Introduction à Rosetta

dirk.stratmann@sorbonne-universite.fr

Oct 2023

Rosetta Commons

The hub for Rosetta modeling software

Search

Home | Software | Documentation & Support | Developer Resources | About | Blog

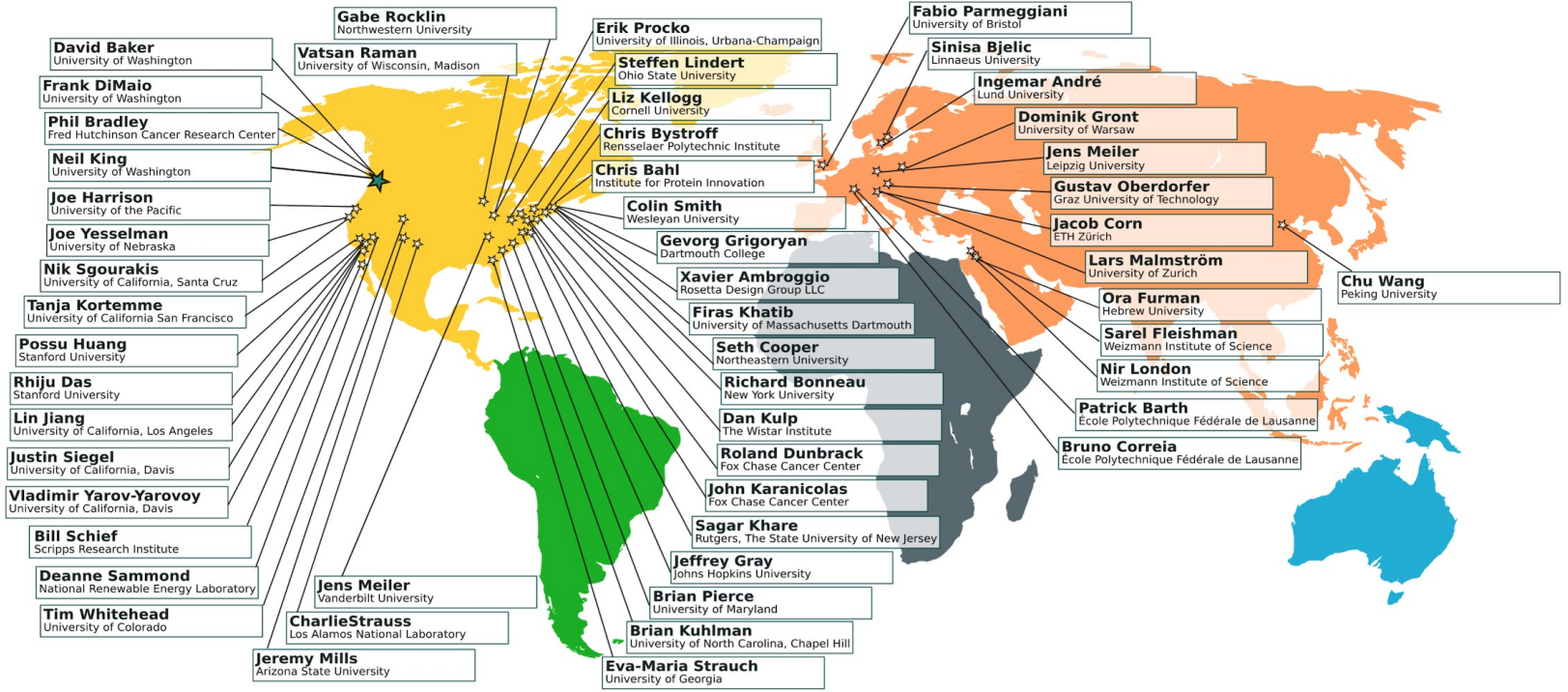
A Team Approach

Close collaboration between the labs the norm, even within single code modules. This allows for rapid enhancements and promotes the values of team science.

Rosetta Software:

RosettaCommons:

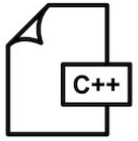
Rosetta News



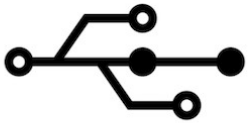
Rosetta software facts	
Development started	1997
Language	C++
Founder	David Baker
RosettaCommons institutions (active)	71
RosettaCommons labs worldwide (PIs)	52
Developers	> 500 (~100 active)
Lines of code	3.1 million

Commercial licenses	80
Academic licenses	35,700 (Rosetta)
	9,700 (PyRosetta)
Publications (estimate)	>1,500
Interfaces	PyRosetta (Python wrapper)
	RosettaScripts (XML scripts)
	Foldit (video game)
Recent success	KumaMax undergoing clinical trials for celiac disease

Choose a language wisely

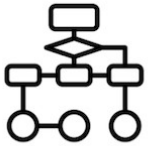


Use version control



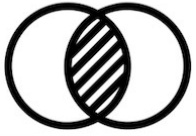
Use coding conventions

Design your code

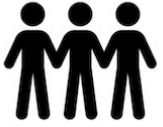


Test your code

Design developer interfaces



Establish a community



Organize meetings

Foster communication

Train developers and users

Maintain your code

Organize hackathons

Allow healthy debate

Foster diversity / outreach

Establish intern program

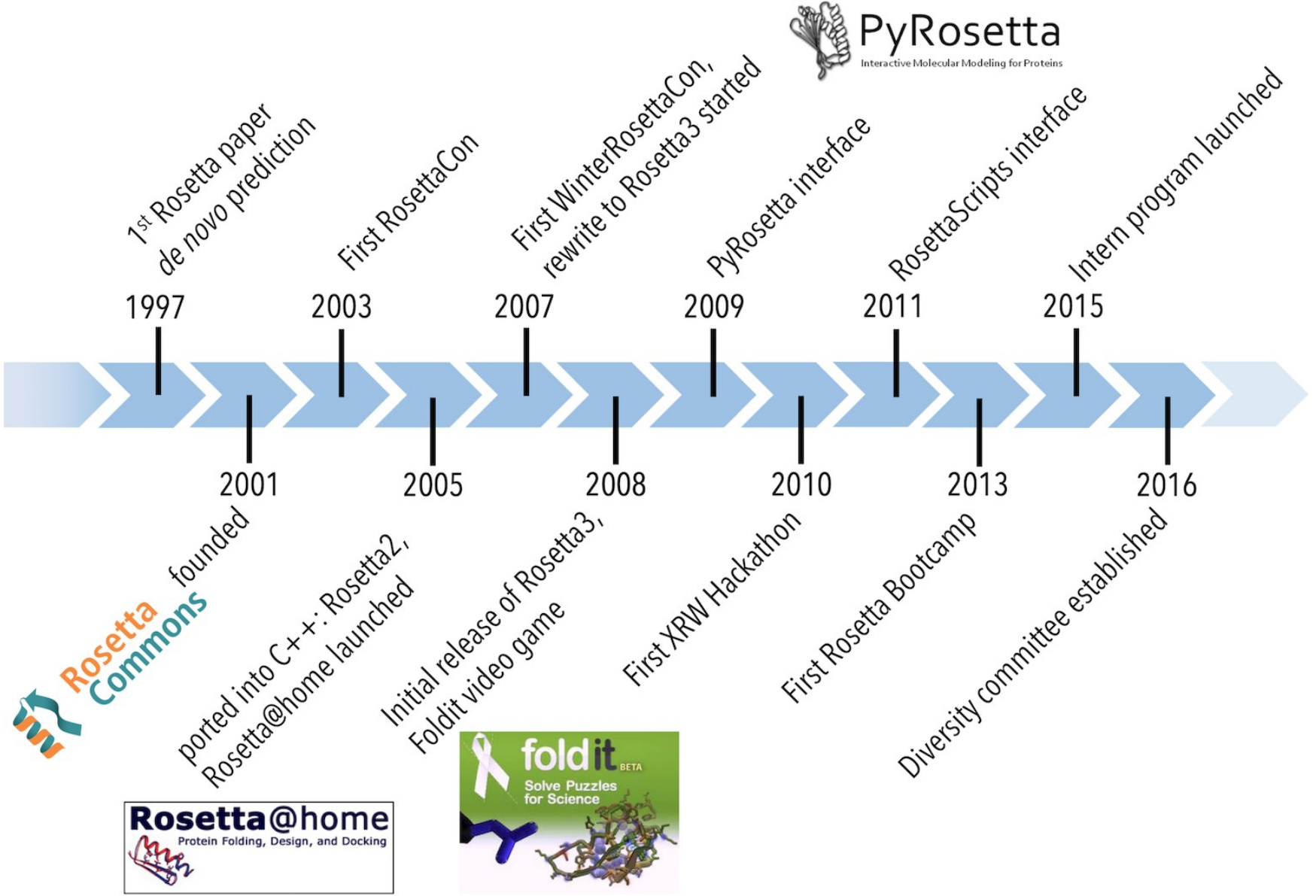
Do great science

Think about Licensing

Write documentation

Simplify user interfaces

Interact with users


PyRosetta
Interactive Molecular Modeling for Proteins
1st Rosetta paper de novo prediction
First RosettaCon
First WinterRosettaCon, rewrite to Rosetta3 started
PyRosetta interface
RosettaScripts interface
Intern program launched
1997
2003
2007
2009
2011
2015
2001
2005
2008
2010
2013
2016
Rosetta Commons founded
ported into C++: Rosetta2, Rosetta@home launched
Initial release of Rosetta3, Foldit video game
First XRW Hackathon
First Rosetta Bootcamp
Diversity committee established
Rosetta@home
Protein Folding, Design, and Docking
fold it BETA
Solve Puzzles for Science


RosettaCon 2004


RosettaCon 2017

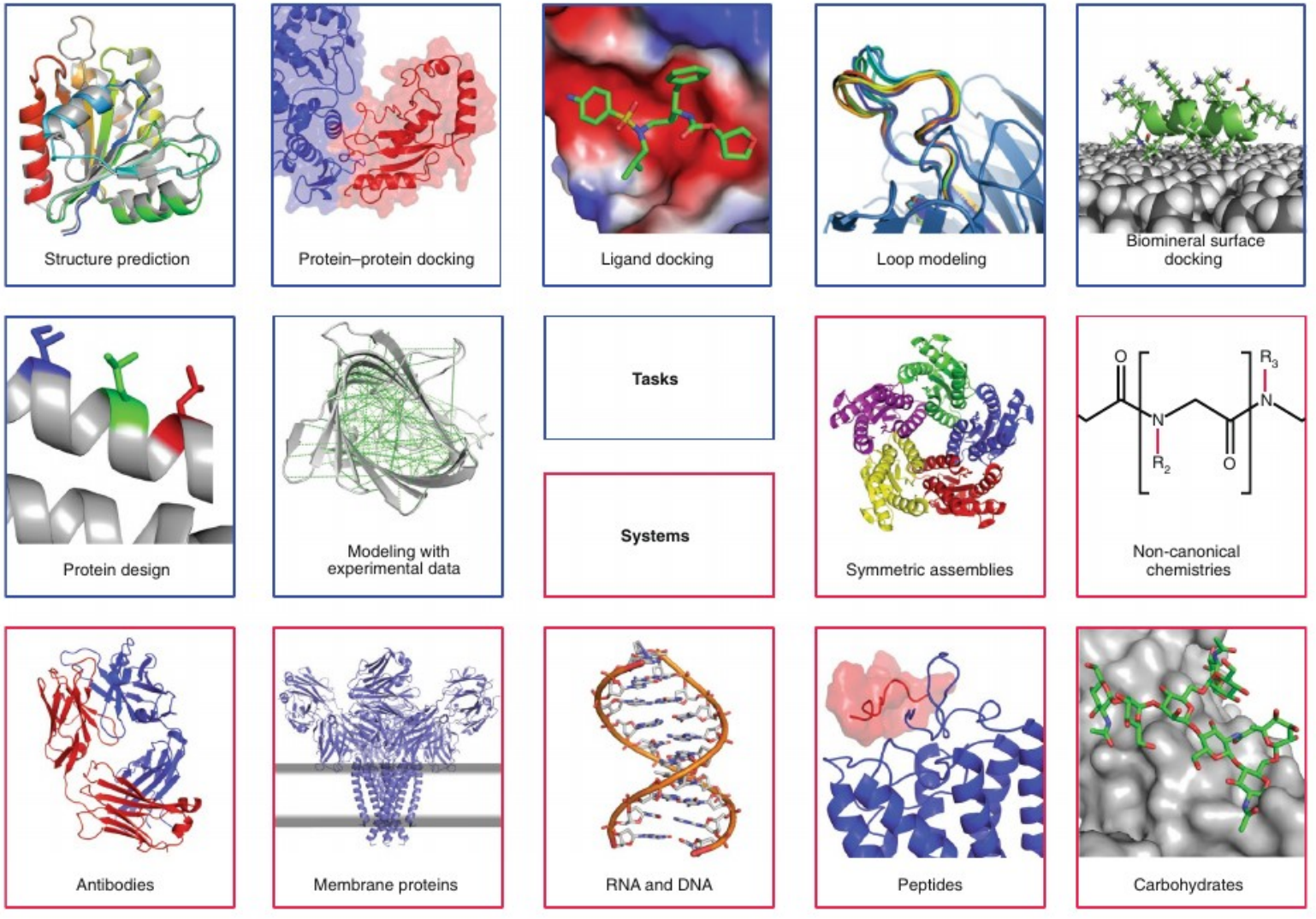


Fig. 1 | Capabilities of the Rosetta macromolecular modeling suite. Some popular tasks that can be addressed in Rosetta (blue) and major systems that can be modeled (red). Note that this is an incomplete list of Rosetta's broad modeling capabilities.

Table 1 | Overview of recent methods developed in Rosetta

Method	Developing laboratory
Score function	
REF2015 score function[28,29]	Frank DiMaio, David Baker
cartesian_ddG[29]	Frank DiMaio, Phil Bradley
HBNet[47,50]	David Baker, Brian Kuhlman
HBNetEnergy[47]	Richard Bonneau, David Baker[a]
AACompositionEnergy	Richard Bonneau, David Baker[a]
AARepeatEnergy	Richard Bonneau, David Baker[a]
VoidsPenaltyEnergy	Richard Bonneau, David Baker[a]
NetChargeEnergy	Richard Bonneau, David Baker[a]
BuriedUnsatPenalty	Richard Bonneau, David Baker[a]
Protein structure prediction	
fragment picker[190]	Dominik Gront[a,b]
RosettaCM[55]	David Baker
iterative hybridize[59,60]	David Baker, Sergey Ovchinnikov[a]
Loop modeling	
NGK (next-generation KIC)[64]	Tanja Kortemme
GenKIC (generalized KIC)[44]	Richard Bonneau, David Baker[a]
LoopHashKIC	Tanja Kortemme
Consensus_Loop_Design[101,191]	David Baker
Protein–protein docking	
RosettaDock4.0[71]	Jeffrey Gray
Rosetta SymDock2[72]	(Ingemar André)[c], Jeffrey Gray
Small molecule ligand docking	
RosettaLigand[74,192,193]	Jens Meiler
RosettaLigandEnsemble[76]	Jens Meiler
pocket optimization[77,78]	John Karanicolas
DARC[194–196]	John Karanicolas
Modeling of antibodies and immune system proteins	
RosettaAntibody[80–83]	Jeffrey Gray
AbPredict[89,90]	Sarel Fleishman
RosettaMHC[197]	Nik Sgourakis
TCRModel[198]	Brian Pierce
SnugDock[91]	Jeffrey Gray
Design of antibodies and immune system proteins	
RAbD[93] (RosettaAntibodyDesign)	Bill Schief, Roland Dunbrack
Epitope removal[95,96]	David Baker, Cyrus Biotechnology
AbDesign[97,98]	Sarel Fleishman
Protein design	
SEWING[103,104]	Brian Kuhlmann
RosettaRemodel[106]	Possu Huang[a,b]
LooDo[199]	Sagar Khare
RECON[108]	Jens Meiler
curved β-sheet design[101]	David Baker
biased forward folding[101]	David Baker
fold_from_loops[111]	Bruno Correia[a,b]
FunFolDes[112]	Bruno Correia
Protein interface design	
FlexDDG[117]	Tanja Kortemme
Coupled Moves[200]	Tanja Kortemme, DSM Biotechnology Center
Parametric design[48,120]	Richard Bonneau[a]

Table 2 | Overview of additional recent methods developed in Rosetta

Method	Developing laboratory
Peptides and peptidomimetics	
FlexPepDock[123,201]	Ora Schueler-Furman
PIPER-FlexPepDock[121]	Ora Schueler-Furman
PeptiDerive[202]	Ora Schueler-Furman
simple_cycpep_predict[44,45,120]	Richard Bonneau, David Baker[a]
MFPred[203]	Sagar Khare
RosettaSurface[124,125,204]	Jeffrey Gray
Modeling with experimental data	
cryo-EM de novo[205]	Frank DiMaio, David Baker
cryo-EM: RosettaES[126]	Frank DiMaio
cryo-EM: iterative refinement[206,207]	Frank DiMaio[a,b]
cryo-EM: automated refinement[127]	Frank DiMaio
NMR: CS-Rosetta[130]	Nik Sgourakis
NMR: PCS-Rosetta, GPS-Rosetta[132,133]	Thomas Huber
RosettaNMR framework[148]: using RDC/PRE/PCS/NOE/CS for ab initio protein–protein docking, ligand docking, symmetric assembly	Jens Meiler, Richard Bonneau, (Jeffrey Gray)[c]
mass-spec: HRF hydroxyl radical footprinting[149,150]	Steffen Lindert
mass-spec: PyTXMS[151]	Lars Malmström

RNA modeling	
SWA (stepwise assembly)[153,154]	Rhiju Das
SWM (stepwise Monte-Carlo)[152]	Rhiju Das
FARFAR (fragment assembly medium resolution structure prediction)[157,208,209]	Rhiju Das
ERRASER (refinement into EM density maps)[155,156]	Rhiju Das
CS-Rosetta-RNA (modeling with NMR data)[210]	Rhiju Das
RECCES (Reweighting of Energy-function Collection with Conformational Ensemble Sampling)[211]	Rhiju Das
DRRAFTER (de novo modeling of protein-RNA complexes into EM densities)[158]	Rhiju Das
Membrane proteins	
RosettaMP framework[172]: mp_ddg, mp_dock, mp_relax, mp_symdock	Jeffrey Gray, Richard Bonneau
RosettaMP toolkit[174]: mp_score, mp_transform, mp_mutate_relax, helix_from_sequence	Jeffrey Gray, Richard Bonneau
mp_lipid_acc[175]	Richard Bonneau
mp_domain_assembly[176]	Richard Bonneau
RosettaCM for membrane proteins[33]	Jens Meiler
Carbohydrates	
RosettaCarbohydrate framework[128,129]	Jeffrey Gray, William Schief
User interfaces	
PyRosetta[30,182,212]	Jeffrey Gray
RosettaScripts[31,33]	Sarel Fleishman[a,b]
InteractiveRosetta[183]	Chris Bystroff
Foldit Standalone[32,184,185,213]	Seth Cooper[a,b], Firas Khatib[a,b], Justin Siegel, Scott Horowitz, David Baker
ROSIE server[186,187]	Jeffrey Gray

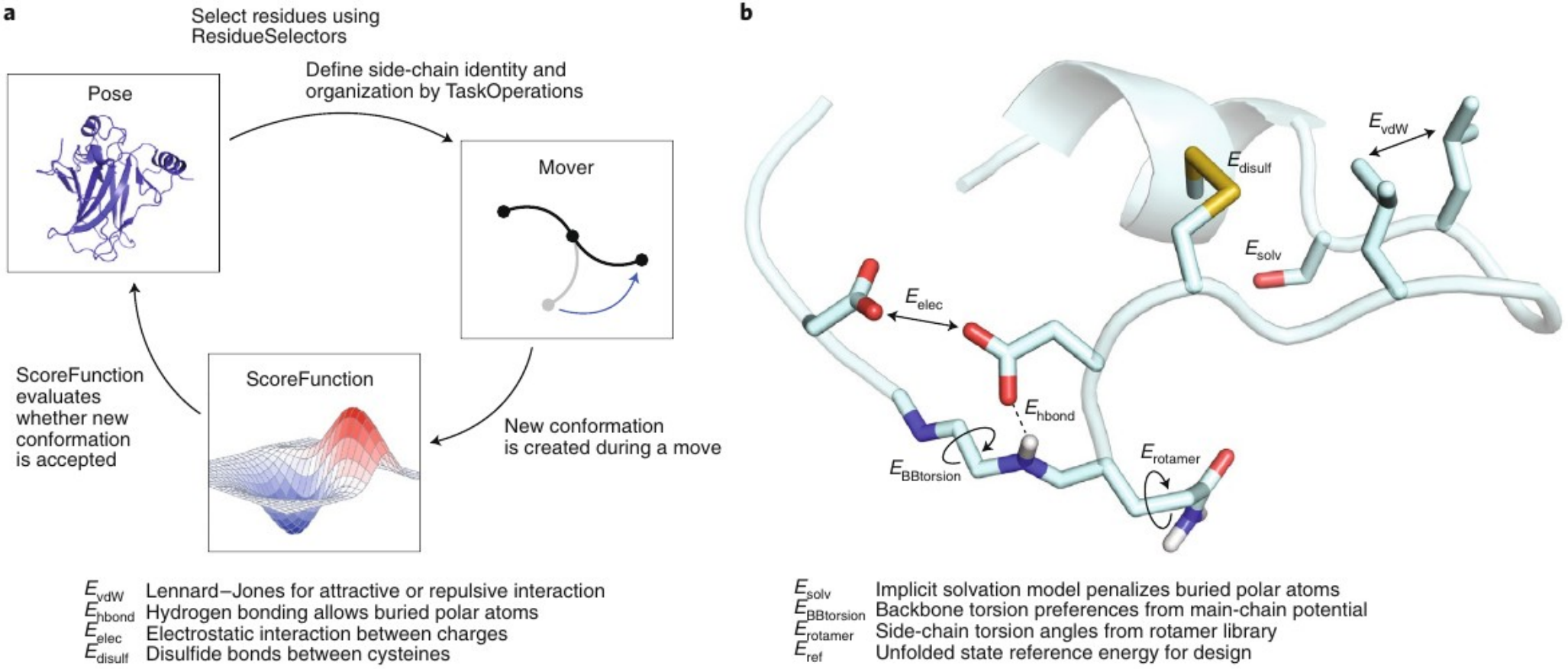


Fig. 2 | Main elements of Rosetta are scoring and sampling. a, Three main elements are required in a Rosetta protocol. The Pose is the biomolecule, such

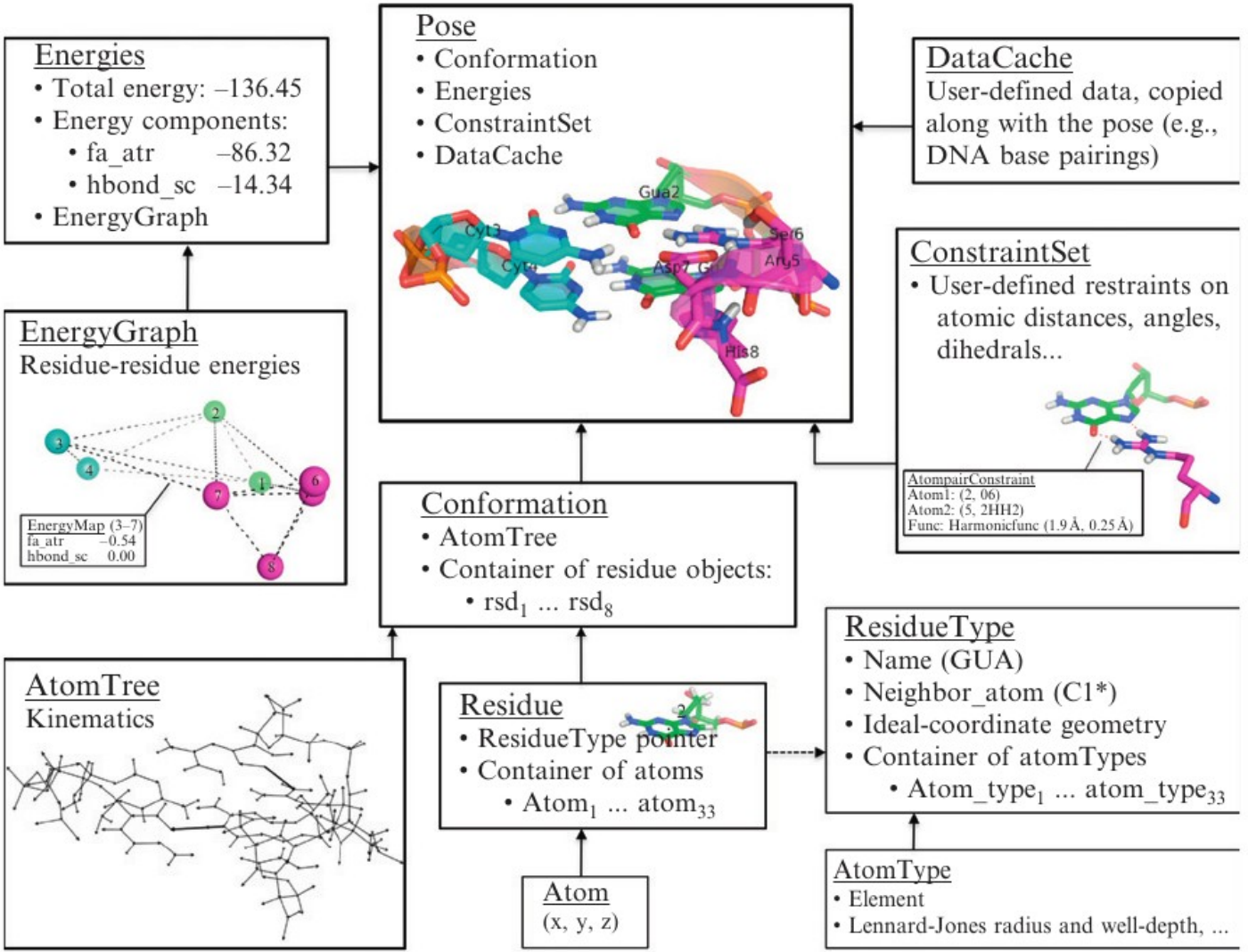
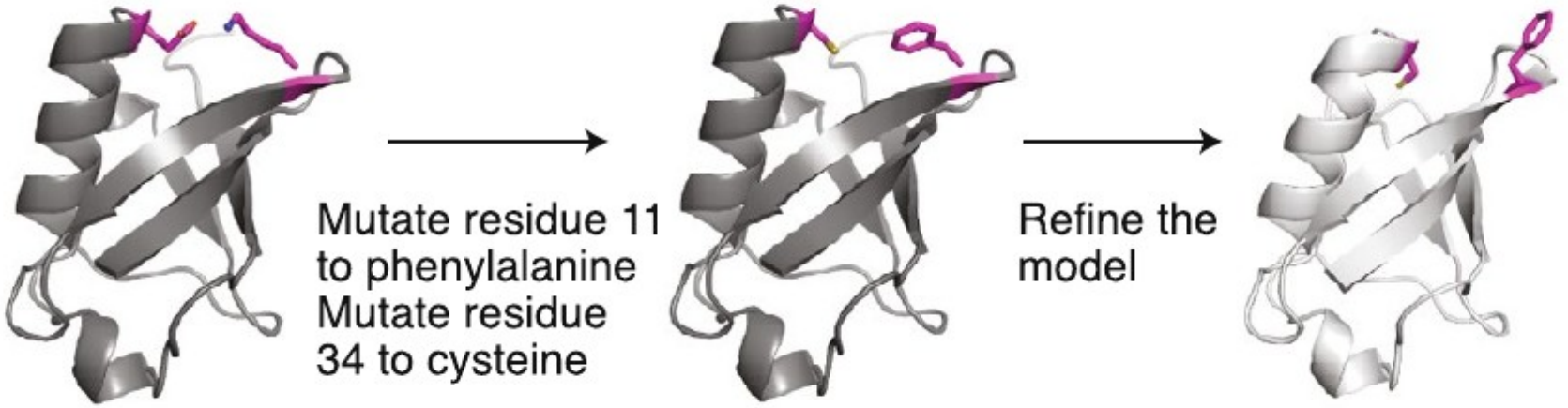


Figure 19.2 `Pose` architecture. The components of the `Pose` class are illustrated for

a



Command line

```
~/Rosetta/main/source/bin/mutate.macosclangrelease \
-database ~/Rosetta/main/database \
-in:file:s input.pdb \
-mutate:mutation K11F E34C \

~/Rosetta/main/source/bin/relax.macosclangrelease \
-database ~/Rosetta/main/database \
-in:file:s input_K11F_E34C.pdb \
-nstruct 100 \
-relax::fast \
```

RosettaScripts

```
<ROSETTASCRIPTS>
    <SCOREFXNS>
        <ScoreFunction name="ref15" weights="ref2015.wts" />
    </SCOREFXNS>
    <MOVERS>
        <MutateResidue name="mut_11" target="11" new_res="PHE" />
        <MutateResidue name="mut_34" target="34" new_res="CYS" />
        <FastRelax name="relax" scorefxn="ref15" repeats="1" />
    </MOVERS>
    <PROTOCOLS>
        <Add mover="mut_11" />
        <Add mover="mut_34" />
        <Add mover="relax" />
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

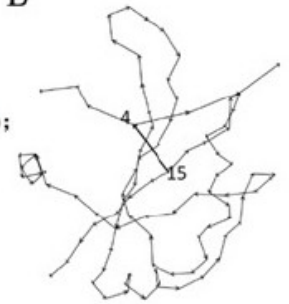
PyRosetta

```
From pyrosetta import *

init()
pose = pose_from_file( "input.pdb" )
sfxn = get_fa_scorefxn()

mutate1 = rosetta.protocols.simple_moves.MutateResidue( 11, "PHE" )
mutate2 = rosetta.protocols.simple_moves.MutateResidue( 34, "CYS" )
relax   = rosetta.protocols.relax.FastRelax( sfxn, 1 )

mutate1.apply( pose )
mutate2.apply( pose )
relax.apply( pose )
```

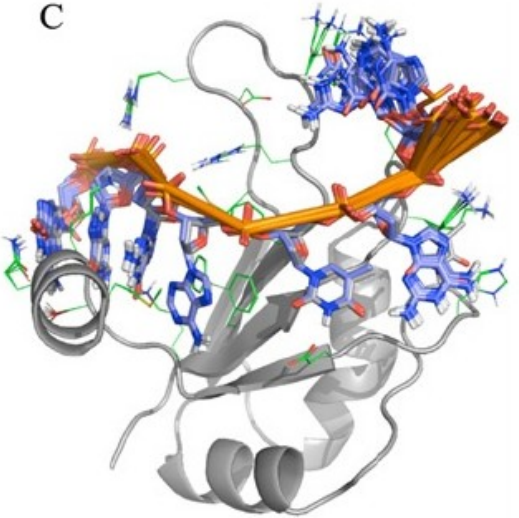
A

```
1  Pose pose;
   pose_from_pdb( "start.pdb" );

   ScoreFunction score_function;
   score_function.initialize_from_file( "scorefxn_weights.txt" );

2  FoldTree fold_tree( pose.total_residue() );
   fold_tree.new_jump( 4, 15, pose.chain_end(1) );
   fold_tree.reorder( 15 );
   pose.fold_tree( fold_tree );

   Pose const start_pose( pose );

   for ( int n=1; n<= 20; ++n ) {
       pose = start_pose;

3      PackerTaskOP packer_task( TaskFactory::create_packer_task( pose ) );
       packer_task->initialize_from_command_line();
       for ( int i=1; i<= pose.total_residue(); ++i ) {
           if ( pose.residue(i).is_protein() ) {
               packer_task->nonconst_residue_task(i).restrict_to_repacking();
           }
       }
       pack_rotamers( pose, score_function, packer_task );

4      MoveMap move_map;
       move_map.set_chi( true ); // all chi angles flexible
       move_map.set_jump( 1, true ); // protein-DNA linkage flexible
       for ( Size i=1; i<= pose.chain_end(1); ++i ) move_map.set_bb( i, true );
       AtomTreeMinimizer minimizer;
       MinimizerOptions minimizer_options( "dfpmin_armijo", 0.001, true );
       minimizer.run( pose, move_map, score_function, minimizer_options );

5      std::cout << "final_energy: " << score_function( pose ) <<
           " DNA_sequence: " << pose.chain_sequence(1) << std::endl;
       pose.dump_pdb("dna_redesign_and_minimize_"+string_of( n )+".pdb" );
   }
```

B



C



D

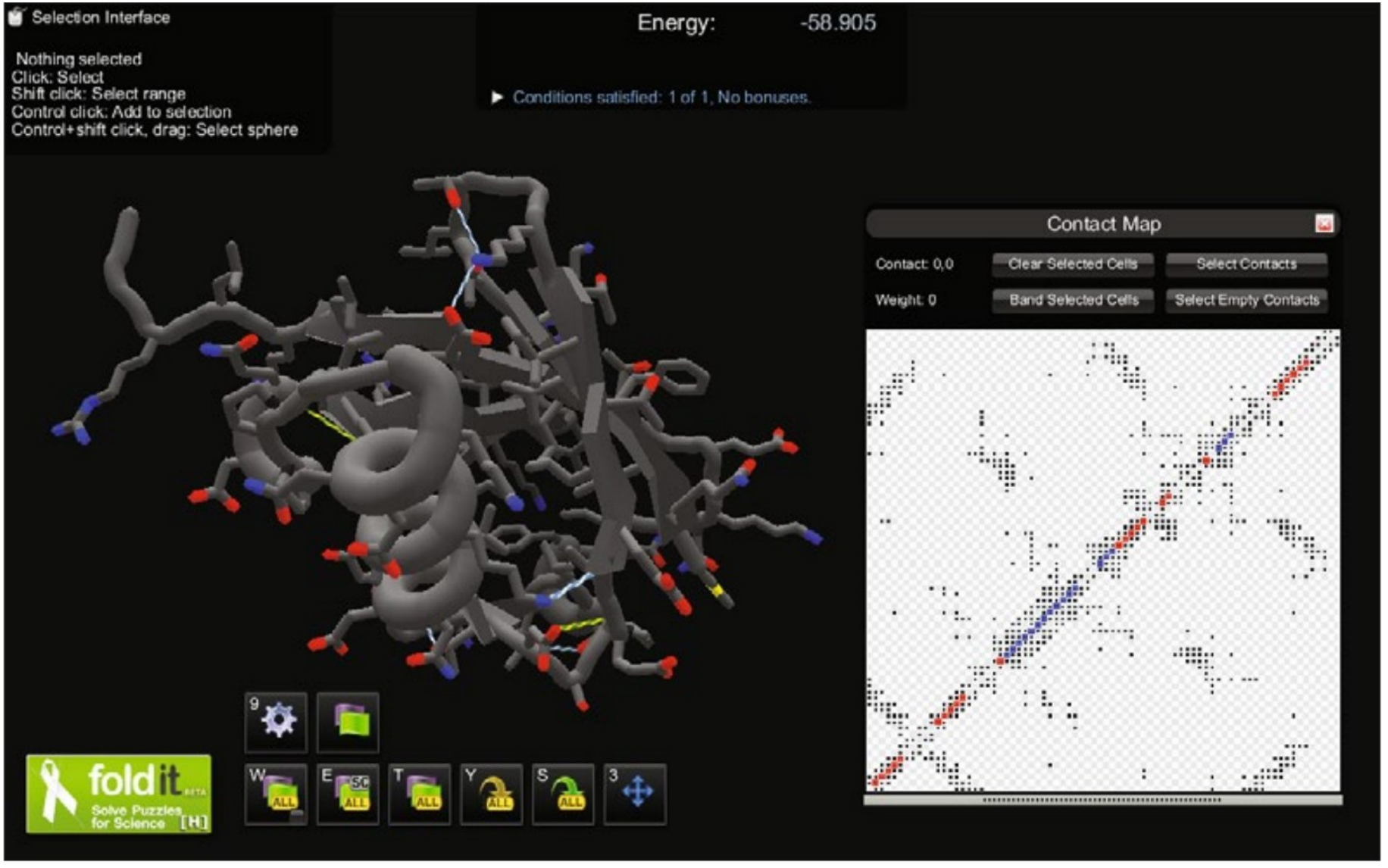
```
final_energy:  -239.123 DNA_sequence: agtaggg
final_energy:  -241.046 DNA_sequence: aataggg
final_energy:  -241.345 DNA_sequence: cataggg
final_energy:  -241.213 DNA_sequence: agtaggg
final_energy:  -236.988 DNA_sequence: cataggg
final_energy:  -242.525 DNA_sequence: aataggg
final_energy:  -238.083 DNA_sequence: cataggg
final_energy:  -236.964 DNA_sequence: cgtaggg
final_energy:  -239.914 DNA_sequence: cataggg
final_energy:  -239.130 DNA_sequence: ggtaggg
final_energy:  -238.840 DNA_sequence: ggtaggg
final_energy:  -234.287 DNA_sequence: cgtaggg
final_energy:  -240.871 DNA_sequence: ggtaggg
final_energy:  -235.621 DNA_sequence: cataggg
final_energy:  -235.892 DNA_sequence: cataggg
final_energy:  -239.190 DNA_sequence: aataggg
final_energy:  -240.089 DNA_sequence: cgtaggg
final_energy:  -239.897 DNA_sequence: aataggg
final_energy:  -237.005 DNA_sequence: cgtaggg
final_energy:  -239.993 DNA_sequence: cataggg
```

E

Figure 19.4 Simple ROSETTA3 protocol for performing a binding specificity calculation

InteractiveRosetta

Foldit


Selection Interface
Nothing selected
Click: Select
Shift click: Select range
Control click: Add to selection
Control+shift click, drag: Select sphere
Energy: -58.905
Conditions satisfied: 1 of 1, No bonuses.
Contact Map
Contact: 0,0
Clear Selected Cells
Select Contacts
Weight: 0
Band Selected Cells
Select Empty Contacts
foldit
Solve Puzzles for Science [H]

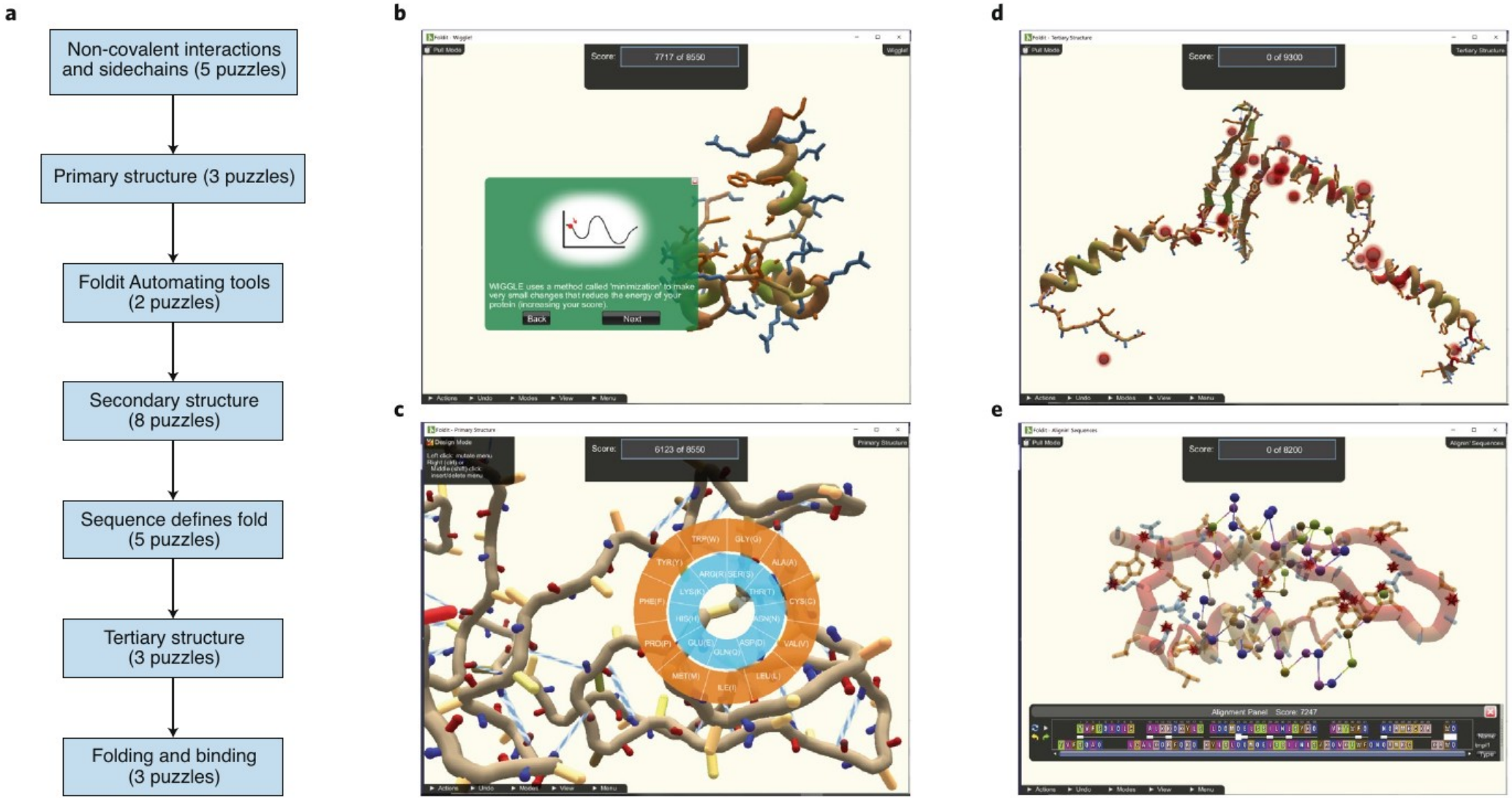


Fig. 1 | Foldit Education Mode. a, A flowchart depicting the current topics and number of puzzles within Foldit Education Mode. **b**, Wiggle puzzle, showing a

ROSIE

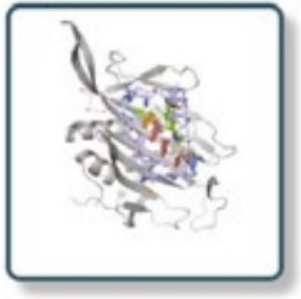
Welcome to ROSIE

Rosetta Online Server that Includes Everyone

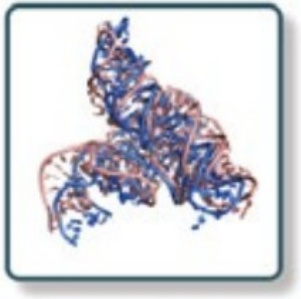
Welcome | Queue | About | ChangeLog | Documentation | Support | Login | Create an account

Recommend 9 Share G+

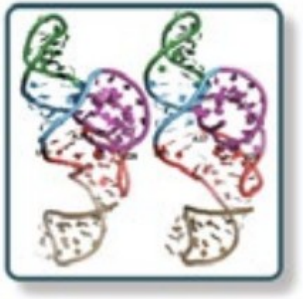
Rosetta Protocols opened for academic users:



[Rnp_ddg]



[Farfar2]



[Stepwise]

ROSIE stats (24hrs):

Users:	6,472	+5
Jobs:	66,216	+49
CPU hours:	6,501,462	+5,895

See more info at our About page.

Get Started with ROSIE

- ROSIE Documentation – Server related documentation and info.
- Rosetta Forums This is a list of forums for Rosetta users to discuss problems with running Rosetta and is monitored by Rosetta developers.

Getting started

https://new.rosettacommons.org/docs/latest/getting_started/Getting-Started

Getting Started

- Solving a Biological Problem
- I want to sample X
- Classifying problems
- Analyzing Results
- Learning biophysics
- Incorporating experimental data
- Rosetta canon

Build Documentation

Rosetta Tutorials

Rosetta Basics

Rosetta Applications

Rosetta Scripting Interfaces

Development Documentation

FAQ

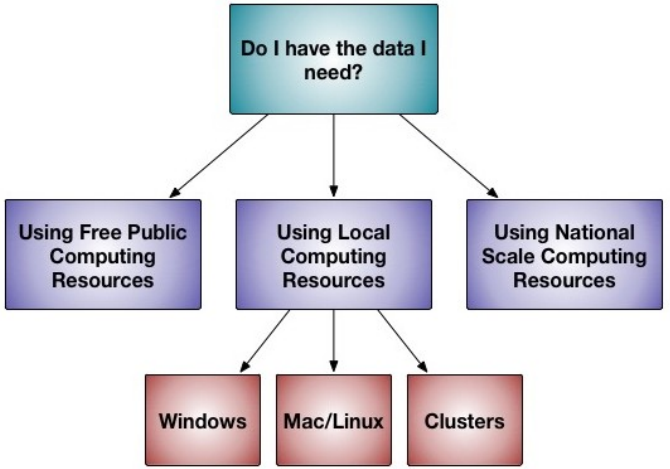
Glossary

RosettaEncyclopedia

Options list

This page is written for an audience of scientists new to Rosetta: perhaps a first year graduate student, or young postdoc, who has received/started a project that needs "some computer modeling". In other words, an individual coming to Rosetta from a cold start. Is Rosetta a good tool for the modeling you need to do? If so, how do you go about getting and using Rosetta? If you are already comfortable with the concepts, feel free to skip ahead.

Rosetta is a very large software suite for macromolecular modeling. By software suite, we mean that it is a large collection of computer code (mostly in C++, some in Python, a little in other languages), but it is not a single monolithic program. By macromolecular modeling, we mean the process of evaluating and ranking the physical plausibility of different structures of biological macromolecules (usually protein, but nucleic acids and ligands are significantly supported and support for implicit lipid membranes is growing). Generally, a user will choose some specific protocol within Rosetta and provide that protocol with inputs for A) what structure to work on, and B) what options within the protocol are appropriate for the user's needs.



Solving a Biological Problem

Getting Started
- Solving a Biological Problem
- I want to sample X
- Classifying problems
- Analyzing Results
- Learning biophysics
- Incorporating experimental data
- Rosetta canon

Build Documentation
Rosetta Tutorials
Rosetta Basics
Rosetta Applications
Rosetta Scripting Interfaces
Development Documentation
FAQ
Glossary
RosettaEncyclopedia
Options list

There are many biological problems which can be approached with Rosetta. These topics often stem from the central idea that **sequence** confers **structure** which in turn confers **function**. For example, Rosetta was initially utilized for *de novo* protein structure prediction (sequence->structure), while current applications can be as extensive as enzyme design (function->sequence).

Generally speaking (for most biological problems), the computational challenges faced are two-fold. First, can we adequately sample the space where the solution to our problem lives? Second, can we identify said solution, if we have sampled it? This should be kept in mind when deciding on which protocol to apply to your problem, how many models to generate, and which score function to use.

If this page doesn't have a problem that sounds like yours, it may be that you haven't determined what the problem is, or that we can't solve it. If you are thinking of your problem from an angle of sampling a specific degree of freedom, see this page.

Table of Contents

- Types of Biological Problems
  - Protein Structure Prediction
    - De Novo Modeling
    - Comparative Modeling (Homology Modeling)
    - Specialized Protocols
  - Protein–Protein Docking
    - Docking Two Partners With Known Structures
      - Docking According to the Lock and Key Model
      - Docking According to the Conformer Selection Model
      - Docking According to the Induced Fit Model
      - Docking According to the Conformer Selection and Induced Fit Model
    - Docking Two Partners Where One Structure Is Unknown
    - Docking Two Partners With Two Unknown Structures
    - Docking Homooligomers

https://new.rosettacommons.org/docs/latest/getting_started/Solving-a-Biological-Problem

Tutorials

These are introductory tutorials intended as a gentle introduction to Rosetta concepts, and using common functionality of Rosetta.
For additional examples and information on using Rosetta, see the demos (below) or the Rosetta documentation

Full input files for the tutorials are located in the `demos/tutorials/` directory of the Rosetta distribution.

Introduction to Rosetta

1. How To Read These Tutorials
2. Installing and Building Rosetta
3. Working With Rosetta
4. Controlling Input and Output
5. Core Rosetta Concepts
6. Working with Non-protein Residues
7. Scoring: Calculating the Energy of a Structure
8. Full-Atom vs. Centroid Representations
9. The Packer: Optimizing Sidechains
10. Minimization: Finding Deeper Energy Wells
11. Relax: Refining Structures
12. Constraints: Biasing Towards a Structure
13. Analyzing Rosetta Output
14. The Fold Tree: Propagating Changes in the Structure
15. Symmetry: Modeling Symmetric Proteins
16. Scripting with RosettaScripts
17. Advanced Scripting with RosettaScripts
18. Commonly Used Options
19. Tips

https://www.rosettacommons.org/demos/latest/Home#tutorials

Commonly Used Rosetta Protocols

- de novo (ab initio) Structure Prediction
    - Advanced de novo Structure Prediction
- Comparative Modeling: Modeling based on Homologs
- Generalized Kinematic Closure (GenKIC): Rapid, versatile loop closure without fragments
    - GenKIC Tutorial 1: Building and closing new loops
    - GenKIC Tutorial 2: Perturbing existing loops
    - GenKIC Tutorial 3: Using pre-selection movers within GenKIC
    - GenKIC Tutorial 4: Closing through disulfides
- Loop Modeling and Rebuilding: Modeling Short Fragments
- Protein Design
- Protein-Protein Docking: Modeling Protein-Protein Binding
    - Advanced Protein-Protein Docking
- Protein-Ligand Docking: Modeling Protein-Ligand Binding

Table of Contents

- Structure Prediction
- Design
- Analysis
- Utilities
- Scripting Interfaces
- Loops
- Docking
- Interfaces
- Surfaces
- Experimental Data
- Symmetry
- Membranes
- Antibodies
- Enzymes
- Ligands
- Peptides
- Noncanonicals
- Metals
- Nucleic Acids
- Dna
- Rna
- Denovo
- Stability Improvement
- Comparative Modeling
- Core Concepts
- General

https://www.rosettacommons.org/demos/latest/tag-search

Design

- anchored_design: Anchored Design
- anchored_design: Anchored Design
- backrub_seqtol: Backrub Sequence Tolerance
- beta_peptide_modeling: Beta-3-peptide design
- beta_strand_homodimer_design: β-Strand Homodimer Design
- broker: Broker Protocol Captures
- broker_domain_insertion: Toplogy Broker - Domain Insertion Protocol Capture
- Constraints: Constraints Tutorial
- design_raf_rac_interface: Design the Rac/Raf interface
- design_with_flex_loops: Design with flexible loops
- design_with_ncaa: Design with non-canonical amino acids (NCAA)
- domain_insertion: Domain Insertion Demo
- doug_dock_design_min: Doug's DockDesignMinimize Demo
- enzyme_design: Enzyme Design Demo
- favor_native_residue: Favor Native Residue
- fixbb_design: Demo for the fixbb application with design
- ideal_proteins: Design of Proteins with Ideal Topologies
- loodo: LooDo (Loop-Directed Domain Insertion)
- multi_state_desing_of_antibodies: Multistate Design of Antibodies
- multistate_apl: Heterodimeric Antibody Design using Multistate Design
- next_generation_kic: Next Generation KIC
- oop_design: Oop Design: Design with oligooxopiperazine helix mimetics
- Optimizing_Sidechains_The_Packer: Optimizing Side-Chains: Introduction to the *Packer*
- peptide_designs: Peptide Backbone and Sequence Design
- peptide_designs: Peptide Backbone and Sequence Design
- protein_design_tutorial: Protein Design
- rna_design: Fixed backbone design of RNA
- rosetta_scripts_enzdes: Rosetta scripts: enzdes
- rosetta_scripts_flxbb_design: Flexible backbone design with RosettaScripts

Documentation & Support

Overview of Rosetta

Forum

User Guide

Demos

Education

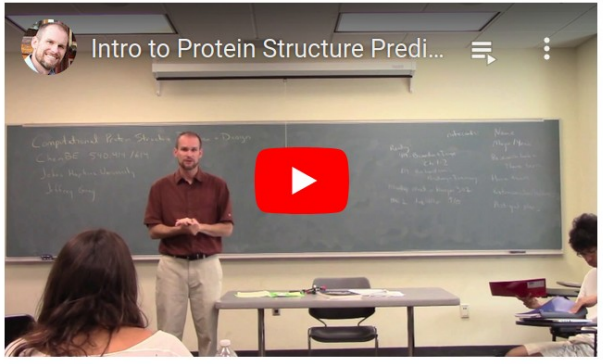
This page offers education resources for Rosetta users.

Meiler Lab teaching resources

The Meiler Lab at Vanderbilt University has an extensive library of written material, video lectures and tutorials about Rosetta, including materials from the Rosetta Workshop. Go

Computational Protein Structure Prediction and Design (JHU)

Videos from the advanced undergraduate and graduate course (540.414 and 540.614) taught at the Whiting School of Engineering of Johns Hopkins University in the Fall semester of 2014. Go



Current Tutorials

Rosetta

Rosetta Workshop 2022

Rosetta Antibody Workshop 2021

Rosetta Protein Design Workshop 2018

BCL

BCL Tutorials 2022

Archive

Rosetta

Rosetta Workshop 2020

Rosetta Antibody Workshop 2019

Rosetta Workshop 2018

Rosetta Workshop 2017

Rosetta Antibody Workshop 2017

Rosetta Workshop 2016

Materials for "Protocols for Molecular Modeling with Rosetta3 and RosettaScripts." (2016)

17th Annual Great Lakes GPCR Conference (2016)

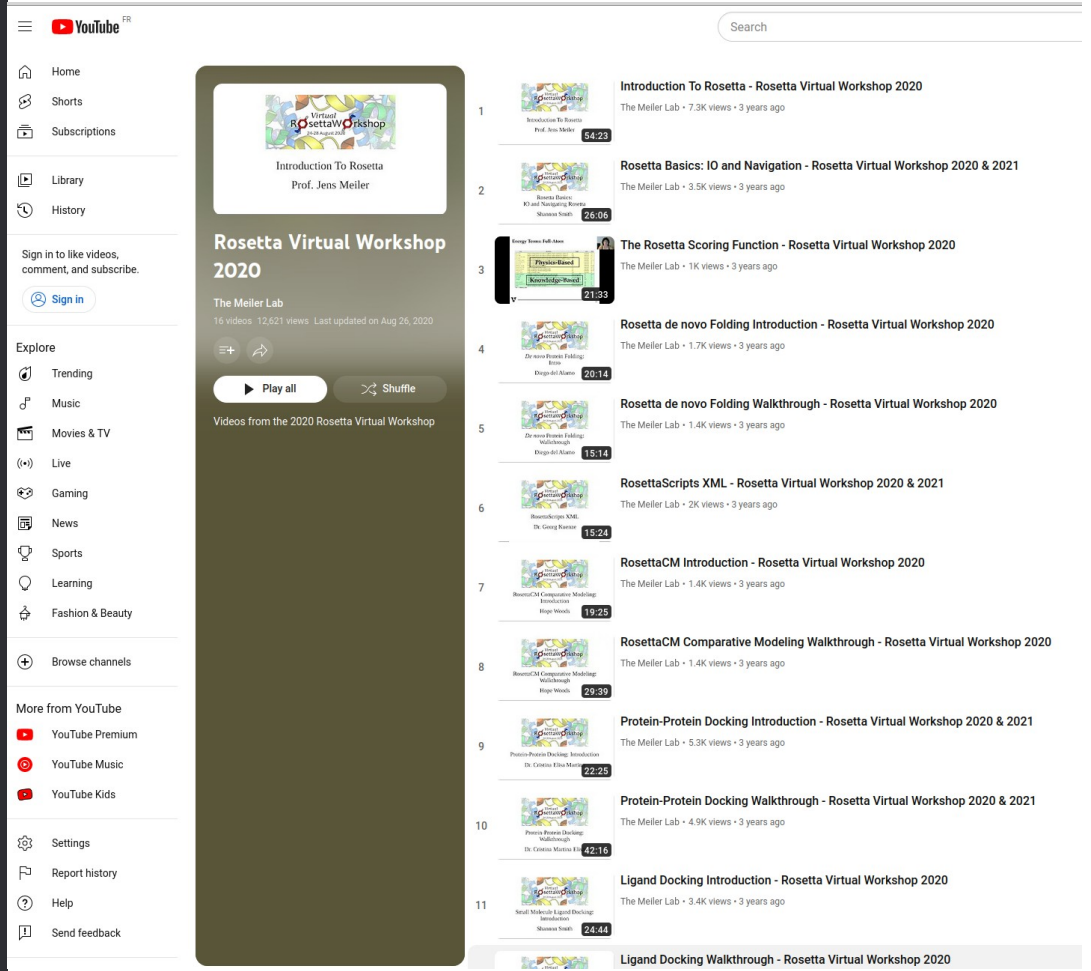
Rosetta Workshop 2015

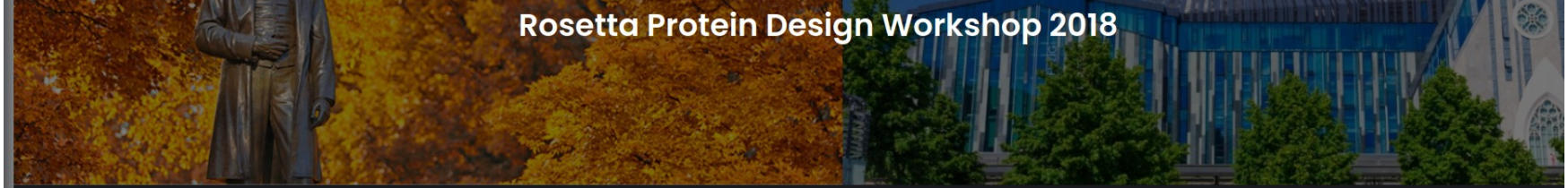
Rosetta Workshop 2014

Rosetta Membrane Workshop 2014

Rosetta Workshop 2011

Materials for "Practically Useful: What the Rosetta Protein Modeling Suite Can Do for You" (2010)





A workshop focusing on protein design was held in May of 2018 based on Rosetta 3.9, this workshop included hands-on tutorials for structure prediction of modified proteins, thermostabilization, single and multistate design, small molecule interface design, peptide design, scaffold and motif grafting as well as presentations on Rosetta basics.

Main Topics:

Tutorial 1: RosettaCM Prediction of Designed Structures

- Presentation (pdf)
- Tutorial (pdf)
- Tutorial Materials (zip)

Tutorial 2: Thermostabilization and General Protein Design

- Presentation (pdf)
- Tutorial (pdf)
- Tutorial Materials (zip)

Tutorial 3: Antibody Single & Multistate Design

- Presentation (pdf)
- Tutorial (pdf)
- Tutorial Materials (zip)

Tutorial 4: Enzyme and protein-small molecule interface design

- Presentation (pdf)
- Tutorial (pdf)

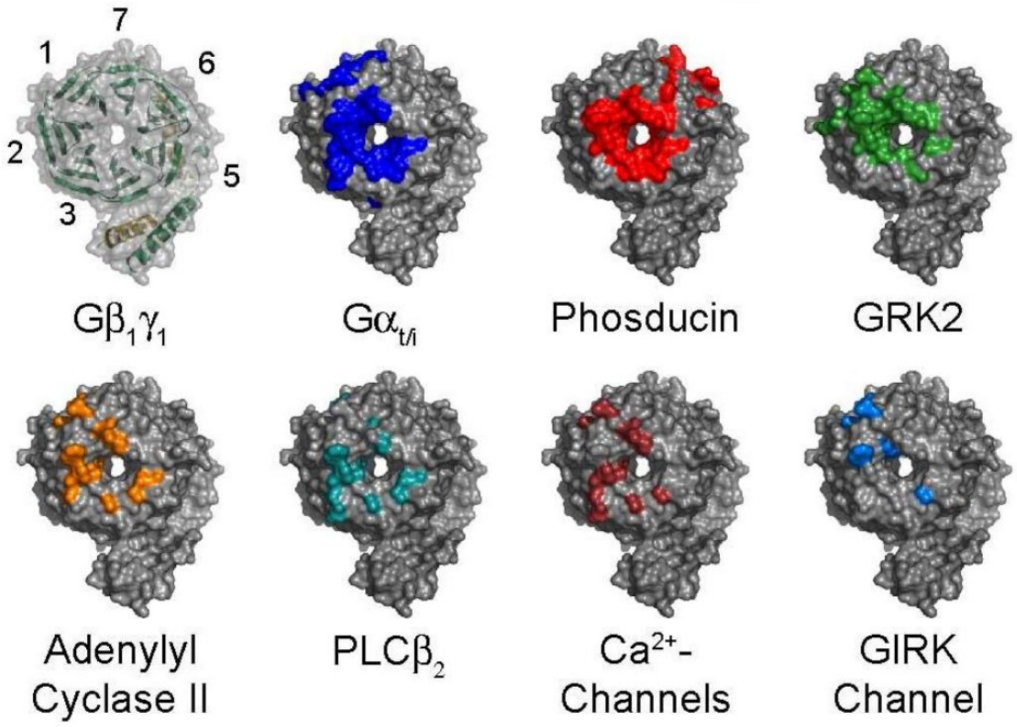
Tutorial 5: Peptide Design

- Presentation (pdf)
- Tutorial Materials (zip)

Tutorial 6: Protein Scaffold and Motif Design

- Presentation (pdf)
- Tutorial (pdf)
- Tutorial Materials (zip)

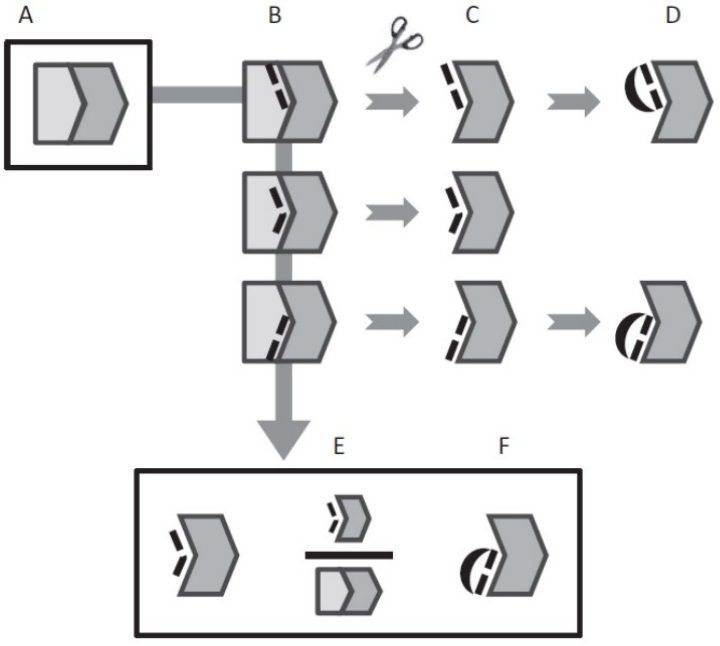
Challenges to Inhibiting Protein-Protein Interactions



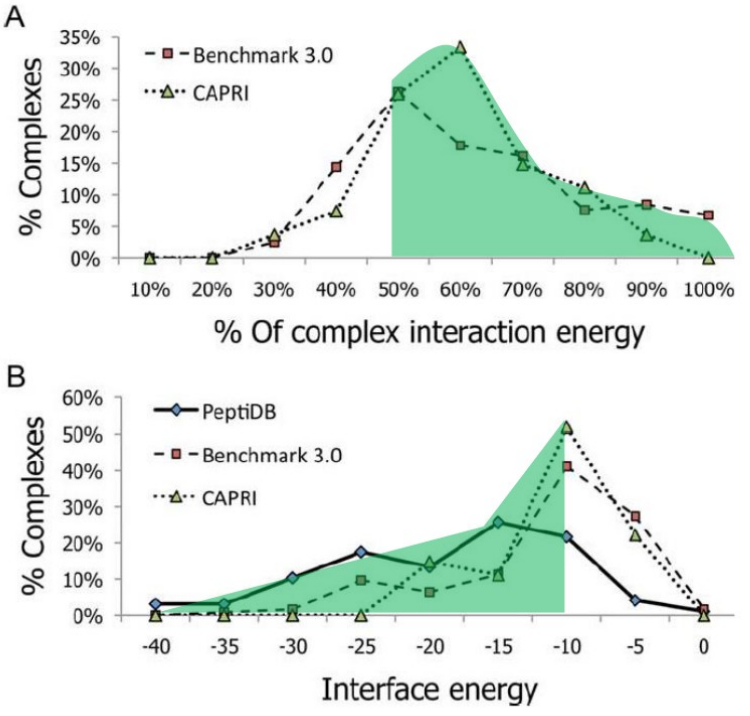
- Interactions are often flat surfaces as opposed to pockets
  - Small molecule drugs often need pockets
  - Small molecules are by definition small and may not cover entire binding surface
  - Peptides can bind along surface and with increased specificity than small molecules

PeptiDerive Identifies Peptides from Complex that Make Up the Majority of Binding Energy

PeptiDerive Algorithm



Energetic Analysis



Peptide Design Tutorial

1. Submit Protein-Protein Complex to PeptiDerive Server on Rosie
   - http://rosie.rosettacommons.org/peptiderive
   - Necessary input: PDB of protein-protein complex
   - Output: Protein-peptide complex

Identifiy peptides that can compete

2. Redock Output Peptide-Protein Complex with FlexPepDock
   - http://flexpepdock.furmanlab.cs.huji.ac.il/
   - Necessary input: PDB of protein-peptide complex (output from PeptiDerive)

Check if the peptide will bind in the same manner without larger protein