A Framework for Understanding Rosetta
Xavier Ambroggio
Rosetta Design Group
Origin of Rosetta
Introduction to Basic Rosetta Methodology
Overview of Rosetta Implementation
Rosetta: an algorithm for *ab initio* structure prediction

**AB INITIO: PREDICTION REPORTS**

Ab Initio Protein Structure Prediction of CASP III Targets Using ROSETTA

Kim T. Simons,1 Rich Bonneau,1 Ingo Ruczinski,2 and David Baker1*

1Department of Biochemistry, University of Washington, Seattle, Washington
2Department of Statistics, University of Washington, Seattle, Washington

Critical Assessment of Techniques for Protein Structure Prediction

CTWEGNKLTC

protein folding problem
CASP8

Structure determination

Give sequences to Organisers
Keep structures secret (if known)

Predict Structure from Sequence
(wait nervously)

Call for structures
Publish seqs on www
Collect predictions

Organisers

4 day meeting to discuss results

Predictors

Structural Biologists

JAN–APR MAY JUN JUL AUG SEP OCT NOV DEC
Functional expansion of Rosetta algorithms

- *ab initio* folding
- design
- docking
- protein-protein interactions
  - ligand docking
  - enzyme design
  - etc.

inverse protein folding

CTWEGNKLTC
Introduction to Basic Rosetta Methodology

- States & State Changes
- Scoring Functions
- Search & Optimization Routines
- Output
States Used in **Rosetta**
*State = Discrete Conformational Unit*

- **Primary structure**
  - Lys
  - Lys
  - Gly
  - Gly
  - Leu
  - Val
  - Ala
  - His

- **Secondary structure**
  - α Helix fragment

- **Tertiary structure**
  - Polypeptide chain

- **Quaternary structure**

- **Amino acid residues**
  - sequence

- **Assembled subunits**
  - subunits
  - decoy
  - pose
  - ligand

- **Dihedral, torsion angle**

- **Rotamer**
States & State Changes

- **sequences**
  - static state for folding & loop modeling
  - amino acid substitutions in parallel design

- rotamers
- dihedrals
- fragments
- ligands
- protein subunits
- pose & fold trees
Rotamers
States for full-atom scoring and design

**Rotamers (rotational isomers):**
- highly populated combinations of side-chain dihedral angles.
  - low energy side-chain conformations.
- a small library of about 100-150 rotamers can cover 96-97% of the conformations found in protein structures.

**Dunbrack rotamer libraries:**
*Backbone dependent* and independent libraries.
rossetta_database/bbdep02.May.sortlib

rotamer move = substitution
Dihedrals

*States used in most protocols*

Small scale dihedral moves (i.e. refinement, minimization)

- Random torsion angle perturbation
  - “small” = randomly perturb paired phi, psi
  - “shear” = randomly perturb phi, equal & opposite perturbation to preceding psi

- *fragment insertion*
  - rapid torsion angle optimization to offset global perturbations
  - “wobble” = continuous variation of phi, psi near perturbation to minimize downstream MSD

- gradient descent = $dE / d\phi,\psi$ evaluated, followed by...
  - linmin (line searches):
    - find minimum in direction of steepest descent and stop
    - not the best way to explore a complex landscape
  - dfpmin (Davidson, Fletcher, Pal - quasi-Newton method):
    - the core minimization routine
    - iterations of moves and derivative calculations
    - smarter than steepest descent
Fragments

- Definition
- Fragment moves
Fragments
States for ab initio and loop modeling

- 3 and 9 residue fragments
- database created from crystal structures
  - < 2.5Å resolution
  - < 50% sequence identity
- rosetta_fragments/nnmake_database/vall.dat.2006-05-05
- *custom fragment database possible*
- low resolution modeling
  - centroid representation of side chains
Making Fragment Libraries

Overview

Fragments are selected from database and ranked according to:

- input amino acid sequence
  - FASTA format
  - possible to use only secondary structure information
- secondary structure predictions
  - programs
    - PSI-PRED
      - default and predictions carry largest weight
    - JUFO
    - SAM
    - PROF
  - more = better
  - manual

Note: we are leaving “Rosetta”
Fragment insertion

- conformation modification occurs in torsion space
- small changes in dihedrals
  - “chuck” = fragments that result in MSD of atoms below threshold randomly inserted (Cartesian)
  - “Gunn” = fragments that result in translation & rotation below threshold are randomly inserted (independent of coordinate system)
  - “crank” = “chuck” + “wobble”
Ligands

- biochemical definition
- metals, small-molecules, etc.
- (<200 non-hydrogen atoms)
- ligand moves
Ligand Moves

*analog of protein design with flexible backbone (& docking)*

1 (Setup)

- **Ligand**
  - Precompute interactions for *ligand library* of likely conformations

- **Protein**
  - Precompute interactions for *rotamer library* of likely side chain conformations

2 (coarse discrete optimization)

- **Ligand**
  - Replacement of ligand conformations (and identities)

3 (fine continuous optimization)

- **Ligand**
  - Minimization of ligand conformation, *orientation*, and *translation*

- **Protein**
  - Replacement of rotamers (and amino acid identities)
  - Minimization of protein backbone and amino acid side chain conformations.

Slide content credits:
Jens Meiler
Pose & Fold Trees

Methodological Inconvenience

Rosetta folding

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8

3 backbone dihedral angles per residue

Sampling and minimization in TORSIONAL space

Rosetta docking

1 2 3 4 5 6 7 8

Backbone dihedral angles fixed (rigid-body)

1′ 2′ 3′ 4′ 5′ 6′ 7′ 8′

6 rigid-body DOFs --
3 translational vectors
3 rotational angles
Pose & Fold Trees

Fold tree representation

Allows simultaneous optimization of rigid-body and backbone/sidechain torsional degrees of freedom.

- "long-range" edge – 6 rigid-body DOFs
- "peptide" edge – 3 backbone dihedral angles

fold-tree based docking

- Construct fold-trees to treat a variety of protein folding and docking problems.

Bradley and Baker, *Proteins* 2006
Energy Functions

- purpose: *score states*
- major classes
  - low resolution
  - high resolution
Major Classes of Energy Functions

- **Low resolution**: *reduced atom representation*
  - simplified energy function
  - used for aggressive search of state space

- **High resolution**: *full-atom representation*
  - detailed energy function
  - local search of state space
  - refinement and minimization
Low resolution:

Atom Model
- centroid reduction of side chains

Energy function terms
- van der Waals repulsion
- “pair” terms (electrostatics)
- residue environment (prob of burial)
- 2º structure pairing terms (H-bonds)
- radius of gyration
- packing density

In general …
- Weighted linear combination
  \[ \text{Energy} = w_1 \cdot \text{term}_1 + w_2 \cdot \text{term}_2 + \ldots \]
- Pair-wise decomposable
- Heavily trained on PDB statistics
  - Discriminate “near native” vs “non native”
- No single low resolution score
  - Several functions with different weights
Rosetta Energy Function

Low resolution:

Implicit terms

fragments (local interactions)

non-redundant protein structures

slide content credits:
Glenn Butterfoss
Rosetta Energy Function

High resolution:

Atom Model
- full atom representation

Energy function terms
- Rotamer (Dunbrack)
- Ramachandran
- Solvation (Lazaridius Karplus)
- Hydrogen bonding
- Lennard-Jones
- Pair (electrostatic)
- Reference energies

In general ...

Weighted linear combination

\[ \text{Energy} = w_1 \cdot \text{term}_1 + w_2 \cdot \text{term}_2 + \ldots \]

Pair-wise decomposable

Pre-tabulate energies

Hybrid Statistical / MM-like score

Weights trained for different applications

slide content credits:
Glenn Butterfoss
Search and Optimization

- size of state spaces
- algorithm(s)
  - Monte Carlo
  - simulated annealing
  - Metropolis
Approximate size of different state spaces

- **Folding**: given either alpha, beta, or loop conformation, for protein of \( nres \), \( 3^{nres} \) possible conformations.
  - Levinthal paradox (Cyrus Levinthal, *J. Chim. Phys. 65, 44; 1968*):
    - If \( nres = 100 \), sampling a conformation every \( 10^{-13} \) seconds, it would take \( 10^{27} \) years to fold. Universe is \( 10^{10} \) years old.
  - Folding is non-random and cooperative.

- **Design**:
  - for protein of \( nres \), \( 20^{nres} \) possible sequences
  - given 10 rotamers per fixed amino acid, \( 10^{nres} \) possible states

- **Docking**: \( 360^3 \times \text{Angstroms}^3 \) (for 10 Angstroms, \( 4.6 \times 10^{10} \) states)
  - etc.
Basic Rosetta optimization algorithm

**Monte Carlo search**

**Simulated Annealing & Metropolis**

= random state substitutions

= acceptance criterion

“jump size” $\alpha$ temp & energy
Rosetta methodology in real time

**NOTE:** MOVIES REPRESENT SINGLE TRAJECTORIES

*typical simulation involves 100-100000 trajectories*

→ **design movie**
→ **ab initio movie**
→ **docking movie**
Overview of Rosetta output

- decoys and funnels
- computational power versus accuracy
- constraints
- filters
Funnels: decoy RMSD to native versus energy

1 decoy/point = 1 trajectory

Similar energy landscapes for Rosetta predictions:
- **energy function accurately scores states**
- **models can be selected by energy/score only**
Constraint: *user input limitation of state space search*

- **constraint methodology**
  - violation of a constraint increases the decoy score
  - Implemented through files (.cst, .dpl, .dst)

- **types of constraints**
  - mainly apply to *ab initio* mode
  - NMR derived dipolar coupling constraints
  - barcode constraints (features like ss, phi/psi, etc.)
  - distance constraints (docking)

- **future expansion to other modes**
Filters: *absolute constraints*

- filter methodology
  - violation causes decoy to be discarded
  - implemented through command line options
- physical attributes
  - disulfides
  - knot
  - SASA
  - vdw
  - radius of gyration
  - score
  - etc.
Overview of Rosetta Implementation

- Implementation Details of Select Modes
- Brief Description of Select Modes
  - Loop Modeling Protocols
- Introduction to the Rosetta command line
- Flow-chart of Rosetta Execution
## Brief Description of Select Modes

<table>
<thead>
<tr>
<th>mode</th>
<th>description</th>
<th>main flag(s)</th>
<th>main code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab initio</td>
<td>predict the structure from sequence</td>
<td><code>none (original mode)</code></td>
<td>fold_abinitio.cc</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>-abrelax</code></td>
<td></td>
</tr>
<tr>
<td>relax</td>
<td>refine the structure using Rosetta energy functions</td>
<td><code>-relax</code></td>
<td>relax_structure.cc</td>
</tr>
<tr>
<td>idealize</td>
<td>replace bond geometries with ideal values</td>
<td><code>-idealize</code></td>
<td>idealize.cc</td>
</tr>
<tr>
<td>loop modeling</td>
<td>build and refine local structurally variable regions in context of a structural template</td>
<td><code>-loops</code></td>
<td>fold_loops.cc</td>
</tr>
<tr>
<td>design</td>
<td>optimize sequence given a structure</td>
<td><code>-design</code></td>
<td>design_structure.cc</td>
</tr>
<tr>
<td>docking</td>
<td>structure prediction for a protein-protein complex given subunits</td>
<td><code>-docking</code></td>
<td>docking.cc</td>
</tr>
<tr>
<td>ligand</td>
<td>ligand docking, design</td>
<td><code>-ligand</code></td>
<td>ligand.cc</td>
</tr>
<tr>
<td>interface</td>
<td>ddG calculation for mutations made across a complex interface</td>
<td><code>-interface</code></td>
<td>analyze_interface_ddg.cc</td>
</tr>
<tr>
<td>scoring</td>
<td>score input conformations with Rosetta energy functions</td>
<td><code>-score</code></td>
<td>scorefxns.cc</td>
</tr>
<tr>
<td>domain assembly</td>
<td>fixed domains connected by variable regions</td>
<td><code>-assemble</code></td>
<td>assemble_domains.cc</td>
</tr>
<tr>
<td>pose</td>
<td>a set of algorithms which improve previous implementations</td>
<td><code>-pose</code> <code>-pose_*$</code></td>
<td>pose_*.cc</td>
</tr>
<tr>
<td>mode</td>
<td>description</td>
<td>mode</td>
<td>description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ab initio</td>
<td>predict the structure from sequence</td>
<td>relax</td>
<td>a set of algorithms which improve previous implementations</td>
</tr>
<tr>
<td></td>
<td>“ab initio”</td>
<td>idealize</td>
<td>assemble_domains.cc</td>
</tr>
<tr>
<td></td>
<td>“Classical”</td>
<td>loop</td>
<td>fixed domains connected by variable regions</td>
</tr>
<tr>
<td></td>
<td>classical ab initio fragment insertion with minimization</td>
<td>modeling</td>
<td>pose_*.cc</td>
</tr>
<tr>
<td></td>
<td>“Pose-based”</td>
<td>design</td>
<td>a set of algorithms which improve previous implementations</td>
</tr>
<tr>
<td></td>
<td>+ explicit cyclic coordinate descent for loop closure</td>
<td>docking</td>
<td>-score</td>
</tr>
<tr>
<td></td>
<td>“Loop relax”</td>
<td>ligand</td>
<td>score input conformations with Rosetta energy functions</td>
</tr>
<tr>
<td></td>
<td>+ full atom minimization</td>
<td>interface</td>
<td>domain assembly</td>
</tr>
<tr>
<td></td>
<td>“Termini”</td>
<td>scoring</td>
<td>fixed domains connected by variable regions</td>
</tr>
<tr>
<td></td>
<td>centroid based extension of protein termini</td>
<td>-scorefxns.cc</td>
<td>score input conformations with Rosetta energy functions</td>
</tr>
<tr>
<td></td>
<td>“Loop design”</td>
<td></td>
<td>-assemble</td>
</tr>
<tr>
<td></td>
<td>specialized flexible backbone design</td>
<td></td>
<td>assemble_domains.cc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>pose_*.cc</td>
</tr>
</tbody>
</table>

**Loop modeling protocols**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Reference</th>
<th>General characteristics</th>
<th>Differing input files</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Classical”</td>
<td>Carol Rohl et al. Proteins 2004.</td>
<td>classical ab initio fragment insertion with minimization</td>
<td>(1pdbC.ssa)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- secondary structure assignments</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1pdb.loops</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- loop library</td>
</tr>
<tr>
<td>“Pose-based”</td>
<td>Chu Wang et al. JMB 2007</td>
<td>+ explicit cyclic coordinate descent for loop closure</td>
<td>1pdbC.pose_loops</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- loop definitions and options</td>
</tr>
<tr>
<td>“Loop relax”</td>
<td>Bin Qian et al. Nature 2007</td>
<td>+ full atom minimization</td>
<td>1pdbC.loopfile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- loop definitions</td>
</tr>
<tr>
<td>“Termini”</td>
<td>Sood et al. JMB 2006</td>
<td>centroid based extension of protein termini</td>
<td>1pdbC.loops</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- special loop library</td>
</tr>
<tr>
<td>“Loop design”</td>
<td>Xiaozhen Hu et al. PNAS 2007</td>
<td>specialized flexible backbone design</td>
<td>(custom method and inputs, stay tuned...)</td>
</tr>
</tbody>
</table>
Introduction to the Rosetta command line

**UNIX-like:**
executable -flags
e.g. ls -a

```bash
rosetta.exe ar lpdb A -abrelax -nstruct 10000 -seed_offset 1 -ex1 -ex2
```

- executable
- protocol
- random seed value
- number of output structures
- run options
- series code
- protein code
- chain id