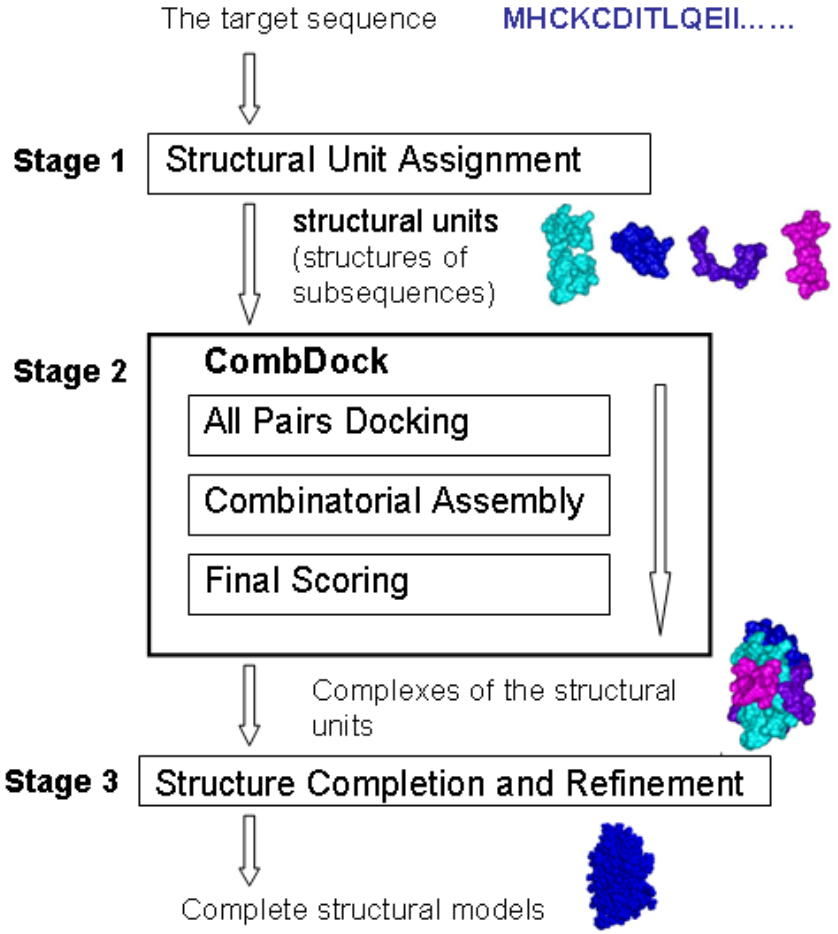
V3 Predicting Structures of Protein Complexes from Connectivities

`CombDock:` automated approach for predicting 3D structure of heterogenous multimolecular assemblies.

Input: structures of *N* individual proteins

Problem appears more difficult than the pairwise docking problem.

Idea: exploit additional geometric constraints that are part of the combinatorial problem.



Haim Wolfson
Tel Aviv University
http://www.cs.tau.ac.il/~wolfson/

Inbar et al., J. Mol. Biol. 349, 435 (2005)

Review: pairwise docking: Katchalski-Kazir algorithm

Discretize proteins A and B on a grid.

Every node is assigned a value

$$f_{A_{l,m,n}} = \begin{cases} 1 & : \text{ surface of molecule} \\ \rho & : \text{ core of molecule} \\ 0 & : \text{ open space} \end{cases}$$

and

$$f_{B_{l,m,n}} = \begin{cases} 1 & : \text{ inside molecule} \\ 0 & : \text{ open space} \end{cases}$$

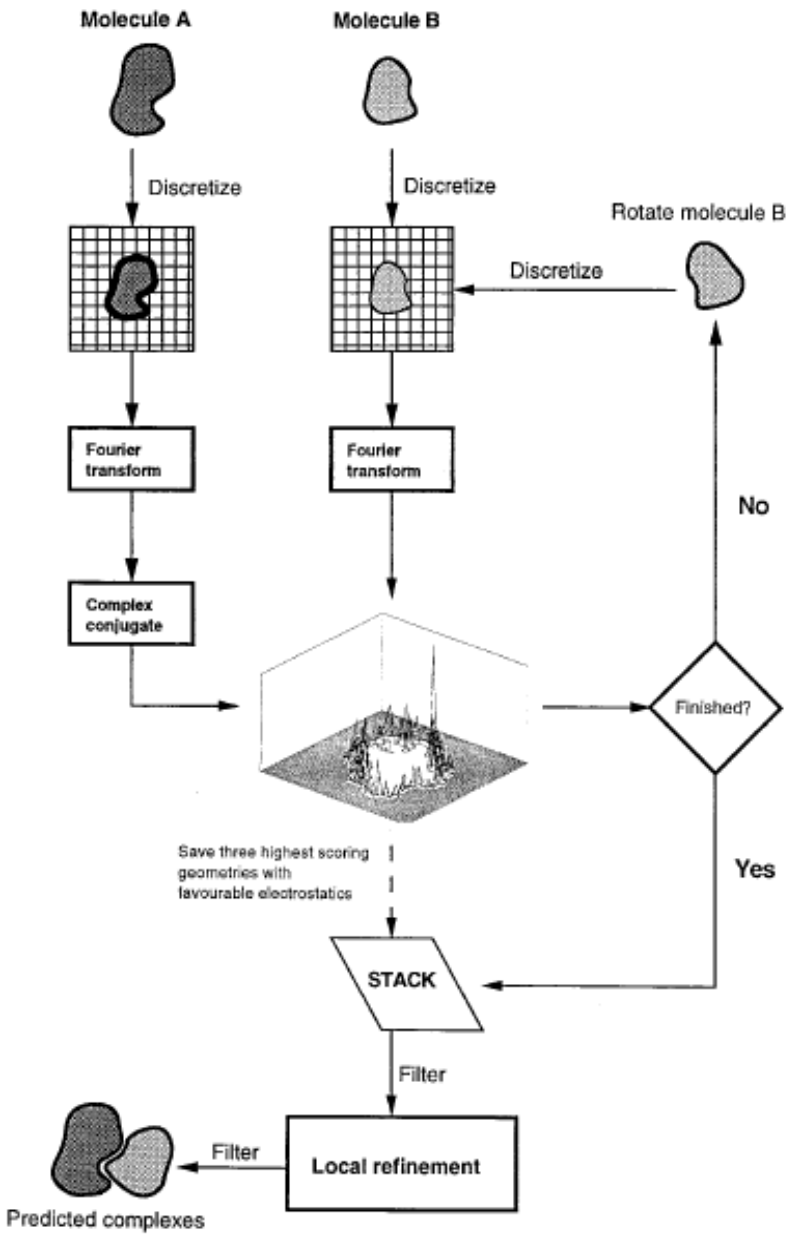
The correlation function of f_A and f_B is:

$$f_{C_{\alpha,\beta,\gamma}} = \sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} f_{A_{l,m,n}} \times f_{B_{l+\alpha,m+\beta,n+\gamma}}$$

Use FFT to compute correlation efficiently.

Output: solutions with best surface complementarity.

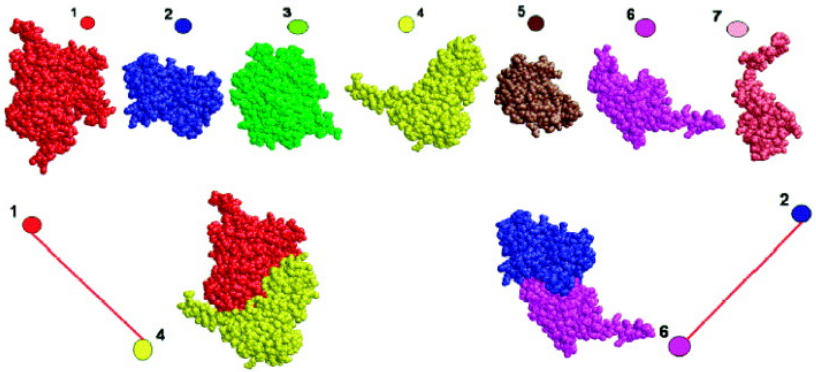
Gabb et al. J. Mol. Biol. (1997)



(1) All pairs docking module

Aim: predict putative pairwise interactions

Based on the *N* individual protein structures perform pairwise docking for each of the *N* (*N* - 1) / 2 pairs of proteins



Since the correct scoring of pairwise-docking is difficult, the correct solution may be among the first few hundred solutions.

→ keep *K* best solutions for each pair of proteins.

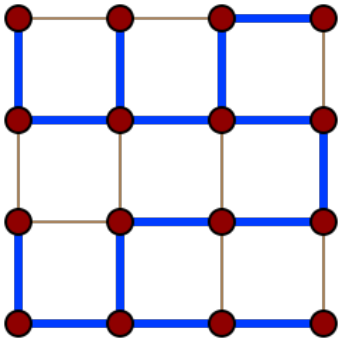
Here, *K* was varied from dozens to hundreds.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

(2) Combinatorial assembly module

Input: *N* subunits and *N* (*N* - 1) / 2 sets of *K* scored transformations.
These are the candidate interactions.

Reduction to a spanning tree



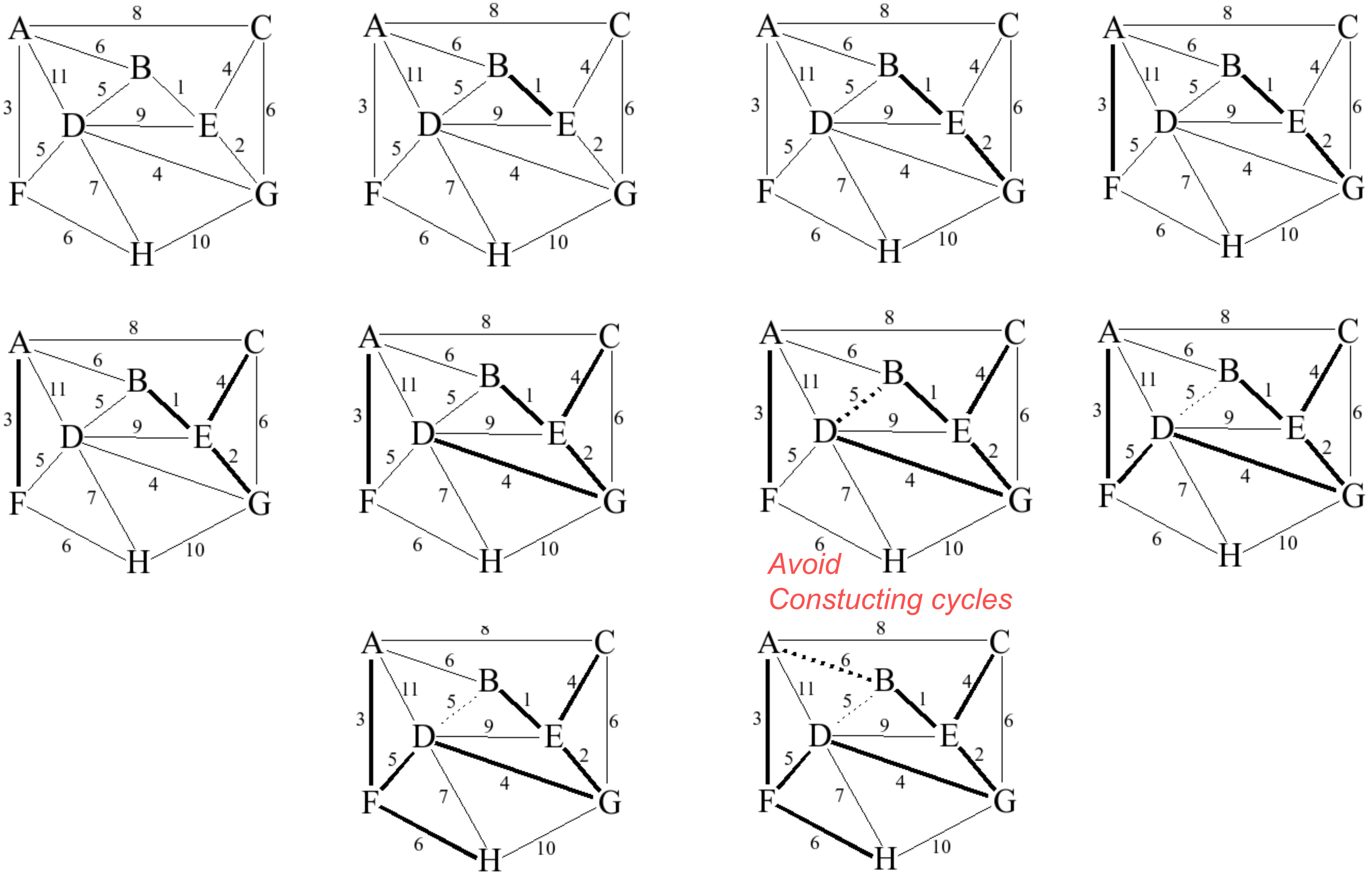
Spanning tree = a graph that connects all vertices and has no circles

Build weighted graph representing the input:
- each protein structure = vertex
- each transformation (docking orientation)
  = edge connecting the corresponding vertices
- edge weight = docking score of the transformation

→ Since the input contains *K* transformations for each pair of subunits, we get a complete graph with ***K* parallel edges** between each pair of vertices.

Inbar et al., J. Mol. Biol. 349, 435 (2005)
www.wikipedia.org

Review: Spanning tree – algorithm of Kruskal



(2) Combinatorial assembly module

For 2 subunits, each candidate binary docking complex is represented by an **edge** and the 2 vertices.

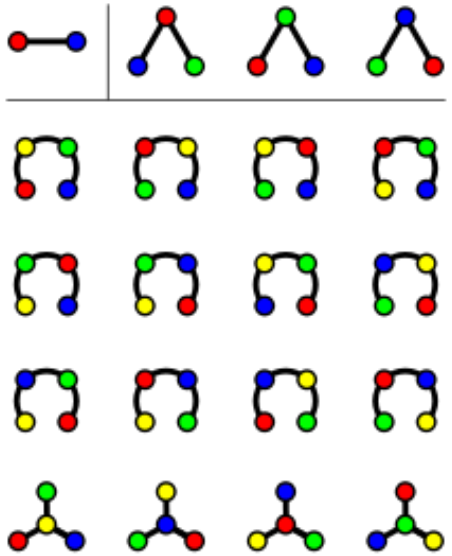
For the full complex, a candidate complex is represented by a **spanning tree**. Each spanning tree of the input graph represents a particular **3D structure** for the complex of all input structures.

→ Problem of finding 3D structures of complexes is equivalent to finding spanning trees.

The number of spanning trees in a complete graph with *N* nodes and **no parallel edges** is N^{N-2} (Cayley's formula).

Here, the input graph has *K* parallel edges between each pair of vertices. → the number of spanning trees is $N^{N-2}\ K^{N-1}$.

→ Exhaustive searches are infeasible!



Cayley's formula (the number of different trees on *n* vertices is n^{n-2}, graphically demonstrated for graphs with 2, 3 and 4 nodes.

Inbar et al., J. Mol. Biol. 349, 435 (2005)
www.wikipedia.org

(2) Combinatorial assembly module:algorithm

`CombDock` algorithm uses 2 basic principles:

(1) hierarchical construction of the spanning tree

(2) greedy selection of subtrees

$\rightarrow$ 2 subtrees of smaller size (that were previously generated) are connected with an input edge to generate trees with *i* vertices

In this way, the common parts of different trees are generated only once.

When connecting subtrees, check whether there are severe **penetrations** between pairs of subunits that are represented by different subtrees.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

(2) Combinatorial assembly module:algorithm

Stage 1: algorithm start with trees of size 1.
Each tree contains a single vertex that represents a subunit.

Stage *i*: the tree complexes that consist of exactly *i* vertices (subunits) are generated by connecting 2 trees generated at a lower stage with an input edge transformation.

Tree complexes that fulfil the penetration constraint are kept for the next stages.

Because it is impractical to search all valid spanning trees, the algorithm performs a greedy selection of subtrees.

For each subset of vertices, the algorithm keeps only the *D* best-scoring valid trees that connect them.

The **tree score** is the sum of its edge weights.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

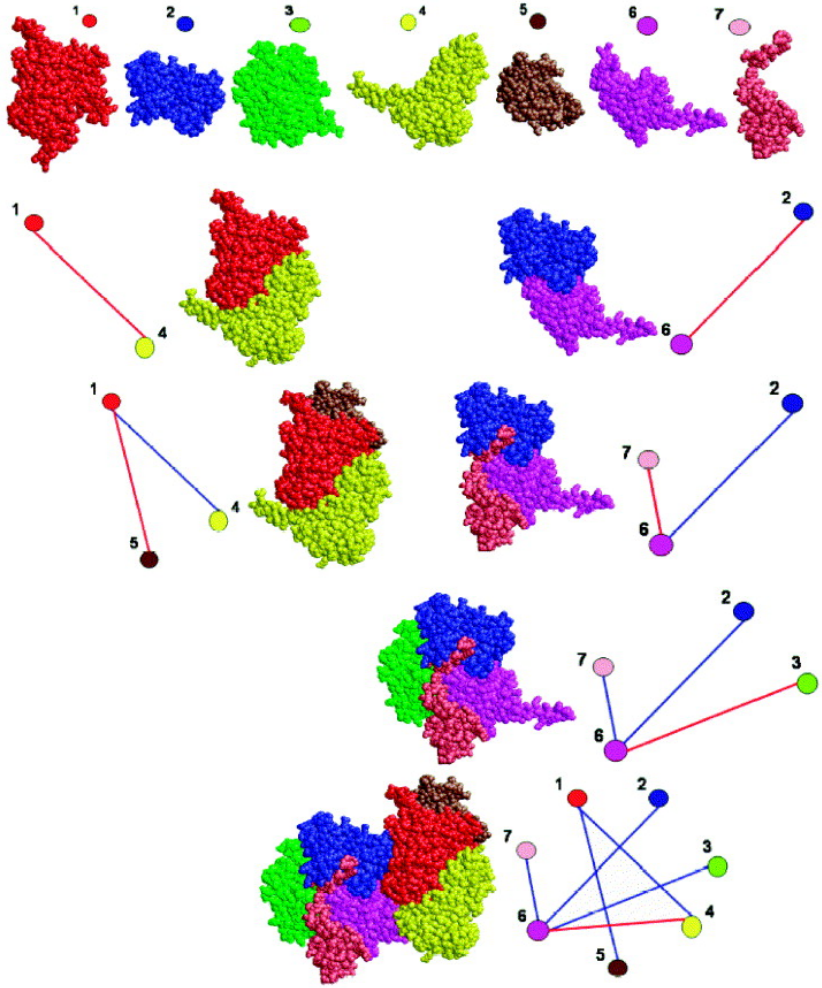
Example: arp2/3 complex

The arp2/3 complex consists of 7 subunits (top).

Shown are only the complexes of the different stages that were relevant to the construction of the third-best scoring solution with RMSD 1.2 Å (bottom).

Red edge: transformation of the current stage,

Blue edges: transformations of previous stages.



Inbar et al., J. Mol. Biol. 349, 435 (2005)

Final scoring

A **geometric score** evaluates the shape complementarity between the subunits:

- check distances between surface points on adjacent subunits.

- close surface points increase score,

- penetrating surface points decrease score.

Physico-chemical component of the final score counts all surface points that belong to non-polar atoms = this gives an estimate of the hydrophobic effect.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

Clustering of solutions

Clustering of solutions:

(1) compute **contact maps** between subunits: array of N (N – 1) bins.

If two subunits are in contact within the complex,
set the corresponding bit to 1, and to 0 otherwise.

(2) superimpose complexes that have the same contact map
and compute RMSD between C^{α} atoms.

If this distance is less than a threshold, consider complexes
as members of a **cluster**.

For each cluster, keep only the complex with the highest score.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

Performance for known complexes

Table 1. *CombDock* multimolecular assembly test cases

Target complex (PDB)	Bound/ unbound	Input			Output		
		No. SUs	Complex size	SU avg. size	**RMSD** (Å) [rank]	Complexes pre/post clustering	Run time HH:MM:SS
Nf-kappa-b p65 subunit (1ikn)	Bound	3	698	233	**1.8 [1]**	1000/49	00:38
	Unbound	3	698	233	**1.9 [6]**	3655/40	00:24
Vhl/ElonginC/ ElonginB (1vcb)	Bound	3	328	109	**0.5 [2]**	406/14	00:17
	Unbound	3	272	91	**1.0 [4]**	152/10	00:15
Arp2/3 complex (1k8k)	Bound	7	1709	244	**1.2 [3]**	5488/145	28:59
	Unbound	7	1728	246	**1.9 [10]**	3475/110	26:09
RNA polymerase II (1i6h)	Bound	10	3519	352	**1.4 [1]**	50,188/1113	15:27:58
	Unbound	10	3576	357	**1.3 [4]**	50,100/1264	15:20:17
MHCII/TCR/Sep3	Unbound	3	1030	343	**3.9 [3]**	1161/25	01:24

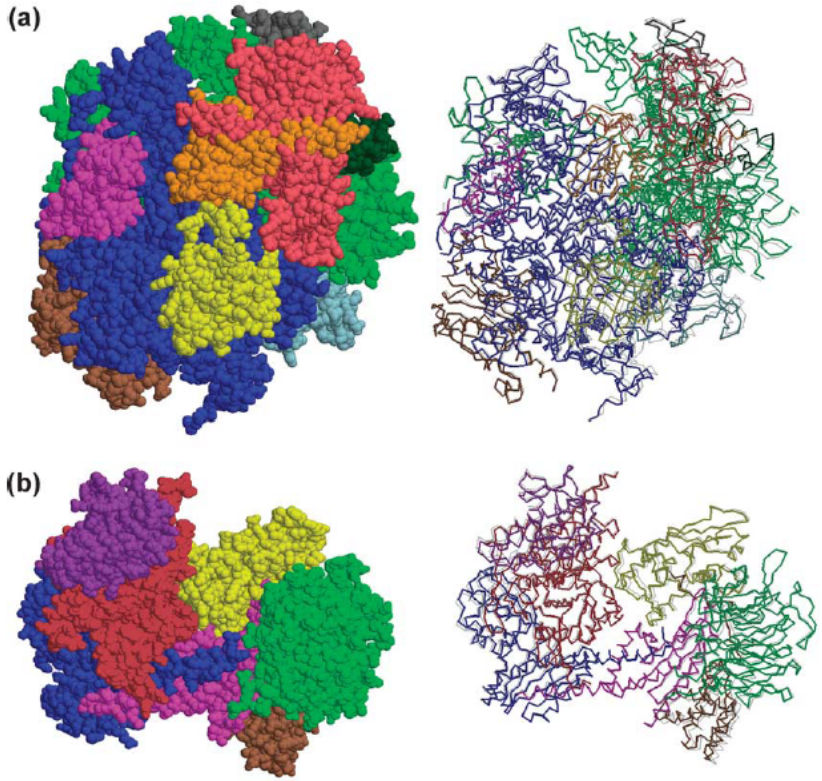
SU, subunit; avg., average; the run time refers to the time of the combinatorial assembly module, running on a Linux machine with a 1 GHz single processor. For the unbound cases, the RMSD distances were calculated between all the C^{α} atoms of the predicted complex and a reference complex that was generated by superimposing the input unbound subunits on the corresponding bound subunits of the determined structure.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

Examples of large complexes

CombDock solution

solution superposed on the crystal structure (gray thiner lines)



(a) the bestranked complex of the 10 subunits of **RNA polymerase II**, RMSD 1.4 Å.

(b) the third-best scoring assembly of the 7 subunits of the **arp2/3 complex**, RMSD 1.2 Å.

CombDock is not as succesful for docking „unbound“ subunit structures that structurally differ from „bound“ conformations.

Inbar et al., J. Mol. Biol. 349, 435 (2005)

DockStar: overcome limitations of CombDock

Bioinformatics, 31(17), 2015, 2801–2807
doi: 10.1093/bioinformatics/btv270
Advance Access Publication Date: 25 April 2015
Original Paper

Structural bioinformatics

DockStar: a novel ILP-based integrative method for structural modeling of multimolecular protein complexes

Naama Amir*, Dan Cohen and Haim J. Wolfson*

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel

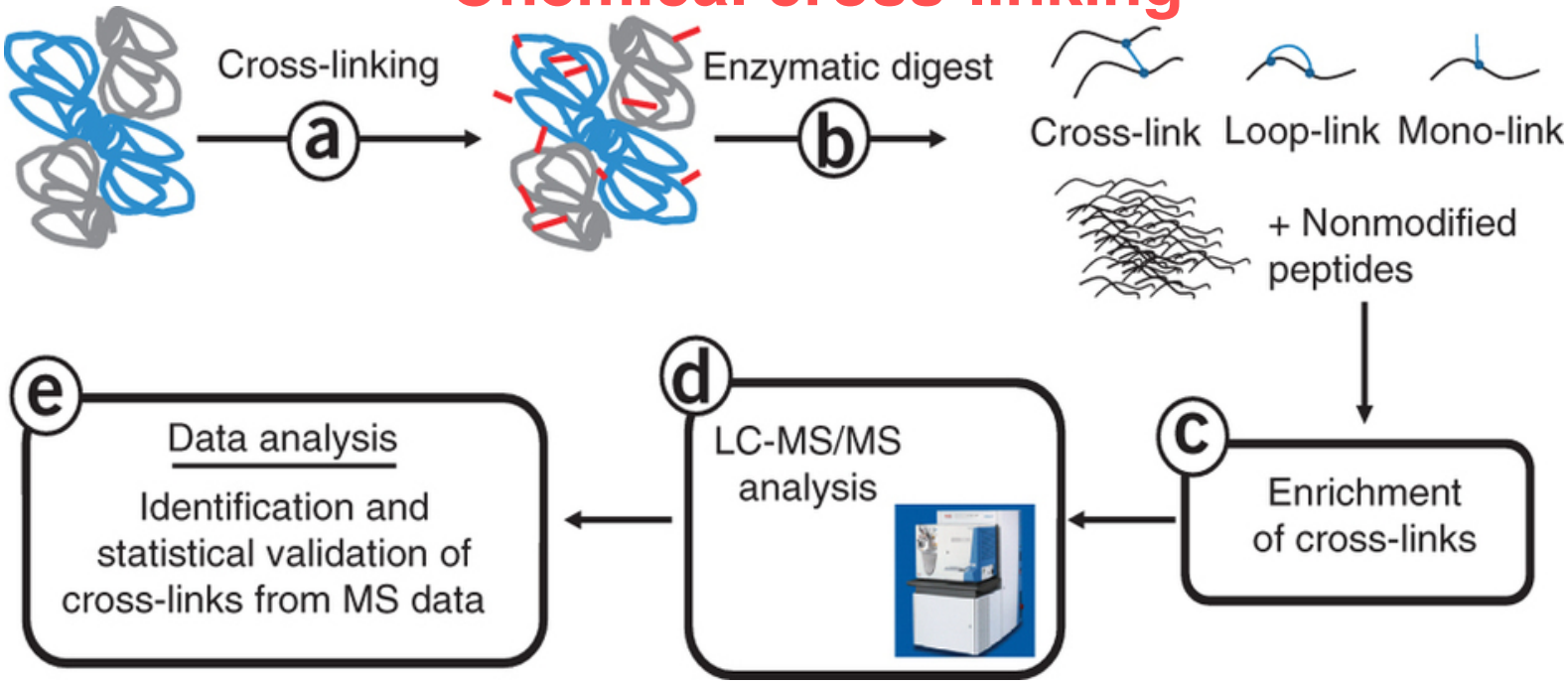
2 subtasks for generation of macromolecular complex structures:

(a) Identify the protein-protein **interaction graph** between the individual subunits; use additional data from **chemical cross-linking** for this,

(b) Detect a globally consistent **pose** of the subunits, so that

- there are no steric clashes between them and
- the binding energy of the whole complex is optimized.

Chemical cross-linking



(**a**) cross-linking reaction using a chemical cross-linking reagent. These molecules have a certain length, have two reactive groups at both ends of the molecule and may covalently bind either to cysteine or lysine residues of a single protein or of two proteins.

(**b**) enzymatic digestion of the proteins to peptides,

(**c**) enrichment of cross-linked peptides,

(**d**) analysis of cross-linked peptides by LC-MS/MS,

(**e**) data analysis.

Leitner et al. Nature Protocols 9, 120–137 (2014)

StarDock

- MS of intact protein complexes and their subcomplexes (→TAP-MS) can determine the **stoichiometry** of the complex subunits and deduce the **interaction graph** of the multimolecular complex.

- Chemical cross-linking combined with MS provides **distance constraints** between surface residues both on the same and on neighboring subunits.

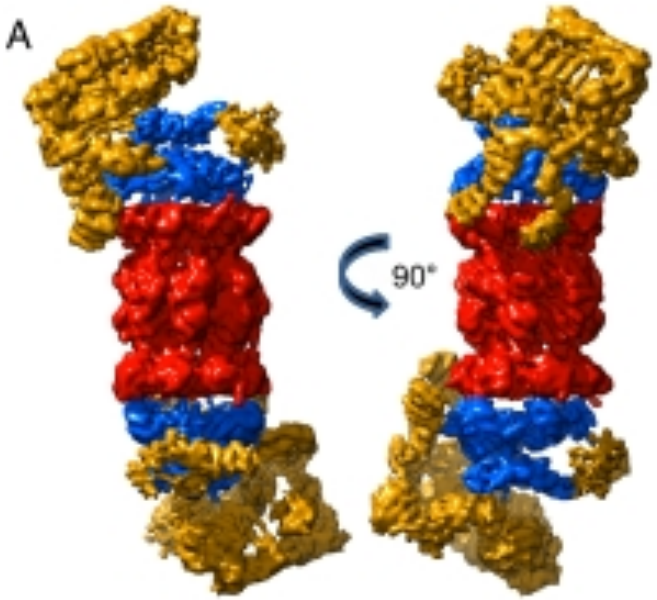
This provides information both for the detection of the interaction graph as well as constraints on the relative spatial poses of neighboring subunits.

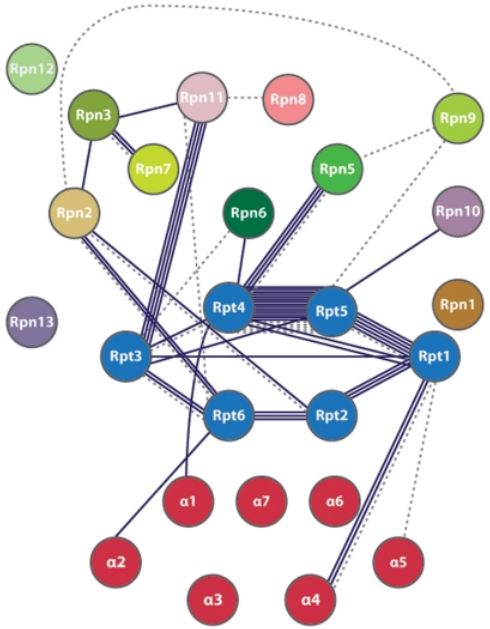
Such constraints have been successfully e.g. exploited in the modeling of the

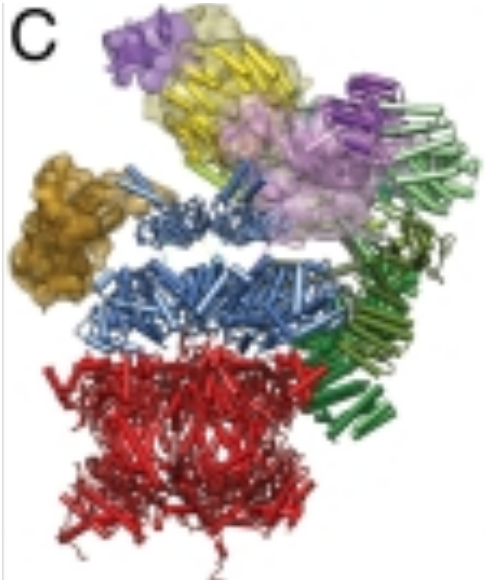
- 26S proteasome,
- the proteasome lid,
- the TRiC/CCT chaperonin,
- the RNA polymerase II–TFIIF complex and more.

Amir et al., Bioinformatics 31, 2801 (2015)

Iterative refinement of the 3D structure of S26 proteasome







Low resolution

EM structure

Chemical cross-links for the *S. pombe* and *S. cerevisiae* 26S proteasomes.

55 (21) pairs of cross-linked lysines from the *S. pombe* (*S. cerevisiae*) 26S proteasome subunits.

Multiple edges between a pair of subunits indicate multiple cross-linked lysine pairs.

Atomistic structure generated

Lasker et al., PNAS (2012) 109: 1380

StarDock: Generate transformation sets

Assume that the **interaction graph** is **known** (task A).

Generate for each subunit a set of candidate rigid transformations.

One subunit is chosen as an **anchor subunit**. Preferably, the anchor subunit should have the largest number of neighbors in the multimolecular assembly interaction graph. All other subunits which are known to interact with the anchor are then docked to it.

This requires a **star shaped spanning tree** topology of the interaction graph.

Pairwise docking is carried out by `PatchDock`, which optimizes shape complementarity, while satisfying maximal distance constraints between residues of neighboring subunits from cross-linking.

The top 1000 `PatchDock` transformations are refined, rescored and re-ranked by the `FiberDock` tool

-> pairwise scores

Amir et al., Bioinformatics 31, 2801 (2015)

StarDock: Select best global solution

For each of the n subunits, let

- P_i $(0 \leq i < n)$ be **subunit** *i,*
- $T(P_i)$ be the set of **candidate transformations** received from the previous stage for subunit P_i.
- $T_{i,r}$ be a particular **transformation** r of subunit P_i .
- $S(T_{i,r}, T_{j,s})$ be the **pairwise interaction score** of subunits P_i and P_j transformed by $T_{i,r}$ and $T_{j,s}$, respectively (obtained by pairwise docking before).

The **globally optimal solution** Sol includes one transformation per subunit and maximizes the score(Sol) defined as:

$$\text{score}(\text{Sol}) = \sum_{T_{i,r}, T_{j,s} \in \text{Sol} \cap i \neq j} S(T_{i,r}, T_{j,s})$$

Amir et al., Bioinformatics 31, 2801 (2015)

DockStar: Select best global solution

This optimization task can be formulated as the following graph theoretic problem:

Let G = (V,E) be an undirected *n*-partite graph with a partition of the vertex set
$V = V_0 \cup \ldots \cup V_{n-1}$,
so that each transformation $T_{i,r} \in T(P_i)$ corresponds to a vertex $u_{i,r} \in V_i$.
(Each V_i contains all transformations r of subunit P_i as its vertices $u_{i,r}$.)

Each pair of vertices is joined by an edge:

$$E = \{(u_{i,r}, v_{j,s}) | u_{i,r} \in V_i; v_{j,s} \in V_j; i \neq j\}$$

with the weight $w(u_{i,r}, v_{j,s}) = S(T_{i,r}, T_{j,s}) \qquad \forall (u_{i,r}, v_{j,s}) \in E$

The optimal solution is achieved by choosing one vertex per V_i
that maximizes the edge-weight of the induced sub-graph.

Amir et al., Bioinformatics 31, 2801 (2015)

Formulate Integer Linear Program (ILP)

This graph theoretic task can be formulated as an ILP. Define a variable $X_{i,r}$ for each vertex $u_{i,r} \in V$ and a variable $Y_{i,r,j,s}$ for each edge $e(u_{i,r}, v_{j,s}) \in E$ as follows

$$X_{i,r} = \begin{cases} 1 & \text{if } u_{i,r} \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{i,r,j,s} = \begin{cases} 1 & \text{if both } u_{i,r} \text{ and } v_{j,s} \text{ are chosen} \\ 0 & \text{otherwise} \end{cases}$$

The ILP **objective function** is

$$\text{Maximize} \qquad \text{score}(\text{Sol}) = \sum_{(u_{i,r}, v_{j,s}) \in E} w(u_{i,r}, v_{j,s}) Y_{i,r,j,s}$$

Subject to the constraints:

$$\sum_{u_{i,r} \in V_i} X_{i,r} = 1 \qquad \forall i, 0 \leq i < n$$

$$\sum_{u_{i,r} \in V_i} Y_{i,r,j,s} = X_{j,s} \qquad \forall j, s, i, \quad j \neq i$$

Amir et al., Bioinformatics 31, 2801 (2015)

The objective function is exactly the edge-weight of the chosen sub-graph. The first constraint ensures that exactly one transformation is chosen for each subunit. The second constraint ensures that an edge is chosen if and only if both vertices that it connects are chosen as well.
The ILP step was solved by the CPLEX 12.5 package

ILP formulation – alternative solutions

The ILP method outputs one single highest scoring global solution.

To retrieve additional high scoring solutions, the ILP step is applied iteratively to find a solution that maximizes the objective function and was not chosen before.

For this, a **linear constraint** is used (see paper by Amir et al.).

Amir et al., Bioinformatics 31, 2801 (2015)

ILP formulation – alternative solutions

Sofar we considered complexes having a **star shaped spanning tree**, where an **anchor** subunit, which interacts with all the other subunits, can be chosen. However, this is a special case.

Arbitrary complexes are divided into overlapping sub-complexes, each with a star shaped spanning tree, which are solved separately as above.

Then, top solutions of subcomplexes that share a subunit are merged, while defining the shared subunit as the new ‘anchor’.

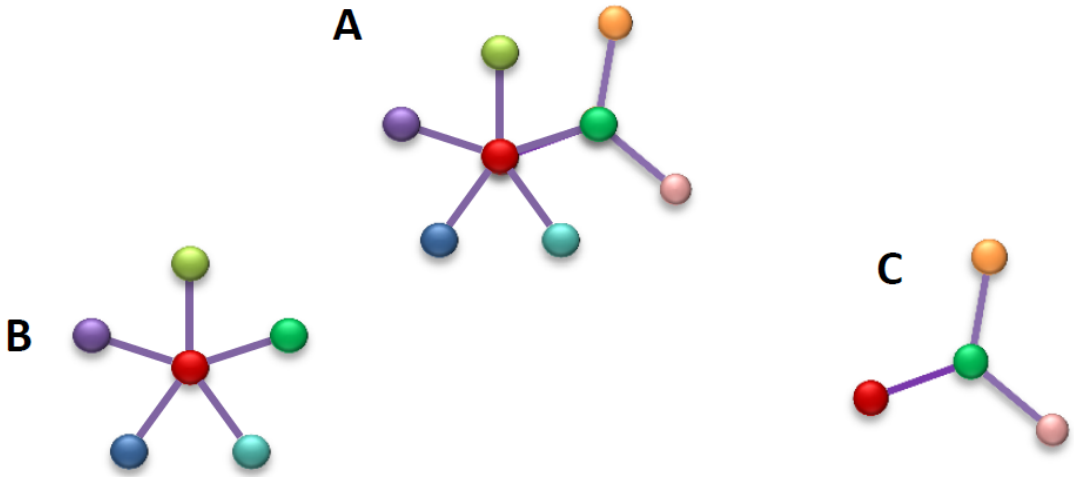
All the transformations in the merged (new) subcomplex are recalculated vis-a-vis the reference frame of the new ‘anchor’.

These new transformation sets are used as input for steps 2–4 of the algorithm in order to solve the larger sub-complex.

In several such iterations one can cover all the subunits of the assembly.

Amir et al., Bioinformatics 31, 2801 (2015)

ILP formulation – alternative solutions



(A) A complex interaction graph that is not star shaped. Therefore, the complex is divided to two sub-complexes and each sub-complex structure is solved separately. The transformation set for each subunit is generated by docking the subunit to the "anchor" subunit.

In (B) the anchor is represented by the red vertex and in (C) by the green. For each sub-complex a set of solutions is generated. Then, top solutions of these sub-complexes are integrated to create the 3D structure of the whole complex.

Amir et al., Bioinformatics 31, 2801 (2015)

DockStar applications

Table 1. Summary of the DockStar's results

Target complex	Bound/ unbound	Subunits number	Rank	Global Cα-RMSD[a]	Number of contacts[b]	Quality of predicted contacts[c]				Run time HH:MM
						high	medium	acceptable	lenient	
PP2A	Bound	3	1	0.68	2	2	0	0	0	00:35
	Unbound	3	1	6.9	2	0	0	0	2	00:43
Beef liver	Bound	4	1	0.85	3	3	0	0	0	02:51
Catalase	Unbound	4	1	2.7	3	0	3	0	0	03:53
RNA polII	Bound	11	1	7.9	10	4	3	2	0	04:53
	Unbound	11	3	4.8	10	0	3	4	1	04:56
Yeast exosome	Bound	10	1	5.1	9	6	1	0	0	10:34
	Unbound	10	12	6.0	9	1	1	1	1	11:22

[a]Global Cα-RMSD between the predicted and the native assemblies including only predictions with lenient to high quality.
[b]Number of contacts in the spanning tree of the complex interaction graph.
[c]Predicted interfaces in the target complex that are of lenient to high quality.

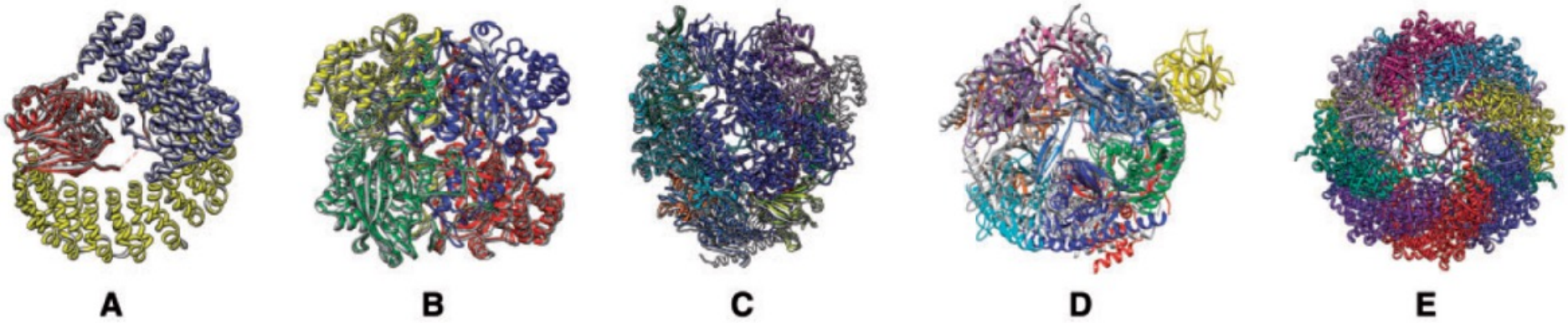


Fig. 1. The predicted models of the bound cases (coloured by chains) superimposed on the correct complex structures taken from the PDB (grey). (**A**) PP2A (A(yellow), B(blue), C(red)), (**B**) The Beef Liver Catalase [A(yellow), B(blue), C(red), D(green)], (**C**) RNA polymerase II [Rbp1(blue), Rbp2(cyan), Rbp3(light blue), Rbp5(purple), Rbp6(green), Rbp7(pink), Rbp8(yellow), Rbp9(dark green), Rbp10(orange), Rbp11(brown), Rbp12(red)], (**D**) The Yeast Exosome [Rrp45(blue), Rrp41(cyan), Rrp43(light blue), Rrp46(green), Rrp42(purple), Mtr3(pink), Rrp40(red), Rrp4(orange), Csl4(yellow), Dis3(dark green)]. (**E**) The predicted order of chains in the model of the TRiC/CCT Chaperonin: Z(red) Q(blue) H(yellow) E(light blue) B(pink) D(grey) A(green) G(purple)

Amir et al., Bioinformatics 31, 2801 (2015)

Mosaic-3D

Input:

(1) high-resolution three-dimensional **structures** of a representative of each protein involved in forming the complex

(2) information on the **stoichiometry** of the complex.

(3) information on pairwise **interfaces** that provide the presumed binding modes in the complex.

Output:

3D-MOSAIC then assembles the complex in an iterative tree-based greedy fashion.

Similar to CombDock, each node represents a monomer attached in a particular orientation.

Dietzen, Kalinina, Lengauer, Hildebrandt *et al.,*
Proteins 83, 1887-1899 (2015)

Mosaic-3D

The algorithm starts from a seed monomer with the largest number of interfaces.

In each iteration, new child solutions are generated by adding an additional monomer to each of the parent solutions retained from the previous iteration.

A new monomer of a particular protein type *p* can be attached to the complex *r* of a previous stage, if
i) the number of occurrences of *p* in the parent solution has not yet reached its maximum multiplicity,
ii) *r* has unoccupied interfaces for an interaction with *p*.
iii) The new monomer does not lead to severe steric clashes with other monomers already present in the parent solution.

The new child monomer is scored according to the number of interfaces it has with all ancestor monomers already present in the complex.

After each iteration: cluster solutions based on C_α-RMSD
Finally: optimize symmetry

Dietzen *et al,* Proteins 83, 1887-1899 (2015)

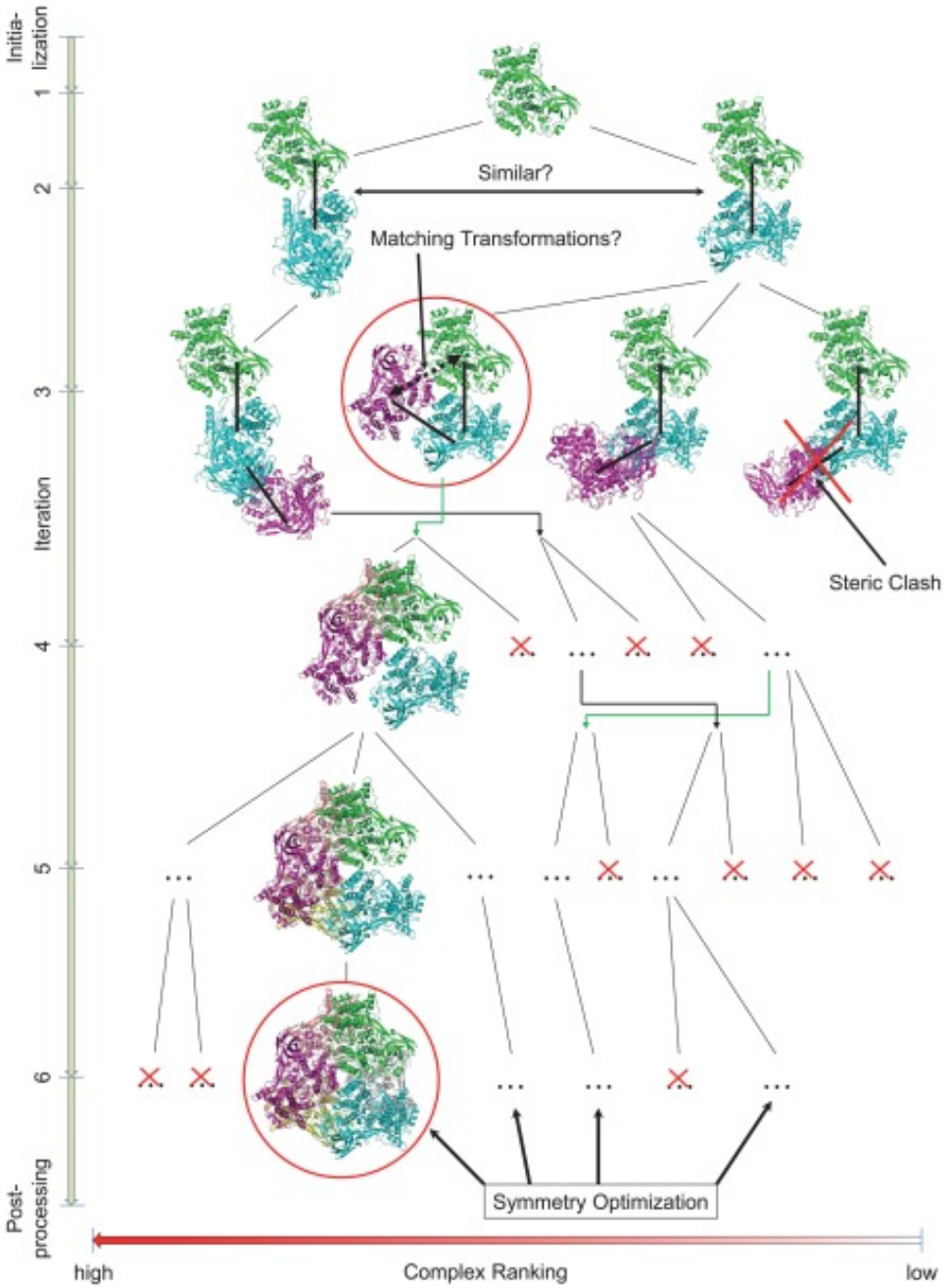
Workflow

Assembly of homo-hexameric hemocyanin from *Panulirus interruptus* (PDB code 1HCY) using 3D-MOSAIC.

In each iteration, new monomers can be attached to all previously retained solutions.

If a matching interface is found, the complex match score increases and the corresponding complex might be ranked further up in the list of solutions (green double-tilted arrows).

Solutions similar to better-ranked ones or yielding severe steric clashes are discarded.
After complex construction, a symmetry optimization can be performed.

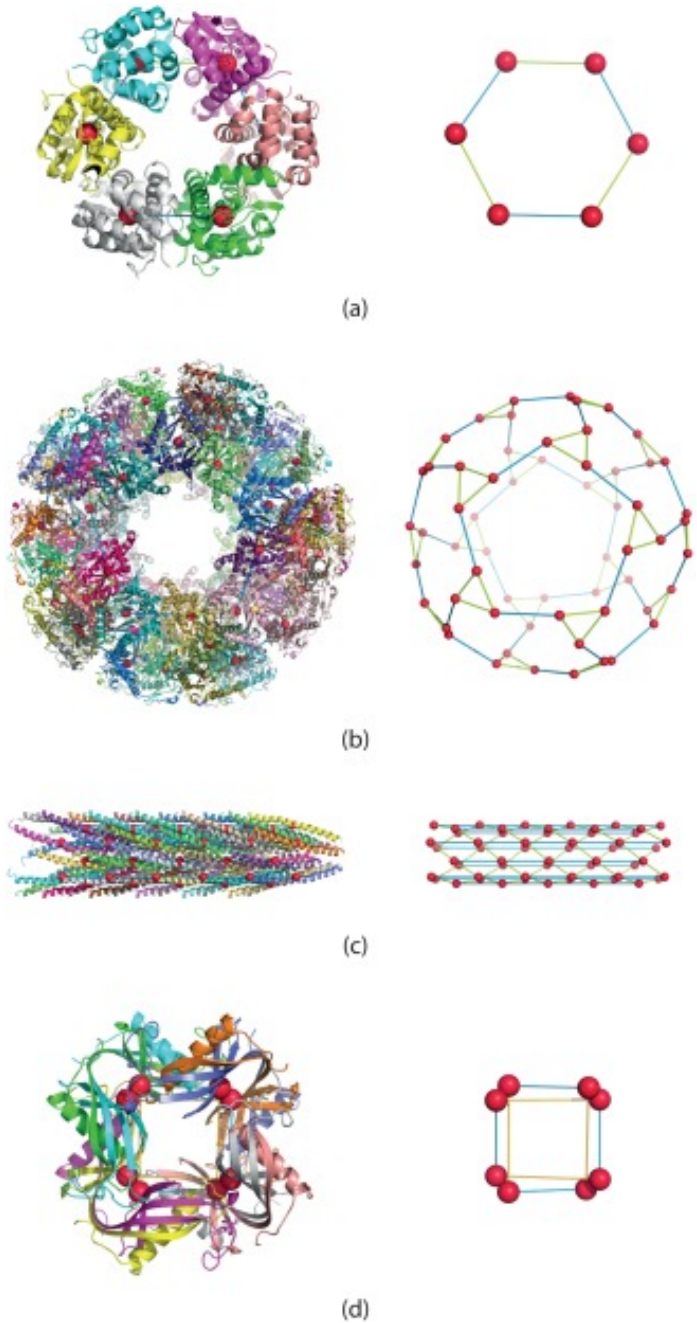


Dietzen et al, Proteins 83, 1887-1899 (2015)

Mosaic-3D

Examples of complexes and corresponding topology graphs for hard cases:
(**a**) ring-like topology of T4 lysozyme hexamer (3SBA),
(**b**) cage-like topology of pyruvate dehydrogenase E2 60-mer core complex (1B5S),
(**c**) inovirus coat protein filament (2C0W) composed of helical monomers,
(**d**) human cystatin C complex (1R4C) forming interchain β-sheets. Different node colors correspond to different protein types, different edge colors to different binding modes.

On a diverse benchmark set of 308 homo and heteromeric complexes containing 6 to 60 monomers, the mean fraction of correctly reconstructed benchmark complexes during crossvalidation, was 78.1%.



Dietzen et al, Proteins 83, 1887-1899 (2015)

Summary

Our current atomistic understanding of how large macromolecular machines work is mainly based on results from protein crystallography. These discoveries were rewarded with several Nobel Prizes in Chemistry and Medicine.

Recent breakthrough: new detectors for EM that improve its resolution down to atomic resolution.

Ideal for structural characterization of large multi-protein complexes: combination of methods in structural biology:
- X-ray crystallography and NMR for high-resolution structures of single proteins and pieces of protein complexes
- (cryo) EM to determine high- to medium-resolution structures of entire protein complexes
- stained EM for still pictures at medium-resolution of cellular organells and
- (cryo) electron tomography for three-dimensional reconstructions of biological cells and for identification of the individual components.

Dietzen et al, Proteins 83, 1887-1899 (2015)

Summary

When aiming at **integrating** the results from different methods,
e.g. by density fitting and by incorporating additional biochemical or
bioinformatics data as restraints during structural modelling,
this requires important contributions from **computational methods.**

Dietzen et al, Proteins 83, 1887-1899 (2015)