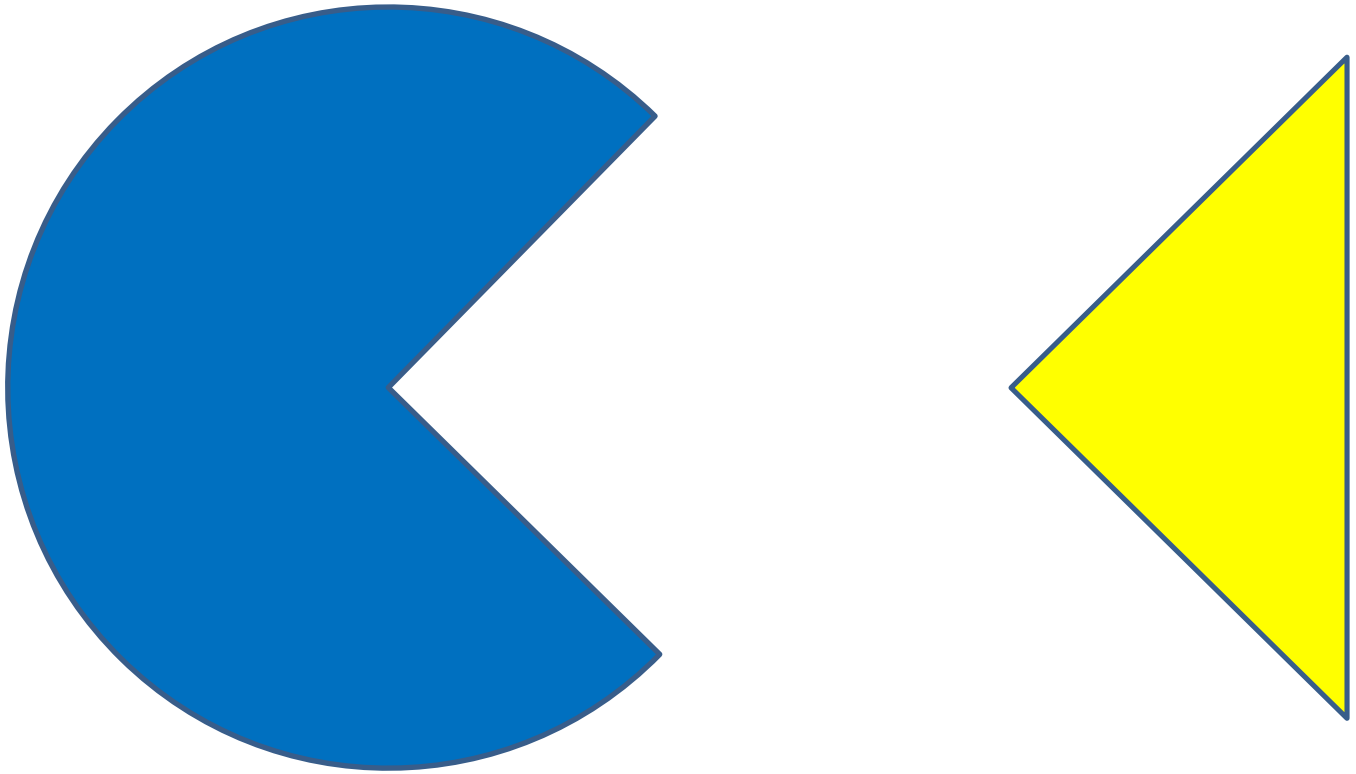
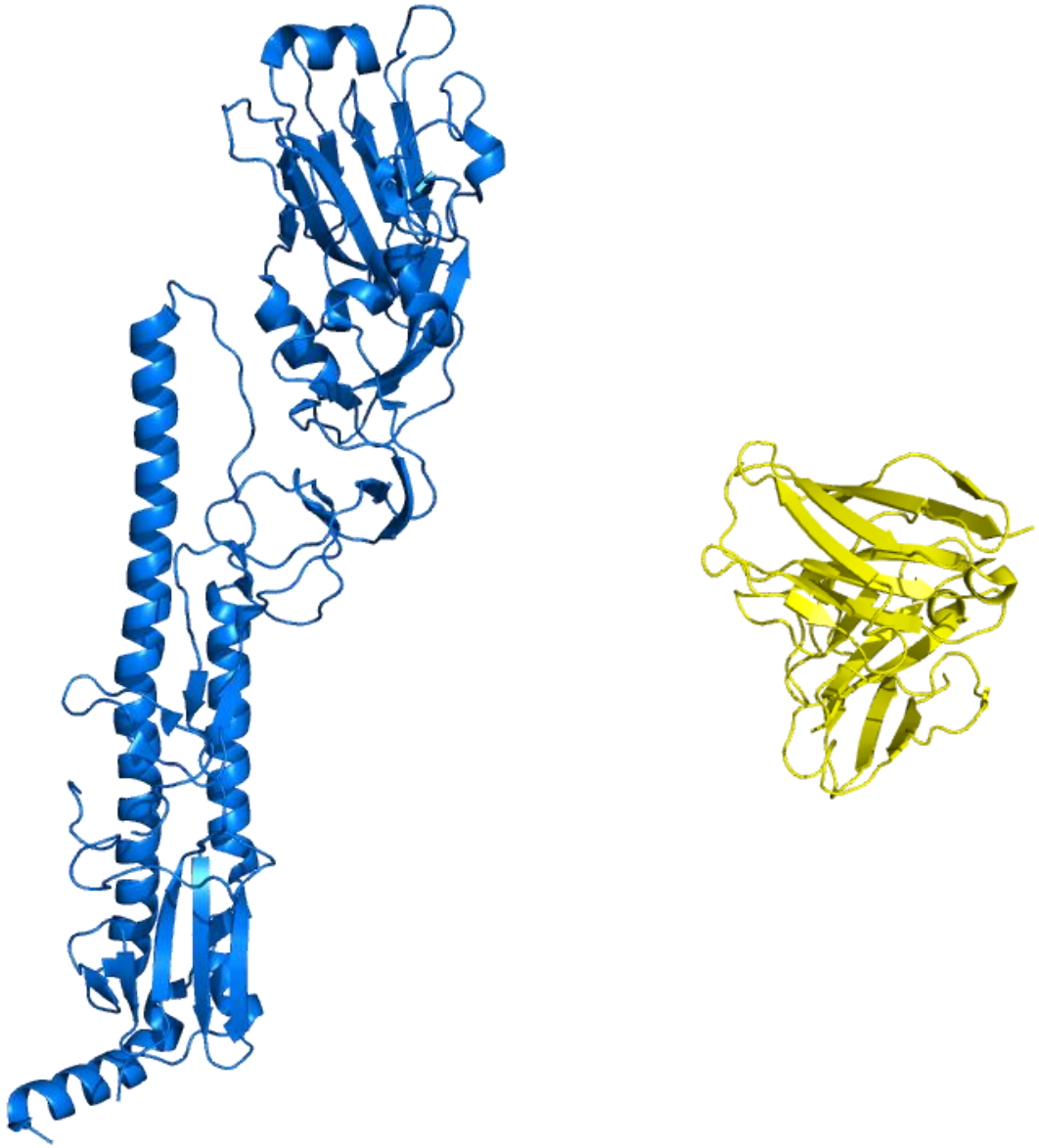
Protein-Protein Docking

Nina Bozhanova
Rosetta Workshop
December 2019, Leipzig

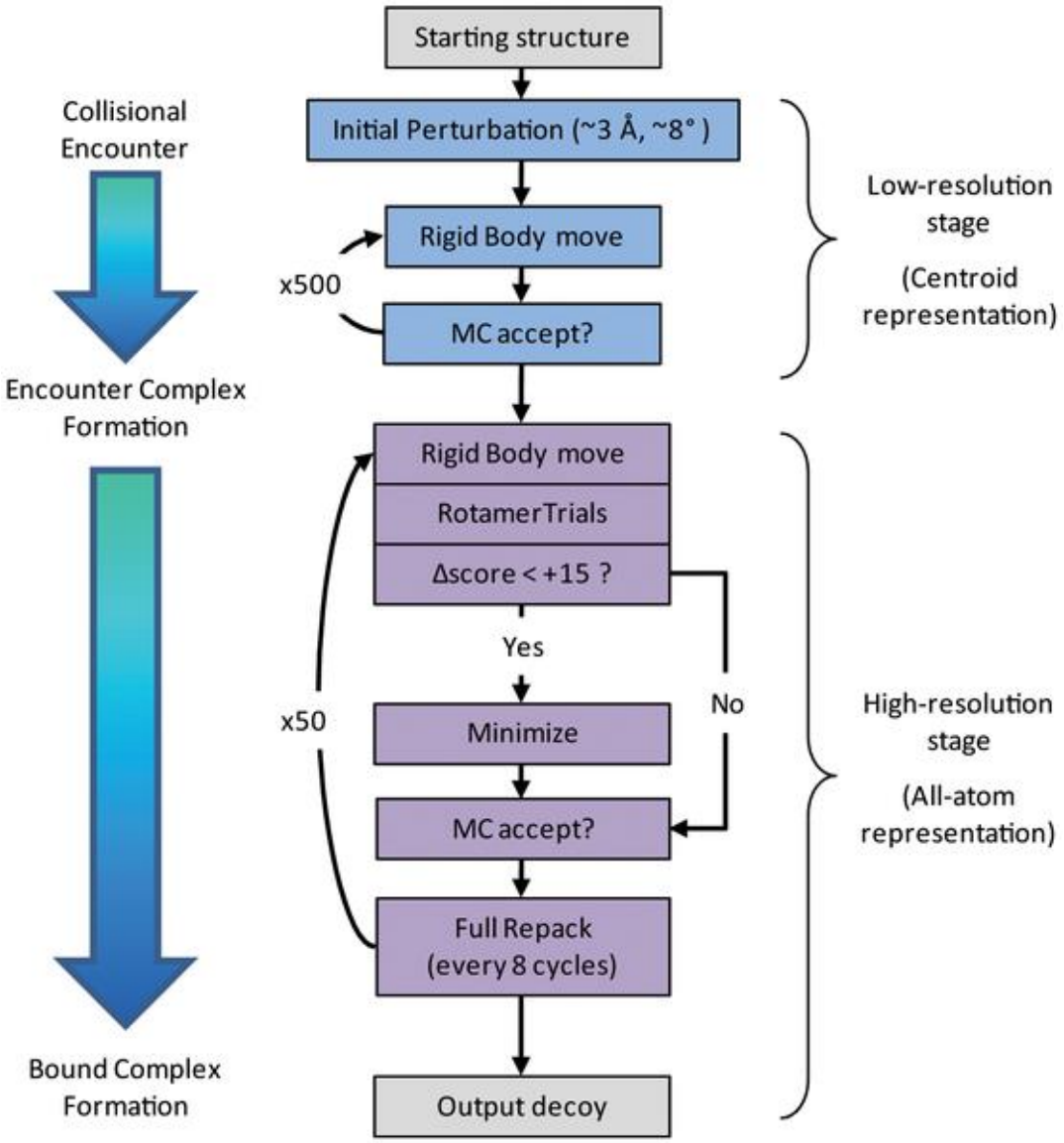
Does Protein A Dock (or Bind to) Protein B?



Does Antibody A bind Antigen B?



Overview – Rosetta Protein Docking



Mimics natural process of protein interaction

Hint: Incorporate experimental data to decrease conformational search space!!

Chaudhury *et al*, PlosONE, 2011

General Docking Protocol

1. **Prepare input files**
   - Input structures
   - RosettaScripts XML File
   - Options File
2. **Perform docking**
3. **Analyze results**
   - Plot Score vs. RMSD
   - Examine interface

Step 1 – Prepare Input Files

Preparing your docking XML file

Open the docking_full.xml file:

```
gedit ~/rosetta_workshop/tutorials/proteinprotein_docking/
input_files/docking_full.xml
```

docking_full.xml

```
<ROSETTASCRIPTS>
            <SCOREFXNS>
            </SCOREFXNS>
            <TASKOPERATIONS>
                        <InitializeFromCommandline name="ifcl"/>
                        <RestrictToRepacking name="rtr" />
                        Restrict to residues within a distance and vector cutoff of the protein-protein int
                        <RestrictToInterfaceVector name="rtiv" chain1_num="1,2" chain2_num="3,4" CB_dist_cu

                        Fix residues known experimentally to be critical in interaction
                        <PreventResiduesFromRepacking name="prfrp" residues="11,41,345" />
            </TASKOPERATIONS>
            <FILTERS>
            </FILTERS>
            <MOVERS>
                        MINIMIZATION MOVERS
                        Single cycle of FastRelax to minimize backbone of docking partners
                        <FastRelax name="minimize_interface" scorefxn="REF2015" repeats="1" task_operations
                        DOCKING MOVERS
                        <Docking name="dock_low" score_low="score_docking_low" score_high="REF2015" fullato
                        <Docking name="dock_high" score_low="score_docking_low" score_high="REF2015" fullat

                        <SaveAndRetrieveSidechains name="srsc" allsc="0" /> Speeds the move from centroid t
            </MOVERS>
            <APPLY_TO_POSE>
            </APPLY_TO_POSE>
            <PROTOCOLS>
                        Run docking protocol
                        <Add mover="dock_low"/>
                        <Add mover="srsc" />
                        <Add mover="dock_high" />

                        Minimize interface
                        <Add mover="minimize_interface" />
            </PROTOCOLS>
            <OUTPUT scorefxn="REF2015" />
</ROSETTASCRIPTS>
```

docking_full.xml

```
<ROSETTASCRIPTS>
            <SCOREFXNS>
            </SCOREFXNS>
            <TASKOPERATIONS>
                        <InitializeFromCommandline name="ifcl"/>
                        <RestrictToRepacking name="rtr" />
                        Restrict to residues within a distance and vector cutoff of the protein-protein int
                        <RestrictToInterfaceVector name="rtiv" chain1_num="1,2" chain2_num="3,4" CB_dist_cu

                        Fix residues known experimentally to be critical in interaction
                        <PreventResiduesFromRepacking name="prfrp" residues="11,41,345" />
            </TASKOPERATIONS>
            <FILTERS>
            </FILTERS>
            <MOVERS>
                        MINIMIZATION MOVERS
                        Single cycle of FastRelax to minimize backbone of docking partners
                        <FastRelax name="minimize_interface" scorefxn="REF2015" repeats="1" task_operations
                        DOCKING MOVERS
                        <Docking name="dock_low" score_low="score_docking_low" score_high="REF2015" fullato
                        <Docking name="dock_high" score_low="score_docking_low" score_high="REF2015" fullat

                        <SaveAndRetrieveSidechains name="srsc" allsc="0" /> Speeds the move from centroid t
            </MOVERS>
            <APPLY_TO_POSE>
            </APPLY_TO_POSE>
            <PROTOCOLS>
                        Run docking protocol
                        <Add mover="dock_low"/>
                        <Add mover="srsc" />
                        <Add mover="dock_high" />

                        Minimize interface
                        <Add mover="minimize_interface" />
            </PROTOCOLS>
            <OUTPUT scorefxn="REF2015" />
</ROSETTASCRIPTS>
```

PROTOCOLS

```
<PROTOCOLS>
        Run docking protocol
        <Add mover="dock_low"/>
        <Add mover="srsc" />
        <Add mover="dock_high" />

        Minimize interface
        <Add mover="minimize_interface" />
</PROTOCOLS>
```

In the PROTOCOLS section we can see that we will run 4 movers – which are defined in the MOVERS section above

MOVERS

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
        </SCOREFXNS>
        <TASKOPERATIONS>
                <InitializeFromCommandline name="ifcl"/>
                <RestrictToRepacking name="rtr" />
                Restrict to residues within a distance and vector cutoff of the protein-protein int
                <RestrictToInterfaceVector name="rtiv" chain1_num="1,2" chain2_num="3,4" CB_dist_cu

                Fix residues known experimentally to be critical in interaction
                <PreventResiduesFromRepacking name="prfrp" residues="11,41,345" />
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
                MINIMIZATION MOVERS
                Single cycle of FastRelax to minimize backbone of docking partners
                <FastRelax name="minimize_interface" scorefxn="REF2015" repeats="1" task_operations
                DOCKING MOVERS
                <Docking name="dock_low" score_low="score_docking_low" score_high="REF2015" fullato
                <Docking name="dock_high" score_low="score_docking_low" score_high="REF2015" fullat

                <SaveAndRetrieveSidechains name="srsc" allsc="0" /> Speeds the move from centroid t
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
                Run docking protocol
                <Add mover="dock_low"/>
                <Add mover="srsc" />
                <Add mover="dock_high" />

                Minimize interface
                <Add mover="minimize_interface" />
        </PROTOCOLS>
        <OUTPUT scorefxn="REF2015" />
</ROSETTASCRIPTS>
```

MOVERS

```
<MOVERS>
          MINIMIZATION MOVERS
          Single cycle of FastRelax to minimize backbone of docking partners
          <FastRelax name="minimize_interface" scorefxn="REF2015" repeats="1" task_operations
          DOCKING MOVERS
          <Docking name="dock_low" score_low="score_docking_low" score_high="REF2015" fullato
          <Docking name="dock_high" score_low="score_docking_low" score_high="REF2015" fullat

          <SaveAndRetrieveSidechains name="srsc" allsc="0" /> Speeds the move from centroid t
</MOVERS>
```

In the MOVERS section we've defined 4 different movers

Docking movers

```
<Docking name=(string) score_low=(string) score_high=(string)
fullatom=(bool) local_refine=(bool) jumps=(Integer vector)
optimize_fold_tree=(bool) conserve_foldtree=(bool) design=(bool)
ignore_default_docking_task=(bool) task_operations=(comma-separated list)/>
```

Terms	Definition
score_low	scorefxn for centroid-level docking
score_high	scorefxn for full atom docking
fullatom	if true, do full atom docking
local_refine	if true, skip centroid.
jumps	where should we carry out rigid body motions (not used here)
optimize_fold_tree	should DockingProtocol make the fold tree for this pose
conserve_fold_tree	should DockingProtocol reset the fold tree to the input one after it is done
design	interface design for all chains downstream of the rb_jump
ignore_default_docking_task	allows you to ignore the default docking task and only use the ones defined in your task_operations section

Docking movers

2 Docking Movers have been defined – low res and high res docking

```
<Docking name="dock_low" score_low="score_docking_low" score_high="REF2015"
fullatom="0" local_refine="0” ignore_default_docking_task="0" design="0"
task_operations="ifcl,prfrp" jumps="1"/>

<Docking name="dock_high" score_low="score_docking_low" score_high="REF2015"
fullatom="1" local_refine="1” design="0" task_operations="ifcl,prfrp"
jumps="1"/>
```

Additionally a mover to speed the move from centroid (low res) to full atom (high res) docking

```
<SaveAndRetrieveSidechains name="srsc" allsc="0" />
```

Minimization movers

Use a single round of FastRelax to refine the backbone of both docking partners

```
<FastRelax name=”minimize_interface” scorefxn=”REF2015” repeats=”1” task_operations=”ifcl,rtr,rtiv,prfrp” />
```

TASKOPERATIONS

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
        </SCOREFXNS>
        <TASKOPERATIONS>
                <InitializeFromCommandline name="ifcl"/>
                <RestrictToRepacking name="rtr" />
                Restrict to residues within a distance and vector cutoff of the protein-protein int
                <RestrictToInterfaceVector name="rtiv" chain1_num="1,2" chain2_num="3,4" CB_dist_cu

                Fix residues known experimentally to be critical in interaction
                <PreventResiduesFromRepacking name="prfrp" residues="11,41,345" />
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
                MINIMIZATION MOVERS
                Single cycle of FastRelax to minimize backbone of docking partners
                <FastRelax name="minimize_interface" scorefxn="REF2015" repeats="1" task_operations
                DOCKING MOVERS
                <Docking name="dock_low" score_low="score_docking_low" score_high="REF2015" fullato
                <Docking name="dock_high" score_low="score_docking_low" score_high="REF2015" fullat

                <SaveAndRetrieveSidechains name="srsc" allsc="0" /> Speeds the move from centroid t
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
                Run docking protocol
                <Add mover="dock_low"/>
                <Add mover="srsc" />
                <Add mover="dock_high" />

                Minimize interface
                <Add mover="minimize_interface" />
        </PROTOCOLS>
        <OUTPUT scorefxn="REF2015" />
</ROSETTASCRIPTS>
```

TaskOperations

```
<TASKOPERATIONS>
        <InitializeFromCommandline name=”ifcl”/>
        <RestrictToRepacking name=”rtr” />

        <RestrictToInterfaceVector name=”rtiv” chain1_num=”1,2”
chain2_num=”3,4” CB_dist_cutoff=”10.0” nearby_atom_cutoff=”5.5”
vector_angle_cutoff=”75” vector_dist_cutoff=”9.0” />

        <PreventResiduesFromRepacking name=”prfrp” residues=”11,41,345” />

</TASKOPERATIONS>
```

Initialize from command line – allows the mover to read options from the command line

Restrict to repacking – default behavior is design (very important)

Restrict to Interface Vector – restrict which residues are operated on within a given geometry

Prevent Res From Repacking – stop experimentally known residues from repacking

Preparing your options file

Open the docking.options file:

```
gedit ~/rosetta_workshop/tutorials/proteinprotein_docking/
input_files/docking.options
```

docking.options

```
-docking                                                    # the docking option group
        -partners AB_HL                                     # set rigid body docking partners
        -dock_pert 3 8                                      # set coarse perturbation parameters (degrees and angstroms)
        -dock_mcm_trans_magnitude 0.1                       # refinement translational perturbation
        -dock_mcm_rot_magnitude 5.0                         # refinement rotational perturbation
-s 3gbm_HA_3gbn_Ab.pdb                                      # input model
-run:max_retry_job 10                                       # if the mover fails, retry 10 times
-use_input_sc                                               # add the side chains from the input pdb to the rotamer library
-ex1                                                        # increase rotamer bins to include mean +- 1 standard deviation
-ex2                                                        # increase rotamer bins to include mean +- 2 standard deviations
-out                                                        # out option group
        -file                                               # out:file option group
                -scorefile docking.fasc                     # the name of the model score file
-score:weights ref2015.wts                                  # Set ref2015 as default score function
```

Important options

Option Terms	Definition
docking:partners AB_HL	sets rigid body docking partners – chains must be in this order in the PDB
docking:randomize 1	randomize the first docking partner – should be passed to run global docking (not used in this tutorial)
docking:dock_pert 3 8	set coarse perturbation parameters (angstroms and degrees)
docking:dock_mcm_trans_magnitude 0.1	refinement translational perturbation
docking:dock_mcm_rot_magnitude 5.0	refinement rotational perturbation

Step 2 – Run Rosetta (see tutorial)

Step 3 – Analyze Results

Perform analysis

Score vs. RMSD plots to identify a docking funnel

- In experimental studies, the 'native structure' is a good scoring structure, one that fulfills experimental restraints, or a cluster center

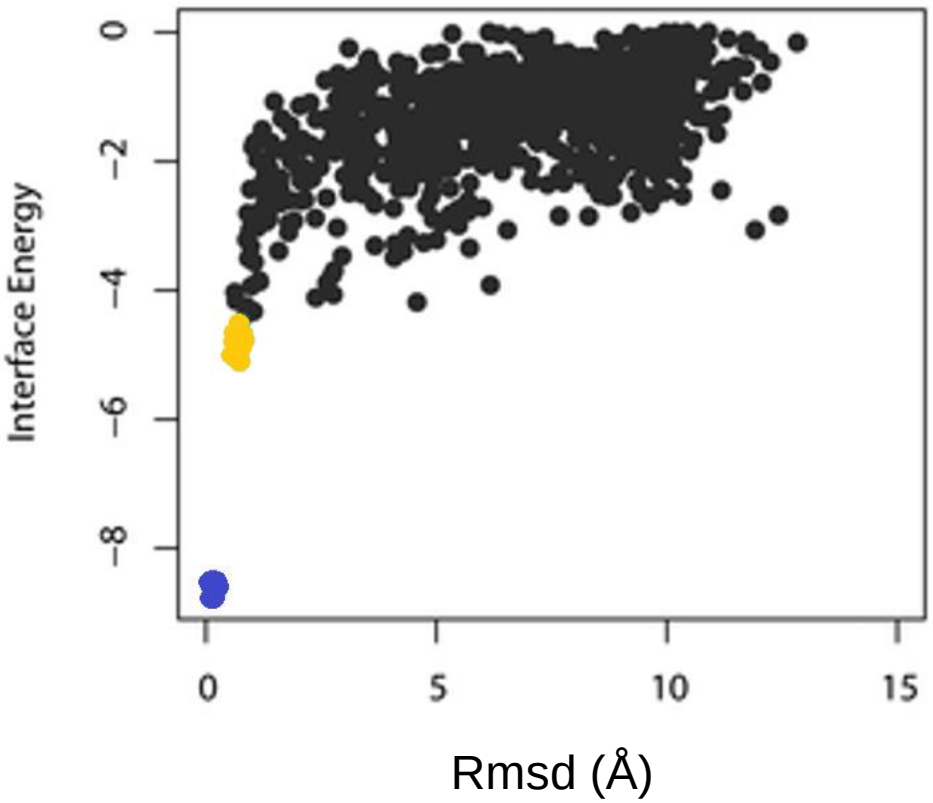
It may be necessary to cluster and dock again, using the top cluster centers as starting templates. This will help to drive the models into an energy well. The 'correct' model will continue down into an energy well while incorrect models will stall.

Look at the structures

- Unsatisfied polar residues
- Binding density
- Average Degree of Residue Burial

Fleishman, S. J.; Baker, D., Role of the biomolecular energy gap in protein design, structure, and evolution. *Cell* **2012,** *149* (2), 262-73
Fleishman, S. J.; Whitehead, T. A.; Strauch, E. M.; Corn, J. E.; … ; Baker, D., Community-wide assessment of protein-interface modeling suggests improvements to design methodology. *Journal of molecular biology* **2011,** *414* (2), 289-302.

Score vs RMSD plots – identifying a docking funnel



Key features:

- Native complex is kept at low RMSD, low scoring
- Good models are sampled (< 2 A RMSD)
- Score function distinguishes between low and high RMSD models

Black – Full docking protocol

Yellow – Top models

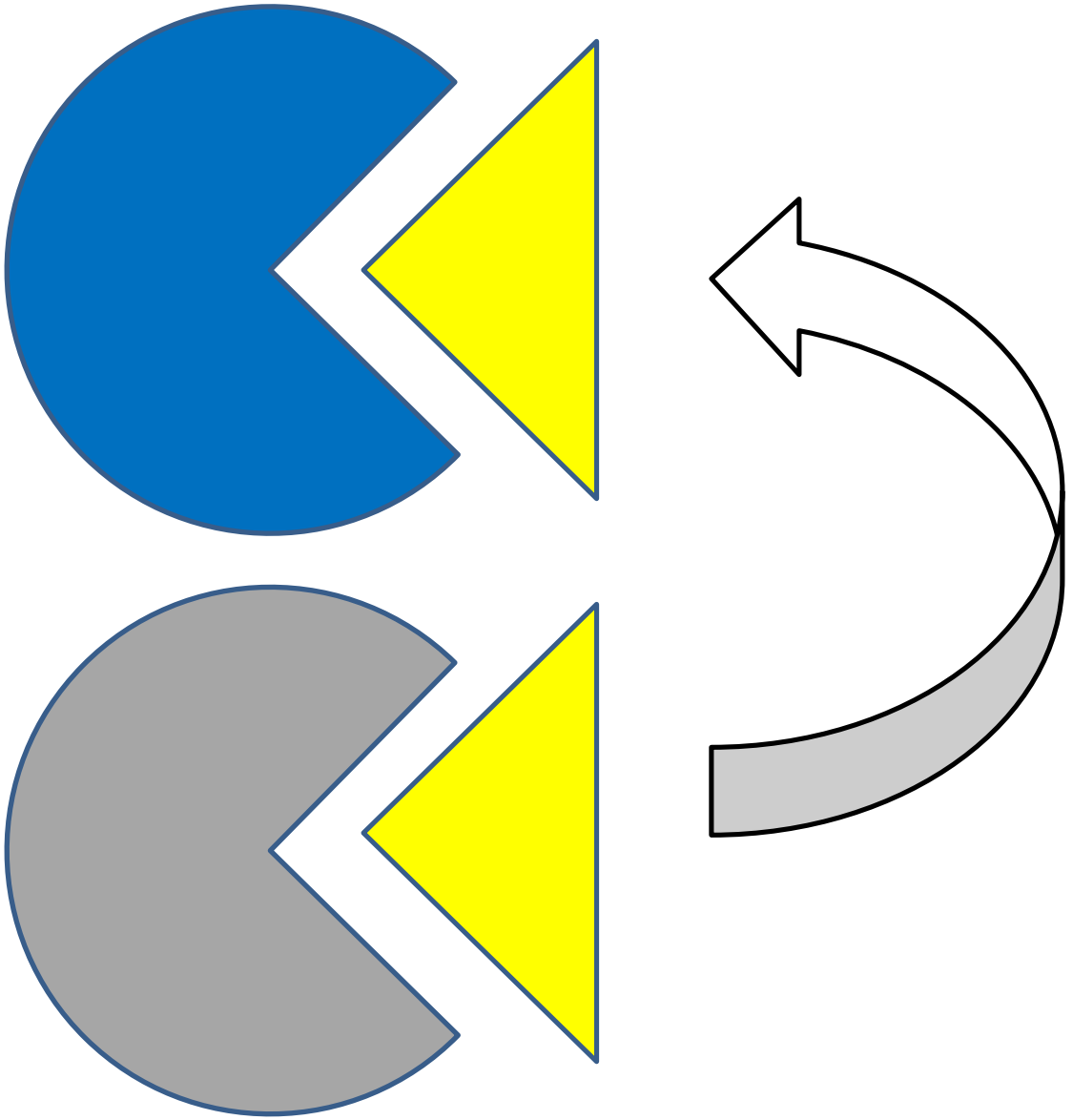
Blue – Refinement of native complex

Today's Tutorial

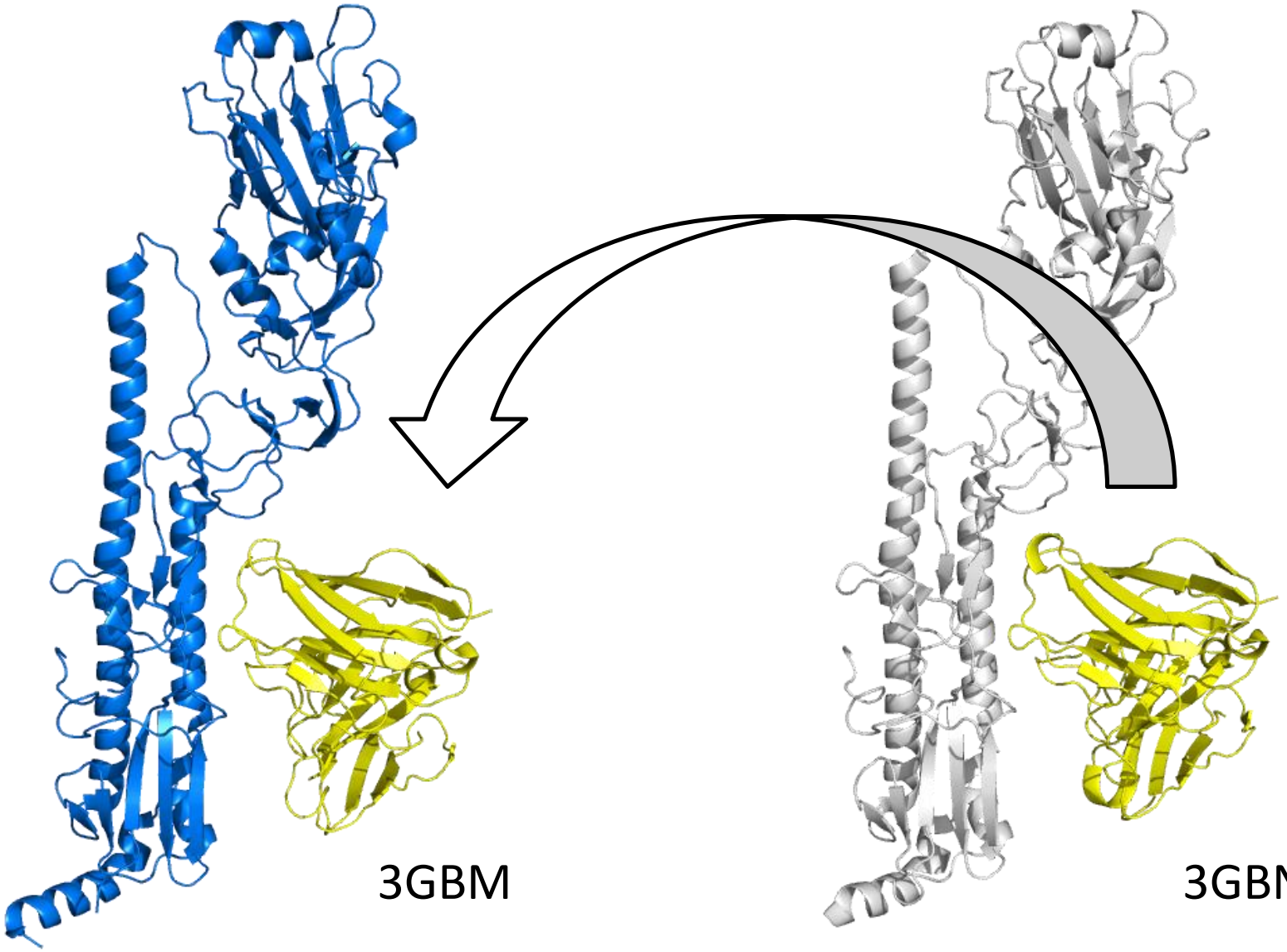
```
cd ~/rosetta_workshop/tutorials/
protein-protein_docking/

firefox protein-
protein_docking_tutorial.html
```

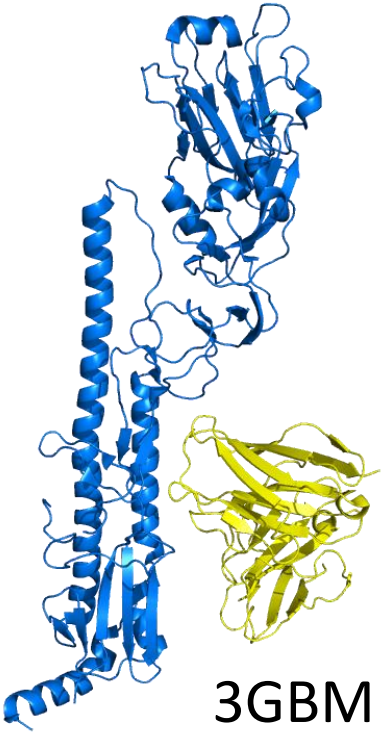
Cross docking experiment



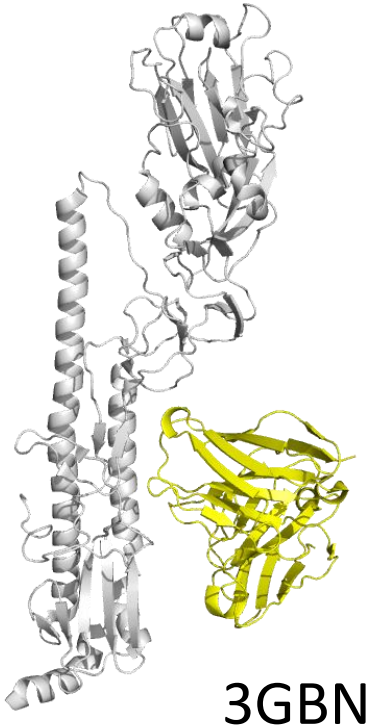
Ab CR6261 bound to both H5 and H1 HA



Ab CR6261 bound to both H5 and H1 HA



3GBM



3GBN

Cross-docking experiment – HA from PDB 3GBM, Ab from PDB 3GBN

The Ab have the same sequence, but slightly different structure

Not very interesting – **but important!** Simple benchmarking (or control) experiment, does something that should bind actually bind in Rosetta

Reference List

Docking

- Gray, J. J.; Moughon, S.; Wang, C.; Schueler-Furman, O.; Kuhlman, B.; Rohl, C. A.; Baker, D., Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of molecular biology* **2003,** *331* (1), 281-99.
- Daily, M. D.; Masica, D.; Sivasubramanian, A.; Somarouthu, S.; Gray, J. J., CAPRI rounds 3-5 reveal promising successes and future challenges for RosettaDock. *Proteins* **2005,** *60* (2), 181-6.
- Chaudhury, S.; Sircar, A.; Sivasubramanian, A.; Berrondo, M.; Gray, J. J., Incorporating biochemical information and backbone flexibility in RosettaDock for CAPRI rounds 6-12. *Proteins* **2007,** *69* (4), 793-800.
- Sircar, A.; Chaudhury, S.; Kilambi, K. P.; Berrondo, M.; Gray, J. J., A generalized approach to sampling backbone conformations with RosettaDock for CAPRI rounds 13-19. *Proteins* **2010,** *78* (15), 3115-23.
- Sivasubramanian, A.; Sircar, A.; Chaudhury, S.; Gray, J. J., Toward high-resolution homology modeling of antibody Fv regions and application to antibody-antigen docking. *Proteins* **2009,** *74* (2), 497-514.
- Sircar, A.; Gray, J. J., SnugDock: paratope structural optimization during antibody-antigen docking compensates for errors in antibody homology models. *PLoS computational biology* **2010,** *6* (1), e1000644

Reference list continued

Design

- Fleishman, S. J.; Whitehead, T. A.; Ekiert, D. C.; Dreyfus, C.; Corn, J. E.; Strauch, E. M.; Wilson, I. A.; Baker, D., Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science* **2011,** *332* (6031), 816-21.
- Fleishman, S. J.; Corn, J. E.; Strauch, E. M.; Whitehead, T. A.; Karanicolas, J.; Baker, D., Hotspot-centric de novo design of protein binders. *Journal of molecular biology* **2011,** *413* (5), 1047-62.
- Der, B. S.; Machius, M.; Miley, M. J.; Mills, J. L.; Szyperski, T.; Kuhlman, B., Metal-mediated affinity and orientation specificity in a computationally designed protein homodimer. *Journal of the American Chemical Society* **2012,** *134* (1), 375-85.
- Mandell, D. J.; Kortemme, T., Computer-aided design of functional protein interactions. *Nature chemical biology* **2009,** *5* (11), 797-807.
- Joachimiak, L. A.; Kortemme, T.; Stoddard, B. L.; Baker, D., Computational design of a new hydrogen bond network and at least a 300-fold specificity switch at a protein-protein interface. *Journal of molecular biology* **2006,** *361* (1), 195-208.

Analysis

- Fleishman, S. J.; Baker, D., Role of the biomolecular energy gap in protein design, structure, and evolution. *Cell* **2012,** *149* (2), 262-73.
- Fleishman, S. J.; Khare, S. D.; Koga, N.; Baker, D., Restricted sidechain plasticity in the structures of native proteins and complexes. *Protein science : a publication of the Protein Society* **2011,** *20* (4), 753-7.
- Fleishman, S. J.; Whitehead, T. A.; Strauch, E. M.; Corn, J. E.; … ; Baker, D., Community-wide assessment of protein-interface modeling suggests improvements to design methodology. *Journal of molecular biology* **2011,** *414* (2), 289-302.