Integrated Modeling with IMP (the Integrative Modeling Platform)

03/06/15

Benjamin Webb
Dina Schneidman
Charles H. Greenberg
Seung Joong Kim

Sali Lab, University of California San Francisco

All workshop materials (if you want to follow along on your laptop) are available from:

http://integrativemodeling.org/keystone2015/

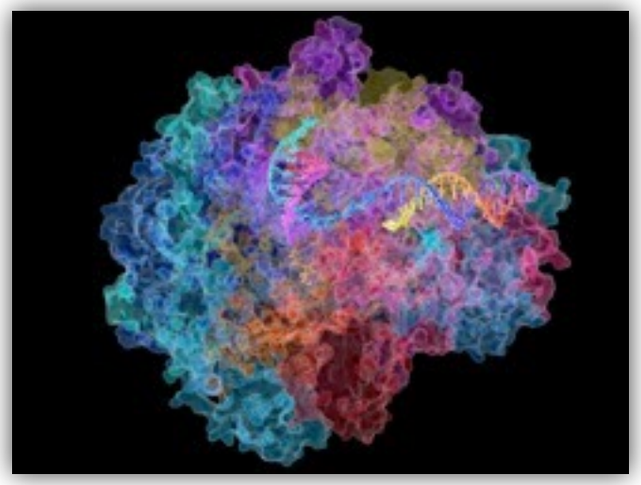
Outcome of the First Hybrid / Integrative Methods Validation Task Force Workshop

Andrej Sali, Torsten Schwede, Jill Trewhella, Helen M. Berman, Gerard Kleywegt, Stephen K. Burley, John Markley, Haruki Nakamura, Paul Adams, Alexandre Bonvin, Wah Chiu, Tom Ferrin[6], Kay Grünewald, Aleksandras Gutmanas[4], Richard Henderson, Gerhard Hummer, Kenji Iwasaki, Graham Johnson, Cathy Lawson[5], Frank di Maio, Jens Meiler, Marc Marti-Renom, Guy Montelione, Michael Nilges, Ruth Nussinov, Ardan Patwardhan, Matteo dal Peraro, Juri Rappsilber, Randy Read, Helen Saibil, Gunnar Schröder, Charles Schwieters, Claus Seidel, Dmitri Svergun, Maya Topf, Eldon Ulrich, Sameer Velankar[4], and John D. Westbrook

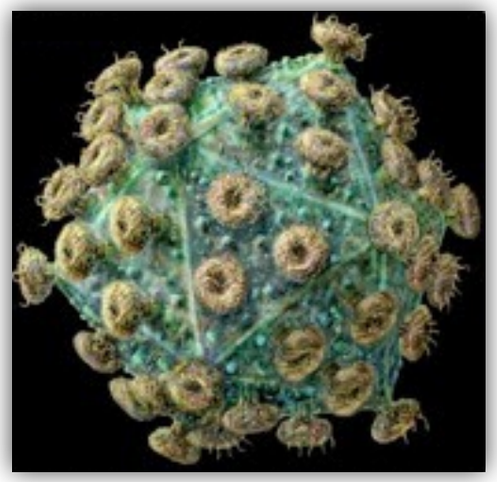
We describe the proceedings and conclusions from the first Hybrid / Integrative Methods Validation Task Force Workshop that was held at the European Bioinformatics Institute in Hinxton, UK, on October 6 and 7, 2014. At the workshop, experts in the various experimental fields that are contributing to these integrative studies, experts in integrative modeling, and experts in data archiving addressed a series of central questions. **What data should be archived? How should integrative models be represented? How should the data and integrative models be validated? How should the data and models be archived? What information should be contained in publication of integrative models?**

Structural biology:
Maximize accuracy, resolution, completeness, and efficiency of the structural coverage of macromolecular assemblies

Motivation: Models will allow us to understand how machines work, how they evolved, how they can be controlled, modified, and perhaps even designed.



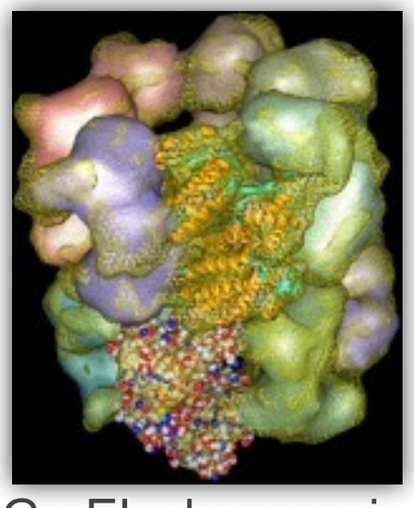
RNA polymerase II



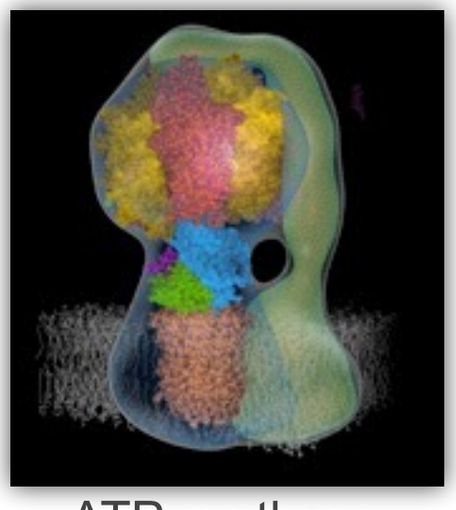
HIV virus

tRNA synthetase

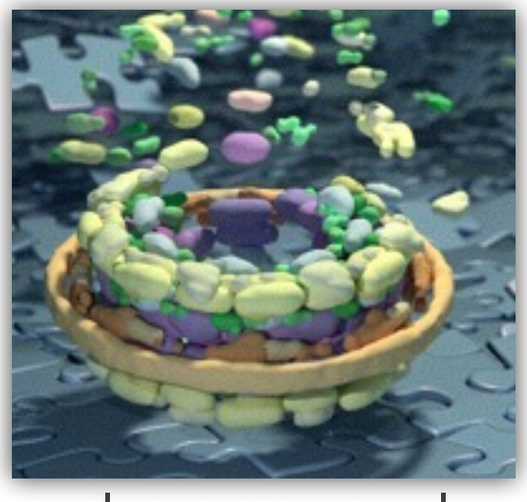
flagellar motor



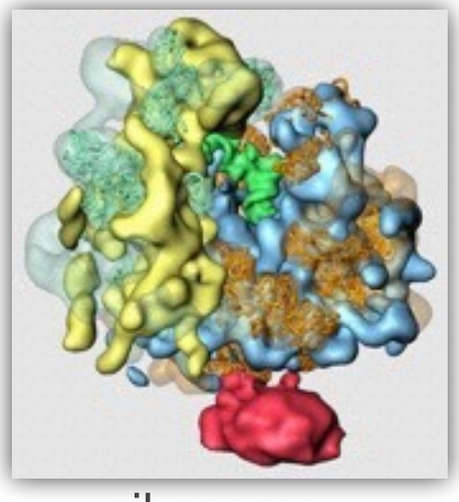
GroEL chaperonin



ATP synthase



nuclear pore complex



ribosome

There may be thousands of biologically relevant macromolecular complexes whose structures are yet to be characterized, involved in a few hundred core biological processes.

Integrative Structural Biology

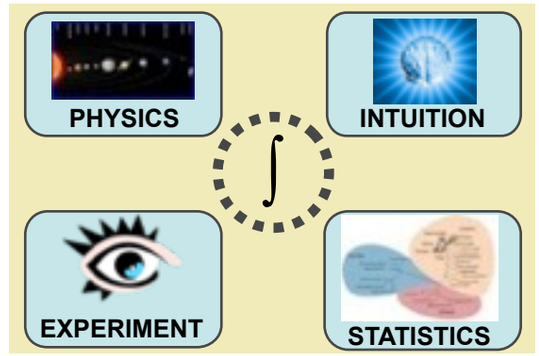
for maximizing accuracy, resolution, completeness, and efficiency of structure determination

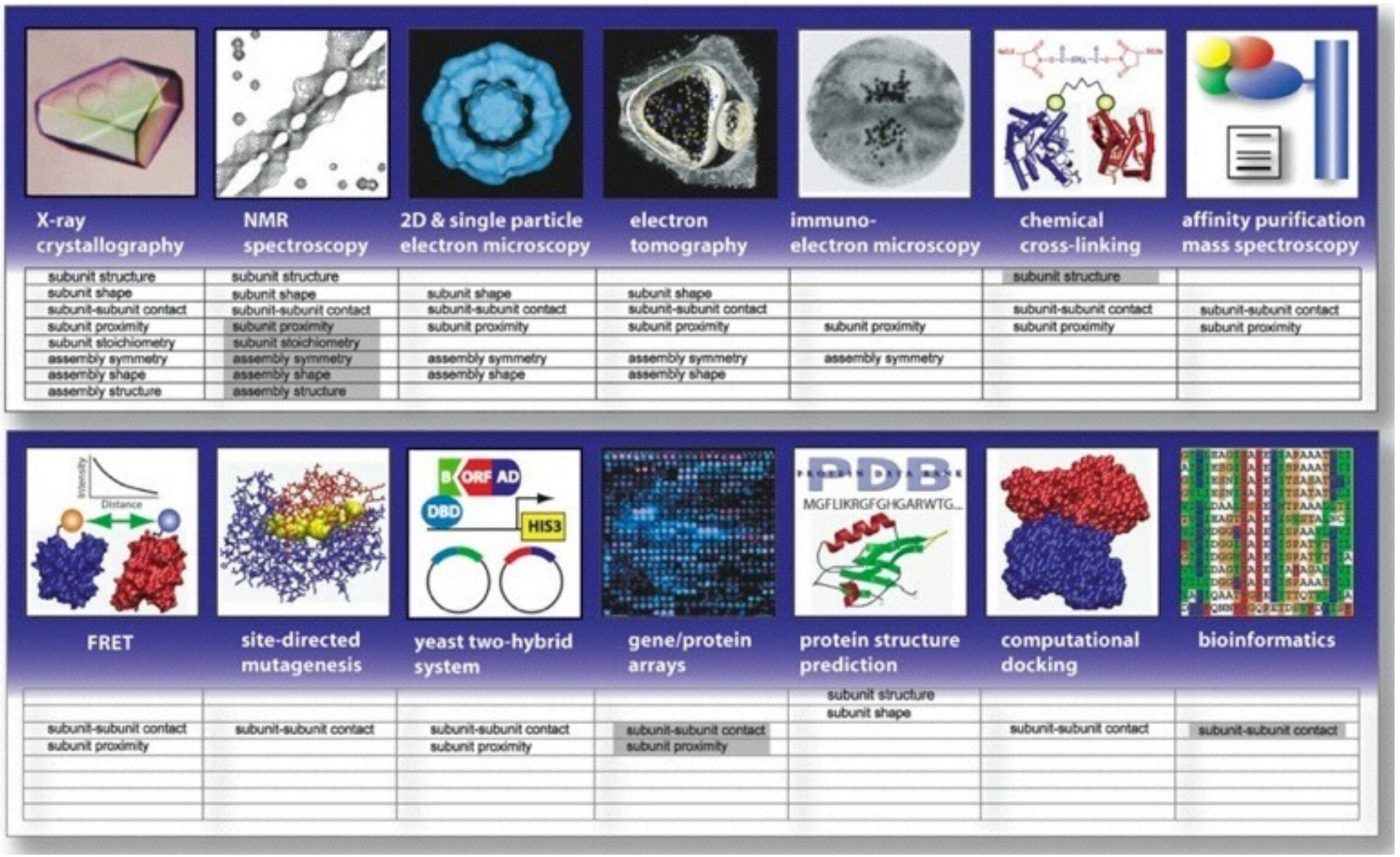
Use structural information from any

source: measurement, first principles, rules;

resolution: low or high resolution

to obtain the set of all models that are consistent with it.





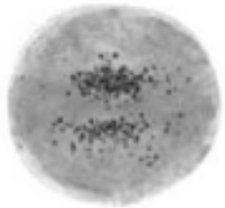
Sali A, Earnest T, Glaeser R, Baumeister W. From words to literature in structural proteomics. *Nature* 422, 216-225, 2003.
Ward A, Sali A, Wilson I. Integrative structural biology. *Science* 339, 913-915, 2013.

A description of integrative structure determination

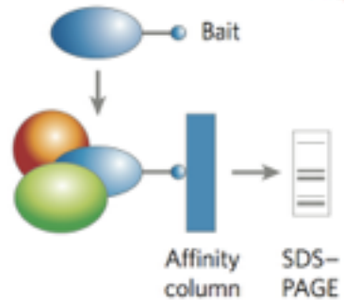
Alber *et al*. *Nature* **450**, 683-694, 2007
Robinson *et al*. *Nature* **450**, 974-982, 2007
Alber *et al*. *Annual Reviews in Biochemistry* **77**, 11.1–11.35, 2008
Russel *et al*. *PLoS Biology* **10**, 2012
Ward *et al*. *Science* **339**, 913-915, 2013
Schneidman *et al*. *Curr.Opin.Str.Biol*., 2014.

Gathering information

Cryo-electron microscopy

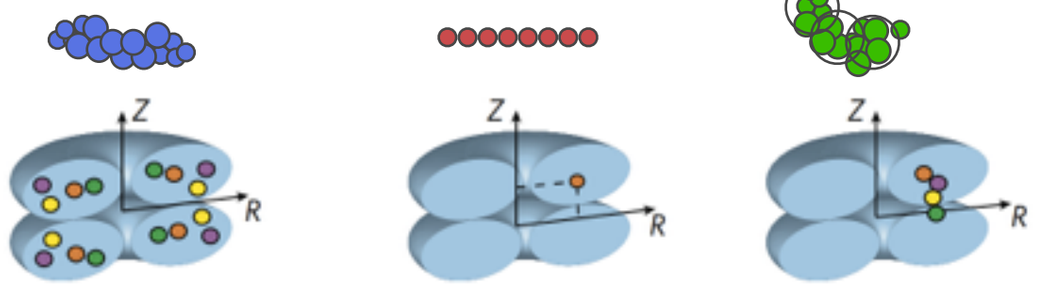


Immuno-electron microscopy

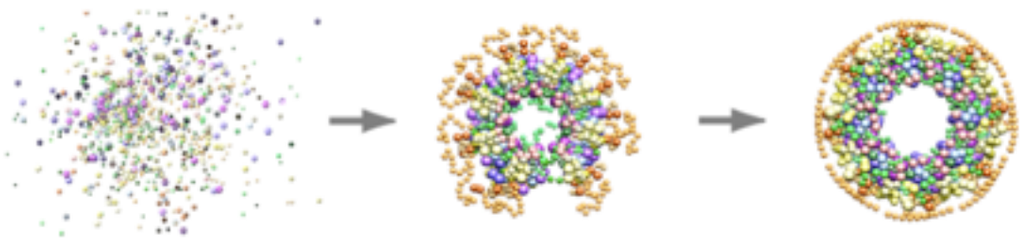


Affinity purification

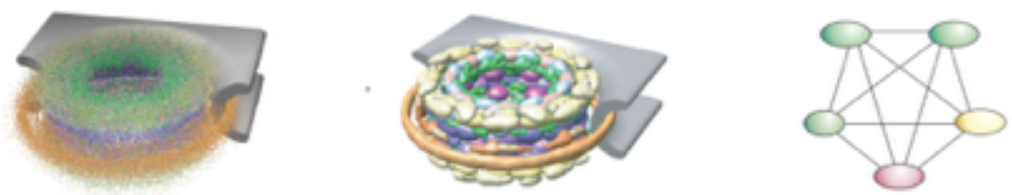
Designing model representation and evaluation



Sampling good models



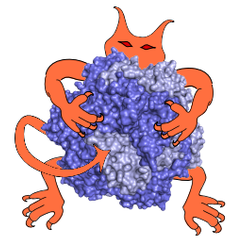
Analyzing models and information



While it may be hard to live with generalization, it is inconceivable to live without it. *Peter Gay*, Schnitzler's Century (2002).

Integrative Modeling Platform (IMP)

http://integrativemodeling.org

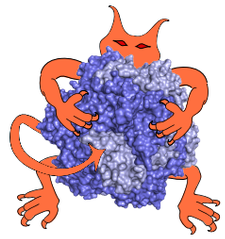


D. Russel, K. Lasker, B. Webb, J. Velazquez-Muriel, E. Tjioe, D. Schneidman, F. Alber, B. Peterson, A. Sali, PLoS Biol, 2012.
R. Pellarin, M. Bonomi, B. Raveh, S. Calhoun, C. Greenberg, G.Dong.

- Our package to solve integrative modeling problems
- Open source (LGPL)
- A collection of C++ classes spread over multiple modules (each with its own namespace)
  - e.g. IMP::atom incorporates classes for handling atomic structures; IMP::em handles electron microscopy data
- In principle, could link directly to the C++ code…

Integrative Modeling Platform (IMP)

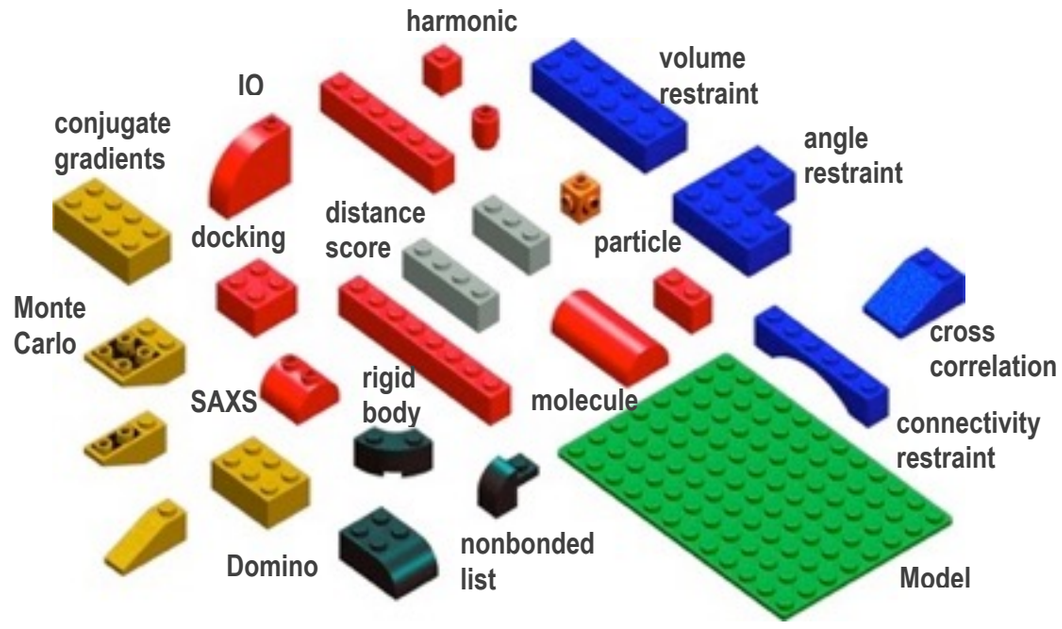
http://integrativemodeling.org



- … but each C++ class is wrapped with an equivalent Python class
  - in practice, IMP is usually used from Python, by writing a script
  - each C++ namespace is a Python package, e.g. IMP.atom, IMP.em
- Link via Python to other packages to avoid code duplication, e.g.
  - MODELLER (via its own Python interface) for comparative modeling
  - BioPython for handling of sequence data
  - scikit-learn for clustering
  - numpy/scipy for matrix/linear algebra
  - etc.

Mix and match classes

- Think of IMP classes as “mix and match” components for developing an integrative modeling protocol
- Protocol is typically Python script(s) plus input data



Representation:

Atomic
Rigid bodies
Coarse-grained
Multi-scale
Symmetry / periodicity
Multi-state systems

Scoring:

Density maps
EM images
Proteomics
FRET
Chemical and Cys cross-linking
Homology-derived restraints
SAXS
Native mass spectrometry
Statistical potentials
Molecular mechanics forcefields
Bayesian scoring
Library of functional forms (ambiguity, ...)

Sampling:

Simplex
Conjugate Gradients
Monte Carlo
Brownian Dynamics
Molecular Dynamics
Replica Exchange
Divide-and-conquer enumeration

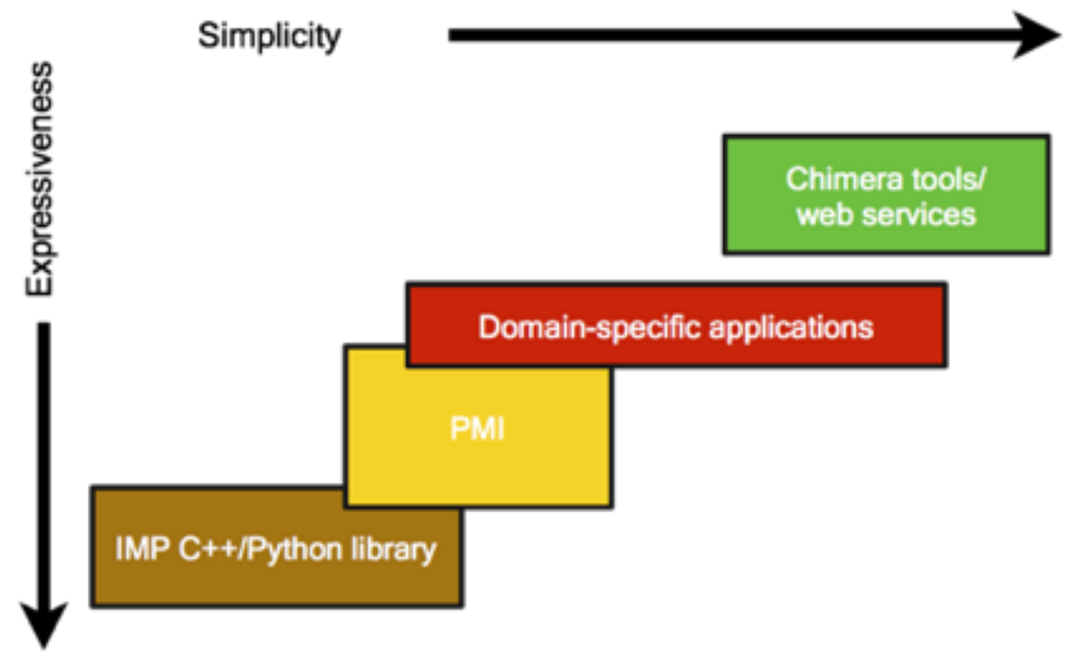
Analysis:

Clustering
Chimera
Pymol
PDB files
Density maps

Example Python script

```
import IMP
import IMP.algebra
import IMP.core

m = IMP.Model()
# Create two "untyped" Particles
p1 = IMP.Particle(m)
p2 = IMP.Particle(m)

# "Decorate" the Particles with x,y,z attributes (point-like particles)
d1 = IMP.core.XYZ.setup_particle(p1)
d2 = IMP.core.XYZ.setup_particle(p2)

# Use some XYZ-specific functionality (set coordinates)
d1.set_coordinates(IMP.algebra.Vector3D(10.0, 10.0, 10.0))
d2.set_coordinates(IMP.algebra.Vector3D(-10.0, -10.0, -10.0))
print d1, d2

# Harmonically restrain p1 to be zero distance from the origin
f = IMP.core.Harmonic(0.0, 1.0)
s = IMP.core.DistanceToSingletonScore(f, IMP.algebra.Vector3D(0., 0., 0.))
r1 = IMP.core.SingletonRestraint(s, p1)
m.add_restraint(r1)

# Harmonically restrain p1 and p2 to be distance 5.0 apart
f = IMP.core.Harmonic(5.0, 1.0)
s = IMP.core.DistancePairScore(f)
r2 = IMP.core.PairRestraint(s, (p1, p2))
m.add_restraint(r2)

# Optimize the x,y,z coordinates of both particles with conjugate gradients
d1.set_coordinates_are_optimized(True)
d2.set_coordinates_are_optimized(True)
o = IMP.core.ConjugateGradients(m)
o.optimize(50)
print d1, d2
```

Higher level interfaces



- In practice, scripts for “real” modeling problems tend to be long and unwieldy
- We provide web services, Chimera interfaces, and command line tools for some simple applications
- But today, we will look at the middle ground: PMI

PMI

- Just another IMP module (IMP.pmi)
- A meta language for modeling
- We still write Python scripts, but…
  - Many protocols (e.g. replica exchange) already packaged up nicely for us, support for multi-scale by default
  - Refer to biological units rather than individual particles
  - Publication-ready plots are more or less automatic
- Regular IMP objects are constructed, so an advanced user can always customize things using the full collection of IMP classes if PMI is insufficient
- Today we will use PMI to model the stalk of the RNA Polymerase II complex: http://integrativemodeling.org/2.4.0/doc/tutorial/rnapolii_stalk.html