Modeling of proteins and their assemblies with the *Integrative Modeling Platform*

Contributed by Benjamin Webb, Keren Lasker, Javier Velázquez-Muriel, Dina Schneidman-Duhovny, Riccardo Pellarin, Massimiliano Bonomi, Charles Greenberg, Barak Raveh, Elina Tjioe, Daniel Russel, and Andrej Sali, Department of Bioengineering and Therapeutic Sciences, Department of Pharmaceutical Chemistry, and California Institute for Quantitative Biosciences (QB3), University of California San Francisco, San Francisco, CA 94158, USA.

Running Head: Integrative Modeling Platform

Abstract

To understand the workings of the living cell, we need to characterize protein assemblies that constitute the cell (for example, the ribosome, 26S proteasome, and the nuclear pore complex). A reliable high-resolution structural characterization of these assemblies is frequently beyond the reach of current experimental methods, such as X-ray crystallography, NMR spectroscopy, electron microscopy, footprinting, chemical cross-linking, FRET spectroscopy, small angle X-ray scattering, and proteomics. However, the information garnered from different methods can be combined and used to build models of the assembly structures that are consistent with all of the available datasets, and therefore more accurate, precise, and complete. Here, we describe a protocol for this integration, whereby the information is converted to a set of spatial restraints and a variety of optimization procedures can be used to generate models that satisfy the restraints as well as possible. These generated models can then potentially inform about the precision and accuracy of structure determination, the accuracy of the input datasets, and further data generation. We also demonstrate the *Integrative Modeling Platform* (IMP) software, which provides the necessary computational framework to implement this protocol, and several applications for specific use cases.

Key words: Integrative modeling, Protein structure modeling, Proteomics of Macromolecular assemblies, X-ray crystallography, Electron microscopy, SAXS

1. Introduction

To understand the function of a macromolecular assembly, we must know the structure of its components and the interactions between them***(1-4)***. However, direct experimental determination of such a structure is generally rather difficult. While multiple methods do exist for structure determination, each has a drawback. For example, crystals suitable for X-ray crystallography cannot always be produced, especially for large assemblies of multiple components***(5)***. Cryo-electron microscopy (cryo-EM), on the other hand, can be used to study large assemblies, but it is generally limited to worse than atomic resolution***(6-8)***. Finally, proteomics techniques, such as yeast two-hybrid***(9)*** and mass spectrometry***(10)***, yield information about the interactions between proteins, but not the positions of these proteins within the assembly or the structures of the proteins themselves.

1.1. Integrative modeling

One approach to solve the structures of proteins and their assemblies is by integrative modeling, in which information from different methods is considered simultaneously during the modeling procedure. The approach is briefly outlined here for clarity; it has been covered in greater detail previously***(11-18)***. These individual methods can include experimental techniques, such as X-ray crystallography***(5)***, nuclear magnetic resonance (NMR) spectroscopy***(19-21)***, electron microscopy (EM)***(6-8)***, footprinting***(22,23)***, chemical cross-linking***(24-27)***, FRET spectroscopy***(28)***, small angle X-ray scattering (SAXS)***(29-31)***, and proteomics***(32)***. Theoretical sources of information about the assembly can also be incorporated, such as template structures used in comparative modeling***(33,34)***, scoring functions used in molecular docking***(35)***, as well as other statistical preferences***(36,37)*** and physics-based energy functions***(38-40)***. Different methods yield information about different aspects of structure and at different levels of resolution. For example, atomic resolution structures may be available for individual proteins in the assembly; in other cases, only their approximate size, approximate shape, or interactions with other proteins may be known. Thus, integrative modeling techniques generate models at the resolution that is consistent with the input information. An example of a simple integrative approach is building a pseudo-atomic model of a large assembly, such as the 26S proteasome***(41-43)***, by fitting atomic structures of its subunits predicted by comparative protein structure modeling into a density map determined by cryo-EM***(44,45)***.

The integrative modeling procedure used here***(13,18)*** is schematically shown in Figure 1. The first step in the procedure is to collect all experimental, statistical, and physical information that describes the system of interest. A suitable representation for the system is then chosen and the available information is translated to a set of spatial restraints on the components of the system. For example, in the case of characterizing the molecular architecture of the nuclear pore complex (NPC)***(13,14)***, atomic structures of the protein subunits were not available, but the approximate size and shape of each protein was known, so each protein was represented as a 'string' of connected spheres consistent with the protein size and

shape. A simple distance between two proteins can be restrained by a harmonic function of the distance, while the fit of a model into a 3D cryo-EM density map can be restrained by the cross-correlation between the map and the computed density of the model. Next, the spatial restraints are summed into a single scoring function that can be sampled using a variety of optimizers, such as conjugate gradients, molecular dynamics, Monte Carlo, and divide-and-conquer message passing methods***(45)***. This sampling generates an ensemble of models that are as consistent with the input information as possible. In the final step, the ensemble is analyzed to determine, for example, whether all of the restraints have been satisfied or certain subsets of data conflict with others. The analysis may generate a consensus model, such as the probability density for the location of each subunit in the assembly.

1.2. *Integrative Modeling Platform*

We have developed the *Integrative Modeling Platform* (IMP) software (http://salilab.org/imp/)***(11,13-16)*** to implement the integrative modeling procedure described above. Integrative modeling problems vary in size and scope, and thus IMP offers a great deal of flexibility and several abstraction levels as part of a multi-tiered platform (Figure 2). At the lowest level, IMP provides building blocks and tools to allow methods developers to convert data from new experimental methods into spatial restraints, to implement optimization and analysis techniques, and to implement an integrative modeling procedure from scratch; the developer can use the C++ and Python programming languages to achieve these tasks. Higher abstraction levels, designed to be used by IMP users with no programming experience, provide less flexible but more user-friendly applications to handle specific tasks, such as fitting of proteins into a density map of their assembly, or comparing a structure with the corresponding SAXS profile. IMP is freely available as open source software under the terms of the GNU Lesser General Public License (LGPL). Integrative modeling, due to its use of multiple sources of information, is often a highly collaborative venture, and thus benefits from openness of the modeling protocols and the software itself.

2. Materials

To follow the examples in this discussion, both the IMP software itself and a set of suitable input files are needed. The IMP software can be downloaded from http://salilab.org/imp/download.html and is available in binary form for most common machine types and operating systems; alternatively, it can be rebuilt from the source code; either the stable 2.0 release of IMP, or a recent development version, should be used. The example files can be downloaded from http://salilab.org/imp/tutorials/basic_apps.zip. Certain applications also make use of third party software, which must be obtained separately from IMP (download locations for each software package are shown in subsequent sections).

2.1. Typographical conventions

`Monospaced text` is used below for computer file and folder/directory names, command lines, file contents, and variable and class names.

3. Methods

3.1. The IMP C++/Python library

The core of IMP is the C++/Python library, which provides all of the necessary components, as a set of classes and modules, to allow method developers to build an integrative modeling protocol from scratch. This core can be used either from C++ (by including the `IMP.h` header file and linking against the IMP libraries) or from Python (by importing the IMP Python module), and provides almost identical functionality in each language, for maximum flexibility. In this text, we will demonstrate the IMP applications that build on top of this core; the core itself has been demonstrated elsewhere***(46)*** and is further described on the IMP website, http://salilab.org/imp/.

3.2. Pairwise protein-protein docking integrating data from SAXS and EM

One major computational approach to predicting structures of protein complexes relies on molecular docking of unbound single-component structures. However, even for complexes with two proteins, the docking problem remains challenging despite recent advances***(47)***. The major bottlenecks include dealing with protein flexibility and the absence of an accurate scoring function***(48)***.

IMP includes an integrative approach to pairwise protein docking, in which additional experimental information about the protein-protein complex is incorporated into the docking procedure to greatly improve the accuracy of predictions. This method succeeds in producing a near-native model among the top 10 models in 42–82% of cases, while state-of-the-art docking methods succeed only in 30–40% of cases, depending on the benchmark and accuracy criterion***(49)***.

The protocol proceeds as follows (Figure 3). First, data from one or more of five different experiment types are translated into the corresponding scoring function terms. These data include (i) the pair-distance distribution function of the complex from a SAXS profile, (ii) 2D class average images of the complex from negative-stain EM micrographs (EM2D), (iii) a 3D density map of the complex from single-particle negative-stain EM micrographs (EM3D), (iv) residue type content at the protein interface from NMR spectroscopy (NMR-RTC)***(50)***, and (v) chemical cross-linking detected by mass spectrometry (CXMS). These five experimental methods were selected because of their feasibility and efficiency of data collection: a SAXS profile of the complex in solution can be collected in several minutes***(30)***; a 3D EM density map can be reconstructed from a smaller sample amount than that for SAXS, but the data collection process is significantly longer***(6)***; 2D class averages can be computed from micrographs more easily and rapidly than performing a full 3D reconstruction; the composition of interface residues from NMR***(50)*** provides information about the interaction interface, unlike the SAXS and EM data; and cross-linking data***(51)*** provide information at intermediate resolution imposing an upper distance bound on inter-molecular pairs of residues. Second, complex models are sampled, relying on efficient global search methods developed for pairwise protein docking, followed

by filtering based on fit to the experimental data, conformational refinement and composite scoring. Third, good-scoring representatives of clusters of models are picked as final models.

Here, we demonstrate the approach by application to the PCSK9 antigen–J16 Fab antibody complex. All input files for this example can be found in the '`idock`' directory of the downloaded zipfile.

Step 1. Inputs. The primary inputs are the Protein Data Bank (PDB)***(52)*** structures of the isolated J16 Fab antibody and PCSK9 antigen, `antibody_cut.pdb` and `2p4e.pdb`, respectively; they can be found in the downloaded zipfile. We also collected SAXS, EM2D and EM3D data on this protein-protein complex, available in the `iq.dat`, `image_*.pgm`, and `complex.mrc` files, respectively. Finally, we added missing residues to both PDB files, for use in SAXS scoring, yielding `antibody.pdb` and `pcsk9.pdb` (*see* **Note 1**).

Step 2. Docking. We can then carry out all steps of the integrative docking by running IMP's `idock.py` application (*see* **Note 2**), giving it the names of our input files:

```
idock.py antibody_cut.pdb 2p4e.pdb --saxs iq.dat --em3d
complex.mrc --em2d image_1.pgm --em2d image_2.pgm --em2d
image_3.pgm --pixel_size 2.2 --complex_type AA --
saxs_receptor_pdb antibody.pdb --saxs_ligand_pdb pcsk9.pdb
--precision 2
```

The application makes use of the PatchDock and FireDock programs for docking and refinement, which must be obtained separately from http://bioinfo3d.cs.tau.ac.il/, and the 'reduce' program for adding hydrogens to PDB files, available from http://kinemage.biochem.duke.edu/software/reduce.php.

Step 3. Results. Once the docking procedure has finished, the primary output file generated is `results_saxs_em3d_em2d.txt`, the first few lines of which look similar to:

```
    # |  Score  | filt| ZScore |  SAXS  | Zscore |  EM2D  |
Zscore |  EM3D  | Zscore | Energy | Zscore | Transformation
    1 |  -5.225 |  +  | -3.318 | 16.304 | -1.454 |  0.685 |
-1.829 |  0.058 | -1.672 | -20.010 | -0.270 |   2.4462
0.7439 2.0137 32.0310 36.5010 74.9757
    2 |  -4.453 |  +  | -2.828 | 17.590 |  0.578 |  0.698 |
-2.521 |  0.064 | -1.243 | -42.220 | -1.267 |   0.1525 -
1.3733 2.1213 -17.2068 -10.3519 13.3553
```

Each line corresponds to one model; the models are ranked by total score, best first. The individual SAXS, EM2D, and EM3D score/z-score pairs are also shown (only

docking solutions that were not filtered out by any of three data sources – i.e. they scored well against every source – are included in this file). The last column is a transformation (3 rotation angles and a translation vector) that transforms the antibody relative to the antigen (the antigen is not transformed).

3.3. Determining macromolecular assembly structures by fitting multiple structures into an electron density map

Often, we have available high-resolution (atomic) information for the subunits in an assembly, and low-resolution information for the assembly as a whole, such as a cryo-EM electron density map. A high-resolution model of the whole assembly can thus be constructed by simultaneously fitting the subunits into the density map. Fitting of a single protein into a density map is usually done by calculating the electron density map of the protein followed by a search for the protein position in the cryo-EM map that maximizes the cross correlation of the two maps. Simultaneously fitting multiple proteins into a given map is significantly more difficult though, since an incorrect fit of one protein will also prevent other proteins from being placed correctly.

IMP contains a MultiFit***(44,45)*** application (http://salilab.org/multifit/) that can efficiently solve such multiple fitting problems for density map resolutions as low as 25Å, relying on a general divide-and-conquer optimizer DOMINO. The application is available both within IMP and as a web interface on the MultiFit website. The fitting protocol is a multi-step procedure that proceeds *via* discretization of both the map and the proteins, local fitting of the proteins into the map, and an efficient combination of local fits into global solutions (Figure 4). It is also able to incorporate additional information about interactions between the proteins from proteomics experiments, and can take advantage of C_n symmetry to generate structures of such symmetric complexes. Here, we will demonstrate the use of MultiFit in building a model of porcine mitochondrial respiratory complex II (PDB id 3sfd), using crystal structures of its 4 constituent proteins and a 15Å density map of the entire assembly. All input files for this procedure can be found in the '`multifit`' subdirectory of the downloaded zipfile. The protocol consists of the following steps:

Step 1: Setup a subunit list. We create an input file listing the subunits involved in the complex. The file contains one line per component with the following information: the name that MultiFit will use for the component, a path to the file containing the atomic coordinates for the component, and a 0/1 fitting flag indicating whether placements of the subunit should be sampled locally (0) or globally (1). The default for the fitting flag is 1 (global search). If the user has prior knowledge or a good hypothesis as to the subunit position, he can provide the proposed subunit placement in the atomic coordinates file and ask for a local search.

In this example, we assume no prior knowledge and provide the following input file (`input/3sfd.subunits.txt`):

```
3sfdA            ../input/3sfd.A.pdb  1
3sfdB            ../input/3sfd.B.pdb  1
3sfdC            ../input/3sfd.C.pdb  1
3sfdD            ../input/3sfd.D.pdb  1
```

Step 2: Create input files. We generate two input files to guide the protocol, by running the MultiFit application, `multifit.py` (*see* **Note 3**). In this case, we use the application's 'param' command, by typing on a command line:

```
multifit.py param -i 3sfd.asmb.input -- 3sfd.asmb
input/3sfd.subunits.txt 30 input/3sfd_15.mrc 15 3. 335 27.0
-6.0 21.0
```

The first file generated by this command, `3sfd.asmb.input`, provides information on each of the subunits and their assembly density map, such as names of the files from which the input structures and map will be read, and those to which outputs from later steps will be written. The second file, `3sfd.asmb.alignment.param`, specifies scoring and optimization parameters for each step of the MultiFit application. The user is advised to read a detailed description of the different parameters on the MultiFit website (http://salilab.org/multifit/) for better understanding of the algorithm and for troubleshooting difficult modeling cases.

The arguments to the 'param' command include the spacing, origin, resolution, and density threshold of the input density map. The spacing and origin are often stored in the map header. To view the map header, run

```
view_density_header.py input/3sfd_15.mrc
```

The resolution is typically not stored in the map header; it is usually provided in the corresponding publication and can also be found in the corresponding EMDB***(53)*** entry. A threshold is often provided by the author in the EMDB entry as "Recommended counter level" under the "Map Information" section. Alternatively, IMP provides a utility to calculate an approximate counter level based on the molecular mass of the complex, which can be run as:

```
estimate_threshold_from_molecular_mass.py input/3sfd_15.mrc
1092
```

Step 3: Create the assembly anchor graph. We determine a reduced representation of the assembly density map using the Gaussian Mixture Model, by running:

```
multifit.py anchors 3sfd.asmb.input 3sfd.asmb.anchors
```

This command computes a reduced representation of the EM map that best reproduces the configuration of all voxels with density above the density threshold provided in the `3sfd.asmb.input` file as a set of 3D Gaussian functions (*see* **Note 4**). The reduced representation is written out as a PDB file containing fake Cα atoms, where each Cα corresponds to a single anchor point, and also as a Chimera***(54)*** cmm file.

Step 4: Fit each protein to the map. We first fit each protein to the map using a FFT search either globally or locally:

```
multifit.py fit_fft -a 30 -n 1000 -v 60 -c 6
3sfd.asmb.input
```

The output is a set of candidate fits. In each file, a single subunit is rigidly rotated and translated to fit into the density map. Each fit is written out as the transformation (rotation and translation) required to place the original subunit in the density map. The fitting of a subunit into the density map is performed by globally searching for subunit transformations yielding high cross-correlation between the subunit and the map via a fast Fourier transform.

Second, we create a list of valid fit indexes. By default, this list is simply the top 10 hits from fit_fft, but they could be filtered by other criteria (e.g., proximity to anchor points) if desired. We do this task by running:

```
multifit.py indexes 3sfd 3sfd.asmb.input 10
3sfd.indexes.mapping.input
```

Step 5: Create a proteomics restraint file. We create the restraint file used in the next assembly step (*see* **Note 5**). This file instructs MultiFit how to combine the individual subunit fits created above into a global solution of all subunits simultaneously fitted into the map. First, we ask MultiFit to generate a basic proteomics file, indicating between which pairs of proteins a complementarity restraint (i.e., that the surfaces of the proteins should fit and complement each other) should be calculated:

```
multifit.py proteomics 3sfd.asmb.input 3sfd.asmb.proteomics
```

The user can then add additional information from proteomics experiments to this file. Here, we add 7 simulated residue-residue cross-link restraints as indicated in `input/3sfd.xlinks`. We also update the excluded volume (EV) pairs to calculate complementarity restraints between pairs of proteins as indicated by the cross-link restraints (*see* **Note 6**). After these additions, the final `3sfd.asmb.proteomics` file is:

```
|proteins|
|3sfdA|1|613|nn|nn|
```

```
|3sfdB|1|239|nn|nn|
|3sfdC|1|138|nn|nn|
|3sfdD|1|102|nn|nn|
|interactions|
|residue-xlink|
|1|3sfdB|23|3sfdA|456|30|
|1|3sfdB|241|3sfdC|112|30|
|1|3sfdB|205|3sfdD|37|30|
|1|3sfdB|177|3sfdD|99|30|
|1|3sfdC|95|3sfdD|132|30|
|1|3sfdC|9|3sfdD|37|30|
|1|3sfdC|78|3sfdD|128|30|
|ev-pairs|
|3sfdB|3sfdA|
|3sfdB|3sfdC|
|3sfdC|3sfdD|
```

Step 6: Assemble subunits. The fits are combined into a set of the best-scoring global configurations by running:

```
multifit.py align 3sfd.asmb.input 3sfd.asmb.proteomics
3sfd.indexes.mapping.input 3sfd.asmb.alignment.param
3sfd.asmb.combinations 3sfd.asmb.combinations.fit.scores
```

The scoring function used to assess each fit includes the quality-of-fit of each subunit in the map, the protrusion of each subunit out of the map envelope, the shape complementarity between subunits, as indicated in the proteomics file, and distance restraints as defined by proteomics data, also from the proteomics file. The optimization avoids exhaustive enumeration of all possible mappings of subunits to anchor points by means of a branch-and-bound algorithm combined with the DOMINO divide-and-conquer message-passing optimizer using a discrete sampling space***(45)***.

Step 7: Ensemble analysis. First, we cluster the top 100 models such that the maximum Cα RMSD between members of a cluster is 5Å:

```
multifit.py cluster 3sfd.asmb.input 3sfd.asmb.proteomics
3sfd.asmb.mapping.input 3sfd.asmb.alignment.param
3sfd.asmb.combinations -r 5 -m 100
```

The first cluster consists of 96 models and the second cluster consists of 4 models. The average Cα RMSD between members of the first cluster is 3.4Å with a standard deviation of 0.3Å.

The clustering procedure also generates a new combination file consisting of combinations of the cluster representatives. We can further investigate these cluster representatives by calculating scores of individual restraints:

```
multifit.py score 3sfd.asmb.input 3sfd.asmb.proteomics
3sfd.indexes.mapping.input 3sfd.asmb.alignment.param
3sfd.asmb.combinations.clustered
3sfd.asmb.combinations.clustered.scores
```

Finally, we can generate models (as PDB files) by running:

```
multifit.py models 3sfd.asmb.input 3sfd.asmb.proteomics
3sfd.indexes.mapping.input 3sfd.asmb.combinations.clustered
3sfd.model
```

These models can be visualized in any PDB viewer, such as Chimera***(54)***.

3.4. Assembly of macromolecular complexes by satisfaction of spatial restraints from EM images

Obtaining a high-resolution density map by EM requires a large number of single-particle images and needs an initial low-resolution template density map to perform 3D reconstruction. This procedure is not always possible because in difficult cases the assembly only shows a set of preferred orientations during imaging. However, calculating average 2D images (class averages) from the images of single particles in the same orientation is relatively simple and fast. IMP provides an 'EMageFit' application that performs integrative modeling to assemble the subunits of a macromolecular complex using a few class averages, in much the same way MultiFit does for density maps. The class averages can be combined with maximum distance and proximity restraints, such as those from chemical cross-linking and proteomics experiments, respectively. Additionally, an excluded volume restraint prevents the subunits from overlapping. The optimization procedure is a two-step method consisting of building a set of models by Simulated Annealing Monte Carlo (SA-MC) optimization followed by a refinement with DOMINO***(55)***. An example of the application of EMageFit to the same complex studied above with MultiFit can be found in the '`emagefit`' directory of the downloaded zipfile (*see* **Note 7**).

The inputs for modeling are the three simulated class averages of the complex located in the `em_images` directory, and the subunits to assemble: `3sdfA.pdb`, `3sfdB.pdb`, `3sfC.pdb`, and `3sfdD.pdb`. The Python file `config_step_1.py` contains all the necessary options and restraint specifications (*see* **Note 8**). Briefly, the class averages are given in the `em2d_restraints` variable; four proteomics restraints are specified in the variable `pair_score_restraints`; and 7 cross-links are specified in the variable `xlink_restraints`. The SA-MC optimization is set as a tempering schedule and the best 50 models are selected for refinement with

DOMINO. For an extensive description of all the parameters in `config_step_1.py` see `config_example.py`.

Building a model requires 4 steps: pairwise docking between interacting subunits, SA-MC optimization, SA-MC model gathering, and DOMINO sampling.

Step 1. Pairwise docking. The pairwise dockings are calculated with the program HEXDOCK***(56)***, which can be obtained from http://hex.loria.fr/, based on the description of connectivity between subunits given by the cross-linking restraints. To perform this step, run from the command line:

```
emagefit.py --exp config_step_1.py --dock --log file.log
```

The docking results will be used during the SA-MC optimization to quickly explore feasible relative positions between pairs of components (although helpful, the dockings are not strictly required and EMageFit can work without them). The command produces multiple files: the PDB files of the initial docking solutions as estimated from the cross-linking restraints (ending in `initial_docking.pdb`); the PDB files with the best solutions from HEXDOCK (ending in `hexdock.pdb`); a set of text files starting with `hex_solutions`, containing all the solutions from HEXDOCK; and 4 text files starting with `relative_positions`, which contain the relative transformations (in IMP convention) between the subunits participating in each pairwise docking. The latter files are used by the SA-MC optimization. All described files can also be found in the `outputs` directory.

Step 2. SA-MC optimization. The parameters controlling the optimization are in the `MonteCarloParams` class in `config_step_2.py`: the profile of temperatures, the number of iterations, number of cycles, and the maximum change in position and orientation tolerated for the random moves. The parameter `non_relative_move_prob` indicates the probability for a component of undergoing a random move instead of a docking-derived relative move. To ignore all docking solutions, or if they are not available, use a value of 1. Other important variables are `dock_transforms`, which specifies the files of relative orientations found previously, and `anchor`, which indicates the components that will not move during the SA-MC optimization. The command for producing one model is:

```
emagefit.py --exp config_step_2.py --o mc_solution1.db --
log file.log --monte_carlo -1
```

The output is the file `mc_solution1.db`, an SQLite database with the solution. To generate multiple candidate solutions, simply run the script multiple times, changing the name of the output file from `mc_solution1.db` each time (*see* **Note 9**).

Step 3. Model gathering. Here we gather all the models produced with SA-MC:

```
emagefit.py --o monte_carlo_solutions.db --gather {all
database files}
```

Here `{all database files}` are the databases to merge and `monte_carlo_solutions.db` is the output database with all the merged results. For this example, we have already included in the zipfile a file `monte_carlo_solutions.db` containing 500 models, so you can skip this step if desired.

Step 4. DOMINO sampling. `DominoSamplingPositions` and `DominoParams` in `config_step_3.py` include the relevant parameters: `read` is the file with the SA-MC solutions obtained before, `max_number` is the maximum number of solutions to combine, and `orderby` is the name of the restraint used to sort the SA-MC solutions. The command is:

```
emagefit.py --exp config_step_3.py --o domino_solutions.db
--log file.log
```

This command will produce a database `domino_solutions.db` with all the results. We include the file in the `outputs` directory.

The best solutions can be written out in the PDB format. To write the 10 best models according to the value of the em2d restraint, run:

```
emagefit.py --exp config_step_3.py --o domino_solutions.db
--w 10 –orderby em2d --log file.log
```

The best solution and its fit into the density map of the complex are shown in Figure 5. Finally, the solutions stored in the database can be clustered with the `emagefit_cluster.py` script. To cluster the first 100 solutions according to the value of the em2d restraint and save the results to `clusters.db`, use:

```
emagefit_cluster.py --exp config_step_3.py --db
domino_solutions.db --o clusters.db --n 100 --orderby em2d
--log clusters.log --rmsd 10
```

And to write the elements of the first cluster as PDB files:

```
emagefit.py --exp config_step_3.py --o domino_solutions.db
--wcl clusters.db 1
```

In summary, we have shown with this example how to combine multiple pairwise dockings, EM class averages and distance restraints for assembling the subunits of a

macromolecular complex; integrating new restraints into the optimization protocol is also possible.

3.5. Summary

The structures of protein assemblies can typically not be fully characterized with any individual computational or experimental method. Integrative modeling aims to solve this problem by combining information from multiple methods to generate structural models. Integrative modeling problems can be tackled by satisfaction of spatial restraints, where information for individual restraints can come from different methods. In this approach, a suitable representation for the system is chosen, the information is converted into a set of spatial restraints, the restraints are simultaneously satisfied as well as possible by optimizing a function that is the sum of all restraints, and the resulting models are analyzed. Further experiments as well as the precision and likely accuracy of both the model and the data can be informed. IMP is an open source and flexible software package that provides all of the components needed to implement an integrative modeling protocol from scratch. It also contains higher-level applications and web services that can tackle specific use cases more conveniently.

4. Notes

1. X-ray crystal structures are often missing coordinates for some of the residues. Since a SAXS profile is typically experimentally determined for the entire structure, including these missing residues, the X-ray structure will not perfectly fit the SAXS profile. To improve the SAXS fit, we added missing loops, N-termini, C-termini and His tags for both structures using MODELLER.

2. All of the IMP applications demonstrated here are command line tools, and must be run by typing at a command line. The user is expected to unzip the downloaded zipfile before running any of the examples, and then to run the command in the directory corresponding to the example (‘`idock`’ in this case). Each of the command lines shown in this text should be entered as a single line, even though some have been wrapped onto multiple lines.

3. For detailed help on each step of the MultiFit protocol, run ‘`multifit.py help`’ from the command line.

4. The default number of Gaussians is the number of components. However, if the sizes of the subunits differ, it is recommended to use the `-s` option to set the number of residues encapsulated in each Gaussian. For example, if you choose 50 residues per Gaussian, a 170-residue protein should use 3 Gaussians and a 260-residue protein should use 5 Gaussians.

5. A detailed description of the format of the proteomics file can be found on the MultiFit website.

6. The restraints will be used to create DOMINO's junction tree. DOMINO works most efficiently if the size of the intermediate subsets is small. Use the '`multifit.py merge_tree`' command to view the tree defined by the restraints. To reduce the size of the subsets, the user can determine which restraints are used to define the merge tree by setting the first value in the xlink definition. Setting the value to 0 instead of the default 1 specifies that the restraint is evaluated only at the root of the tree and not in an intermediate merging step.

7. An extended version of this manual is available on the IMP website.

8. We have used different configuration files for each step in this example for clarity, but it is possible to use a single one for all steps with all the options.

9. Different models are generated each time the script is run, because Monte Carlo relies on random moves, the specific sequence of which is uniquely determined by a random number seed, and the '`-1`' argument to the '`--monte-carlo`' option instructs the script to use the current time as the seed. This can be a problem if multiple copies of the script are started at exactly the same time (e.g. on a cluster or a multi-core computer) as they will generate the same model. To avoid this scenario, or to generate models that can be exactly reproduced, replace -1 with a specific seed for each model (e.g. 1 for the first model, 2 for the second, and so on).

Acknowledgements

We are grateful to all members of our research group, especially to Frank Alber, Friedrich Förster, and Bret Peterson who contributed to early versions of IMP, and Marc Marti-Renom, Davide Baù, Benjamin Schwarz, and Yannick Spill who currently contribute to IMP. We also acknowledge support from National Institutes of Health (R01 GM54762, U54 RR022220, PN2 EY016525, and R01 GM083960) as well as computing hardware support from Ron Conway, Mike Homer, Hewlett-Packard, NetApp, IBM, and Intel.

References

1. Schmeing T.M., Ramakrishnan V. (2009) What recent ribosome structures have revealed about the mechanism of translation. *Nature* **461** (7268) 1234-1242
2. Sali A., Glaeser R., Earnest T., et al. (2003) From words to literature in structural proteomics. *Nature* **422** (6928) 216-225
3. Mitra K., Frank J. (2006) Ribosome dynamics: insights from atomic structure modeling into cryo-electron microscopy maps. *Annu Rev Biophys Biomol Struct* **35** 299-317
4. Robinson C., Sali A., Baumeister W. (2007) The molecular sociology of the cell. *Nature* **450** (7172) 973-982

5. Blundell T., Johnson L. (1976) Protein Crystallography. Academic Press, New York
6. Stahlberg H., Walz T. (2008) Molecular electron microscopy: state of the art and current challenges. *ACS Chem Biol* **3** (5) 268-281
7. Chiu W., Baker M.L., Jiang W., et al. (2005) Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure* **13** (3) 363-372
8. Lucic V., Leis A., Baumeister W. (2008) Cryo-electron tomography of cells: connecting structure and function. *Histochem Cell Biol* **130** (2) 185-196
9. Parrish J.R., Gulyas K.D., Finley R.L., Jr. (2006) Yeast two-hybrid contributions to interactome mapping. *Curr Opin Biotechnol* **17** (4) 387-393
10. Gingras A.C., Gstaiger M., Raught B., et al. (2007) Analysis of protein complexes using mass spectrometry. *Nat Rev Mol Cell Biol* **8** (8) 645-654
11. Russel D., Lasker K., Webb B., et al. (2012) Putting the pieces together: integrative structure determination of macromolecular assemblies. *PLoS Biol* **10** (1) e1001244
12. Alber F., Kim M., Sali A. (2005) Structural characterization of assemblies from overall shape and subcomplex compositions. *Structure* **13** (3) 435-445
13. Alber F., Dokudovskaya S., Veenhoff L., et al. (2007) Determining the architectures of macromolecular assemblies. *Nature* **450** (7170) 683-694
14. Alber F., Dokudovskaya S., Veenhoff L., et al. (2007) The molecular architecture of the nuclear pore complex. *Nature* **450** (7170) 695-701
15. Lasker K., Phillips J.L., Russel D., et al. (2010) Integrative Structure Modeling of Macromolecular Assemblies from Proteomics Data. *Mol Cell Proteomics* **9** 1689-1702
16. Russel D., Lasker K., Phillips J., et al. (2009) The structural dynamics of macromolecular processes. *Curr Opin Cell Biol* **21** 97-108
17. Alber F., Forster F., Korkin D., et al. (2008) Integrating diverse data for structure determination of macromolecular assemblies. *Annu Rev Biochem* **77** 443-477
18. Alber F., Chait B.T., Rout M.P., et al. (2008) Integrative Structure Determination of Protein Assemblies by Satisfaction of Spatial Restraints. In: Panchenko A, Przytycka T (eds) Protein-protein interactions and networks: identification, characterization and prediction. Springer-Verlag, London, UK, pp. 99-114
19. Bonvin A.M., Boelens R., Kaptein R. (2005) NMR analysis of protein interactions. *Curr Opin Chem Biol* **9** (5) 501-508
20. Fiaux J., Bertelsen E.B., Horwich A.L., et al. (2002) NMR analysis of a 900K GroEL GroES complex. *Nature* **418** (6894) 207-211
21. Neudecker P., Lundstrom P., Kay L.E. (2009) Relaxation dispersion NMR spectroscopy as a tool for detailed studies of protein folding. *Biophys J* **96** (6) 2045-2054
22. Takamoto K., Chance M.R. (2006) Radiolytic protein footprinting with mass spectrometry to probe the structure of macromolecular complexes. *Annu Rev Biophys Biomol Struct* **35** 251-276
23. Guan J.Q., Chance M.R. (2005) Structural proteomics of macromolecular assemblies using oxidative footprinting and mass spectrometry. *Trends Biochem Sci* **30** (10) 583-592

24. Taverner T., Hernandez H., Sharon M., et al. (2008) Subunit architecture of intact protein complexes from mass spectrometry and homology modeling. *Acc Chem Res* **41** (5) 617-627
25. Chen Z.A., Jawhari A., Fischer L., et al. (2010) Architecture of the RNA polymerase II-TFIIF complex revealed by cross-linking and mass spectrometry. *EMBO J* **29** (4) 717-726
26. Sinz A. (2006) Chemical cross-linking and mass spectrometry to map three-dimensional protein structures and protein-protein interactions. *Mass Spectrom Rev* **25** (4) 663-682
27. Trester-Zedlitz M., Kamada K., Burley S.K., et al. (2003) A modular cross-linking approach for exploring protein interactions. *J Am Chem Soc* **125** (9) 2416-2425
28. Joo C., Balci H., Ishitsuka Y., et al. (2008) Advances in single-molecule fluorescence methods for molecular biology. *Annu Rev Biochem* **77** 51-76
29. Mertens H.D., Svergun D.I. (2010) Structural characterization of proteins and complexes using small-angle X-ray solution scattering. *J Struct Biol*
30. Hura G.L., Menon A.L., Hammel M., et al. (2009) Robust, high-throughput solution structural analyses by small angle X-ray scattering (SAXS). *Nat Methods* **6** (8) 606-612
31. Schneidman-Duhovny D., Kim S.J., Sali A. (2012) Integrative structural modeling with small angle X-ray scattering profiles. *BMC Struct Biol* **12** 17
32. Berggard T., Linse S., James P. (2007) Methods for the detection and analysis of protein-protein interactions. *Proteomics* **7** (16) 2833-2842
33. Sali A., Blundell T.L. (1993) Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol* **234** (3) 779-815
34. Sali A., Blundell T. (1994) Comparative protein modeling by statisfaction of spatial restraints. In: Bohr H, Brunak S (eds) Protein Structure by Distance Analysis. Symposium on Distance Based Approaches to Protein Structure Determination. TECH UNIV DENMARK, CTR BIOL SEQUENCE ANAL, LYNGBY, DENMARK, pp. 64-86
35. Vajda S., Kozakov D. (2009) Convergence and combination of methods in protein-protein docking. *Curr Opin Struct Biol* **19** (2) 164-170
36. Shen M.Y., Sali A. (2006) Statistical potential for assessment and prediction of protein structures. *Protein Sci* **15** (11) 2507-2524
37. Melo F., Sanchez R., Sali A. (2002) Statistical potentials for fold assessment. *Protein Sci* **11** (2) 430-448
38. Brooks B.R., Brooks C.L., 3rd, Mackerell A.D., Jr., et al. (2009) CHARMM: the biomolecular simulation program. *J Comput Chem* **30** (10) 1545-1614
39. Case D.A., Cheatham T.E., 3rd, Darden T., et al. (2005) The Amber biomolecular simulation programs. *J Comput Chem* **26** (16) 1668-1688
40. Christen M., Hunenberger P.H., Bakowies D., et al. (2005) The GROMOS software for biomolecular simulation: GROMOS05. *J Comput Chem* **26** (16) 1719-1751
41. Forster F., Lasker K., Beck F., et al. (2009) An Atomic Model AAA-ATPase/20S core particle sub-complex of the 26S proteasome. *Biochem Biophys Res Commun* **388** 228-233

42. Nickell S., Beck F., Scheres S.H.W., et al. (2009) Insights into the Molecular Architecture of the 26S Proteasome. *Proc Natl Acad Sci U S A* **29** (104) 11943-11947
43. Lasker K., Forster F., Bohn S., et al. (2012) Molecular architecture of the 26S proteasome holocomplex determined by an integrative approach. *Proc Natl Acad Sci USA* **109** 1380-1387
44. Lasker K., Sali A., Wolfson H.J. (2010) Determining macromolecular assembly structures by molecular docking and fitting into an electron density map. *Proteins:Struct Funct Bioinform* **78** 3205-3211
45. 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** (1) 180-194
46. Webb B., Lasker K., Schneidman-Duhovny D., et al. (2011) Modeling of Proteins and their Assemblies with the Integrative Modeling Platform. In: Methods in Molecular Biology. Humana Press, pp. 377-397
47. Lensink M.F., Wodak S.J. (2010) Docking and scoring protein interactions: CAPRI 2009. *Proteins* **78** (15) 3073-3084
48. Ritchie D.W. (2008) Recent progress and future directions in protein-protein docking. *Curr Protein Pept Sci* **9** (1) 1-15
49. Schneidman-Duhovny D., Rossi A., Avila-Sakar A., et al. (2012) A Method for Integrative Structure Determination of Protein-Protein Complexes. *Bioinformatics* **28** 3282-3289
50. Reese M.L., Dotsch V. (2003) Fast mapping of protein-protein interfaces by NMR spectroscopy. *Journal of the American Chemical Society* **125** (47) 14250-14251
51. Rappsilber J. (2011) The beginning of a beautiful friendship: cross-linking/mass spectrometry and modelling of proteins and multi-protein complexes. *Journal of structural biology* **173** (3) 530-540
52. Berman H.M., Battistuz T., Bhat T.N., et al. (2002) The Protein Data Bank. *Acta Crystallogr D Biol Crystallogr* **58** (Pt 6 No 1) 899-907
53. Lawson C.L., Baker M.L., Best C., et al. (2011) EMDataBank.org: unified data resource for CryoEM. *Nucleic Acids Research* **39** (Database issue) D456-464
54. Pettersen E.F., Goddard T.D., Huang C.C., et al. (2004) UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem* **25** (13) 1605-1612
55. Velazquez-Muriel J.A., Lasker K., Russel D., et al. (2012) Assembly of macromolecular complexes by satisfaction of spatial restraints from electron microscopy images. *Proc Natl Acad Sci USA* **109** 18821-18826
56. Ritchie D.W., Venkatraman V. (2010) Ultra-fast FFT protein docking on graphics processors. *Bioinformatics* **26** (19) 2398-2405

Figures

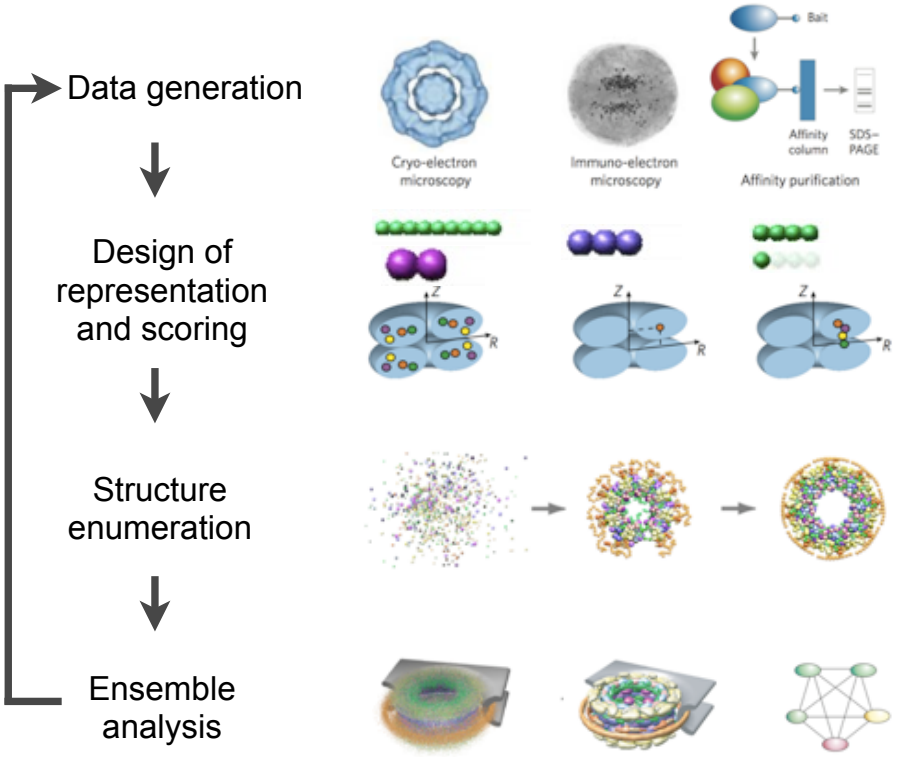

Figure 1. Integrative modeling protocol. After the datasets to be used are enumerated, a suitable representation is chosen for the system, and the input information is converted into spatial restraints. Models are generated that are optimally consistent with the input information by optimizing a function of these restraints. Analysis of the resulting models informs about the model and data accuracy and may help guide further experiments. The protocol is demonstrated with the construction of a bead model of the NPC*(13)*.

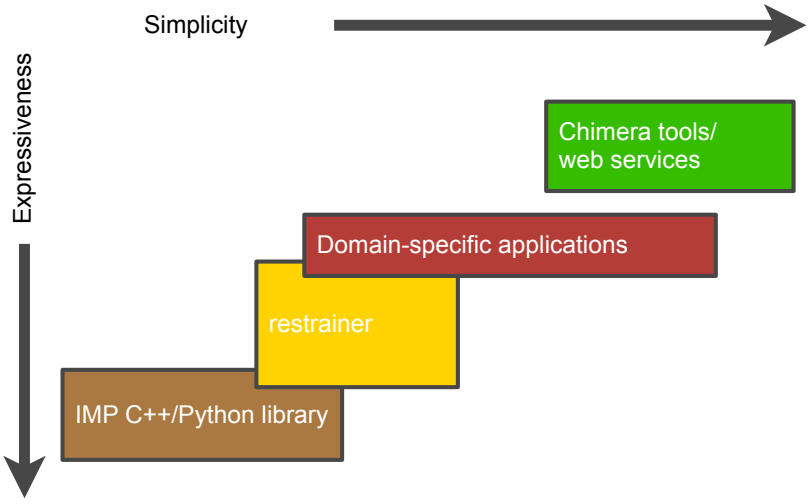

Figure 2. Overview of the IMP software. Components are displayed by simplicity (or user-friendliness) and expressiveness (or power). The core C++/Python library allows protocols to be designed from scratch, at a cost of user friendliness; higher-level modules and applications provide more user-friendly interfaces, at a cost of flexibility.

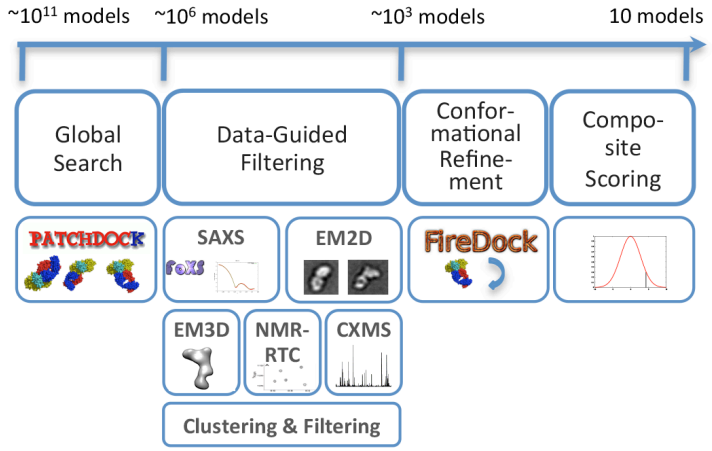

Figure 3. Schematic representation of the integrative docking method. The number of possible configurations for two docked proteins is on the order of ~10^{11} (three rotational degrees of freedom sampled at 5° intervals and three translational degrees of freedom sampled at 1Å intervals). As the method proceeds, the number of considered configurations decreases.

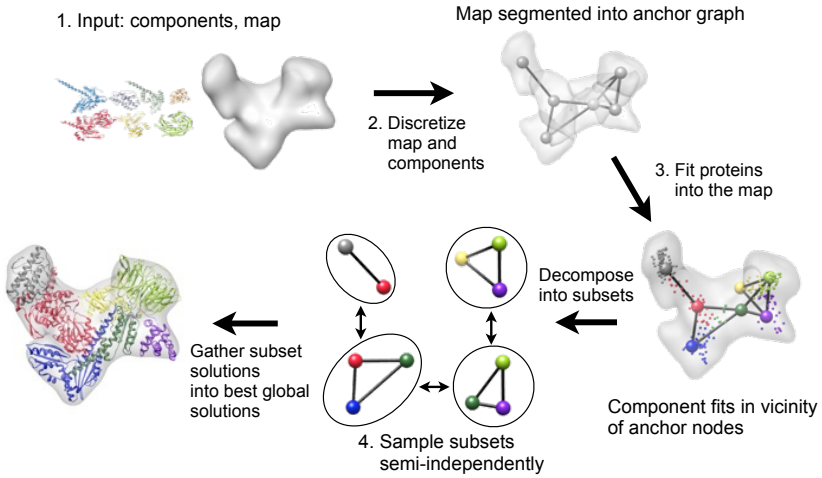

Figure 4. The MultiFit protocol*(45)*. Protein subunits are fitted into a density map of the assembly by discretizing both the map and the components, locally fitting each protein, and efficiently combining the local fits into global solutions.

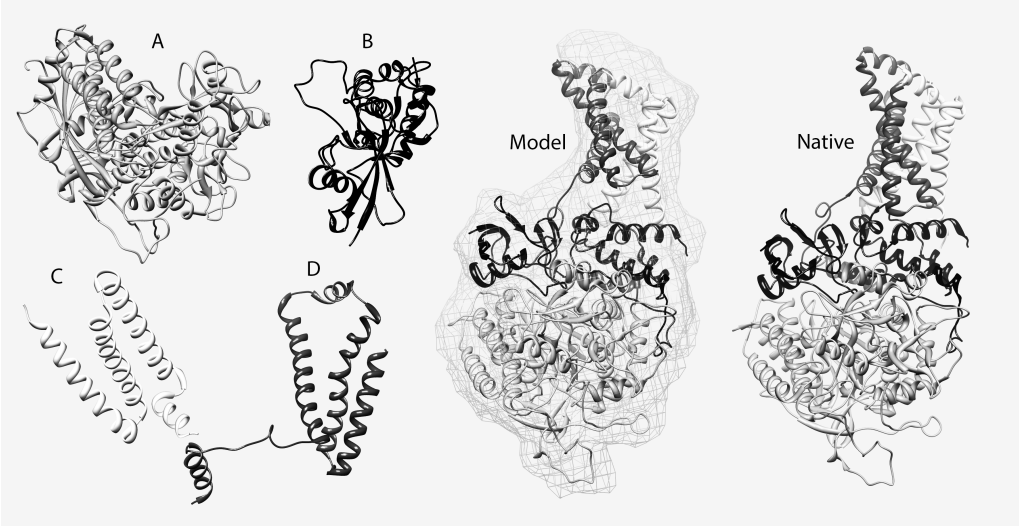

Figure 5. Results from the application of EMageFit to the macromolecular complex with PDB id 3sfd (porcine mitochondrial respiratory complex II). Left: The four subunits of the complex, each labeled with their chain identifier. Center: Model for the complex fitted into the simulated density map of the native configuration. Right: Native configuration of the complex as stored in the PDB file.