,"refined_temp_%03d.pdb")
    refined_sols = IMP.em.local_rigid_fitting(
        rb,rb_refiner,
        IMP.core.XYZR.get_default_radius_key(),
        IMP.atom.Mass.get_mass_key(),dnmap,pdb_opt_state,1,3,1000)
    IMP.core.transform(rb,refined_sols.get_transformation(0))
    IMP.atom.write_pdb(mh_chain,t[0]+t[1]+"_fitted.pdb")
    template_fit_sols.append([
        refined_sols.get_transformation(0)*sols.get_transformation(0),refined_sols.get_score(0)])
    IMP.core.transform(rb,refined_sols.get_transformation(0).get_inverse())
    IMP.core.transform(rb,sols.get_transformation(0).get_inverse())
#--- write the best fitting score for each template
output=open("score_templates_by_cc.log","w")
output.write('{0:<2}{1:<2}{2:<2}{3:<3}\n'.format('name','seq id','cc score','transformation'))

for i,t in enumerate(templates):
    rot=template_fit_sols[i][0].get_rotation().get_quaternion()
    v=template_fit_sols[i][0].get_translation()
    pretty_trans='{0:3.6f} {1:3.6f} {2:3.6f} {3:3.6f} {4:3.6f} {5:3.6f} {6:3.6f}\n'.format(
        rot[0],rot[1],rot[2],rot[3],v[0],v[1],v[2])
    output.write('{0:<2}{1:<2}{2:<2}{3:<3}\n'.format(
        t[0]+t[1],seq_ids[i],1.-template_fit_sols[i][1],pretty_trans))
output.close()
```

Figure 4

