Rosetta Basics:
IO and Navigating Rosetta

VANDERBILT
UNIVERSITY

Shannon Smith
Graduate Student | Meiler Lab
Shannon.t.smith.1@Vanderbilt.edu

Goals for this Talk

1. General Rosetta Concepts:
   - How do I run basic Rosetta applications?
   - Input/Output: file types, options, etc.

2. Learn where things are in Rosetta
   - Your working directory is independent of these Rosetta directories (AKA your data is stored outside of Rosetta)

QUESTIONS ARE ENCOURAGED!

This talk is located in:
~/rosetta_workshop/tutorials/short_talks/navigating_rosetta_IO.pptx

Example files are located in:
Files for this talk are in ~/rosetta_workshop/tutorials/short_talks/RosettaIO/

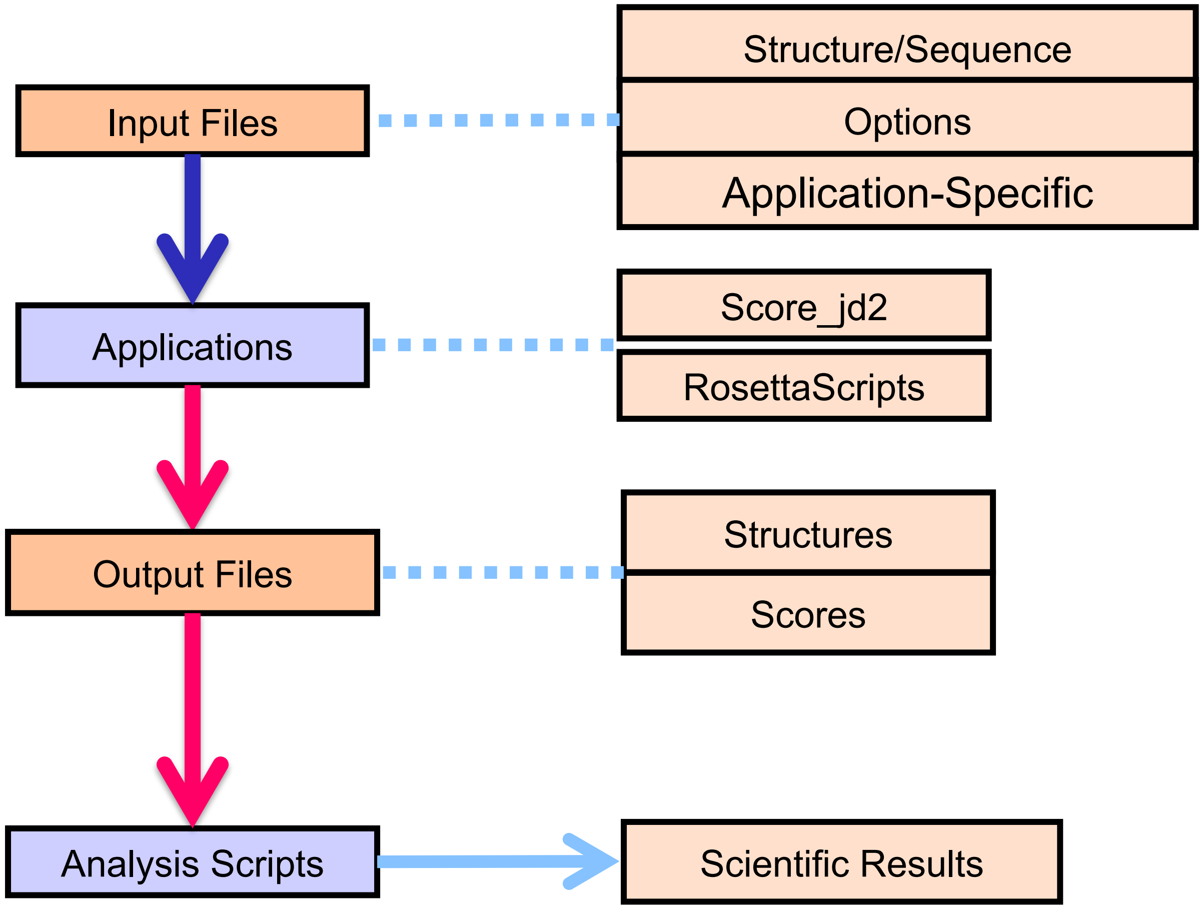
Notice these files are located outside of Rosetta--you do NOT want to store your input/output files in the Rosetta directories

Tip: Open each file in your favorite text editor (gedit, vi, emacs, etc.) as we introduce them through the talk

How do I get Rosetta?

- https://www.rosettacommons.org/software/license-and-download
- Weekly Releases: (e.g. “2020.07”)
  - Latest version of the code, released roughly every week
  - Every revision passes scientific performance tests
- Numbered Releases (e.g. “3.12”)
  - A weekly release that’s relabeled, released roughly every 6 months
- All tutorials use version 3.12
- Links to documentation, forum and demos:
  - https://www.rosettacommons.org/docs/latest/Home
  - https://www.rosettacommons.org/demos/latest/Home

General Workflow



How do I run a Rosetta command?

Every command has the same basic layout:

Path to the Rosetta application

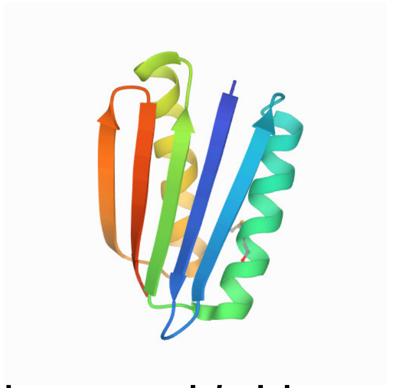
Arguments/flags/options

```
<path_to_rosetta>/main/source/bin/<app_name>.default.linuxgccrelease –arg1 –arg2
```

Arguments consist of multiple things:

1. Point to input files
2. Point to where you want output files to go
3. Other arguments are protocol-dependent

Your first Rosetta command:



- cd ~/rosetta_workshop/short_talks/RosettaIO/
- Scoring 1qys (https://www.rcsb.org/structure/1QYS)
- Crystal structure of Top7: A computationally designed protein with a novel fold

```
<path_to_rosetta>/main/source/bin/score_jd2.default.linuxgccrelease –in:file:s 1qys.pdb -out:pdb > 1qys_score.log
```

- Inputs:
  - Running score_jd2 application, which simply scores in the input protein
  - -in:file:s 1qys.pdb : tells Rosetta we're inputting the 1qys.pdb PDB file
  - -out:pdb argument tells Rosetta that we want to save the output PDB file of the scored protein
  - > 1qys_score.log : print terminal output to file called 1qys_score.log
- Outputs:
  - 1qys_0001.pdb : output PDB
  - score.sc : default name for output scorefile
  - 1qys_score.log: tracer of run AKA what is output to the terminal screen when running command

Reading structures into Rosetta

PDB files

- International standard
- Readable by PyMol, MOE, Chimera, etc
- One line per atom
- Useful for small number of structures

www.wwpdb.org/documentation/file-format

Silent files

- Specific to Rosetta
- Compact
- One line per residue
- Useful for archiving many structures
- Binary silent files: more compact, but not human-readable

https://www.rosettacommons.org/docs/latest/rosetta_basics/file_types/silent-file

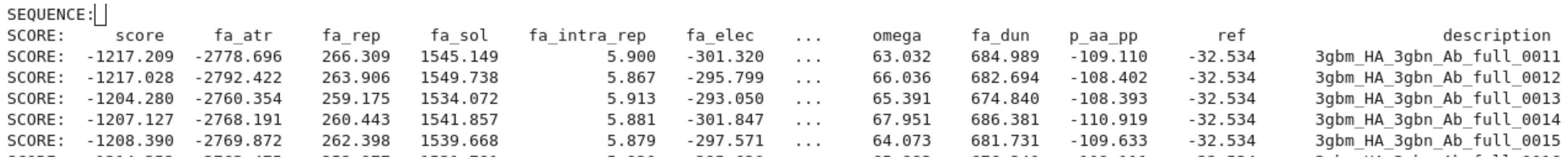
Common command line arguments

- Common input options:
  - -in:file:s example.pdb ## input a PDB structure file
  - -parser:protocol example.xml ## RosettaScripts XML file
  - -in:file:fasta example.fasta ## input a FASTA sequence file
  - -in:file:silent example.silent ## input a Rosetta silent file
  - -nstruct 42 ## produce 42 outputs
- Common output options:
  - -out:file:silent example_out.silent ## output structures to silent file
  - -out:file:scorefile example_out.sc ## output scorefile for run

Examples of output: the scorefile

score.sc

- One output per line—name of output is in the very last column
- Each column defines a specific score term for the respective output structure
- Second column is the "total_score"
- The following columns are individual scoreterms (described in detail in later talk)
- Excerpt of example scorefile here, but recommend you look at your own score.sc output file

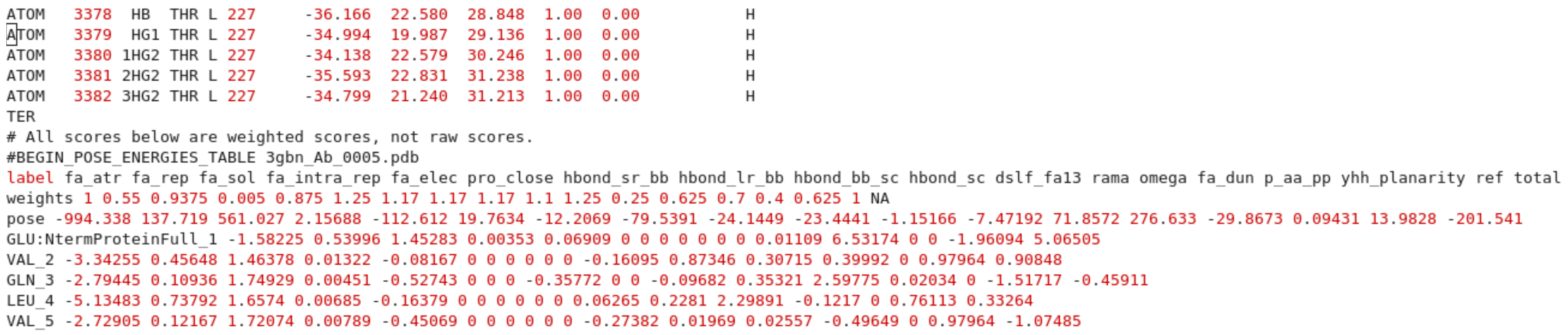


```
SEQUENCE:
SCORE:     score     fa_atr     fa_rep    fa_sol    fa_intra_rep   fa_elec    ...    omega    fa_dun    p_aa_pp      ref              description
SCORE:  -1217.209  -2778.696   266.309  1545.149          5.900  -301.320    ...   63.032   684.989  -109.110   -32.534   3gbm_HA_3gbn_Ab_full_0011
SCORE:  -1217.028  -2792.422   263.906  1549.738          5.867  -295.799    ...   66.036   682.694  -108.402   -32.534   3gbm_HA_3gbn_Ab_full_0012
SCORE:  -1204.280  -2760.354   259.175  1534.072          5.913  -293.050    ...   65.391   674.840  -108.393   -32.534   3gbm_HA_3gbn_Ab_full_0013
SCORE:  -1207.127  -2768.191   260.443  1541.857          5.881  -301.847    ...   67.951   686.381  -110.919   -32.534   3gbm_HA_3gbn_Ab_full_0014
SCORE:  -1208.390  -2769.872   262.398  1539.668          5.879  -297.571    ...   64.073   681.731  -109.633   -32.534   3gbm_HA_3gbn_Ab_full_0015
```
]

Examples of output: the output PDB

1qys_0001.pdb

- One atom/line just like normal PDBs
- Scroll to the bottom and there is per residue score information

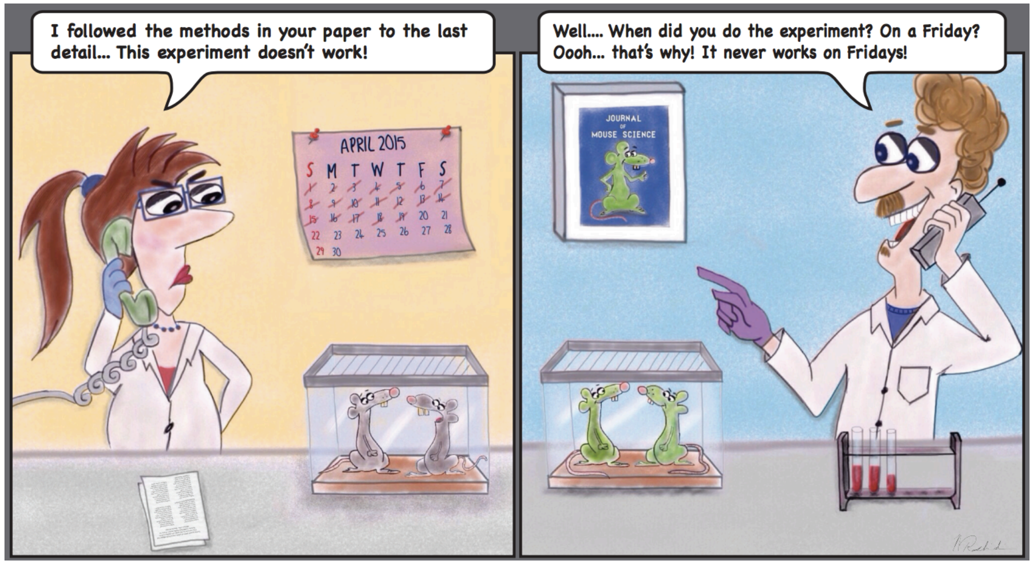


```
ATOM   3378  HB  THR L 227     -36.166  22.580  28.848  1.00  0.00           H
ATOM   3379  HG1 THR L 227     -34.994  19.987  29.136  1.00  0.00           H
ATOM   3380 1HG2 THR L 227     -34.138  22.579  30.246  1.00  0.00           H
ATOM   3381 2HG2 THR L 227     -35.593  22.831  31.238  1.00  0.00           H
ATOM   3382 3HG2 THR L 227     -34.799  21.240  31.213  1.00  0.00           H
TER
# All scores below are weighted scores, not raw scores.
#BEGIN_POSE_ENERGIES_TABLE 3gbn_Ab_0005.pdb
label fa_atr fa_rep fa_sol fa_intra_rep fa_elec pro_close hbond_sr_bb hbond_lr_bb hbond_bb_sc hbond_sc dslf_fa13 rama omega fa_dun p_aa_pp yhh_planarity ref total
weights 1 0.55 0.9375 0.005 0.875 1.25 1.17 1.17 1.17 1.1 1.25 0.25 0.625 0.7 0.4 0.625 1 NA
pose -994.338 137.719 561.027 2.15688 -112.612 19.7634 -12.2069 -79.5391 -24.1449 -23.4441 -1.15166 -7.47192 71.8572 276.633 -29.8673 0.09431 13.9828 -201.541
GLU:NtermProteinFull_1 -1.58225 0.53996 1.45283 0.00353 0.06909 0 0 0 0 0 0 0 0.01109 6.53174 0 0 -1.96094 5.06505
VAL_2 -3.34255 0.45648 1.46378 0.01322 -0.08167 0 0 0 0 0 0 -0.16095 0.87346 0.30715 0.39992 0 0.97964 0.90848
GLN_3 -2.79445 0.10936 1.74929 0.00451 -0.52743 0 0 0 -0.35772 0 0 -0.09682 0.35321 2.59775 0.02034 0 -1.51717 -0.45911
LEU_4 -5.13483 0.73792 1.6574 0.00685 -0.16379 0 0 0 0 0 0 0.06265 0.2281 2.29891 -0.1217 0 0.76113 0.33264
VAL_5 -2.72905 0.12167 1.72074 0.00789 -0.45069 0 0 0 0 0 0 -0.27382 0.01969 0.02557 -0.49649 0 0.97964 -1.07485
```

Examples of output: Tracer output (log files)

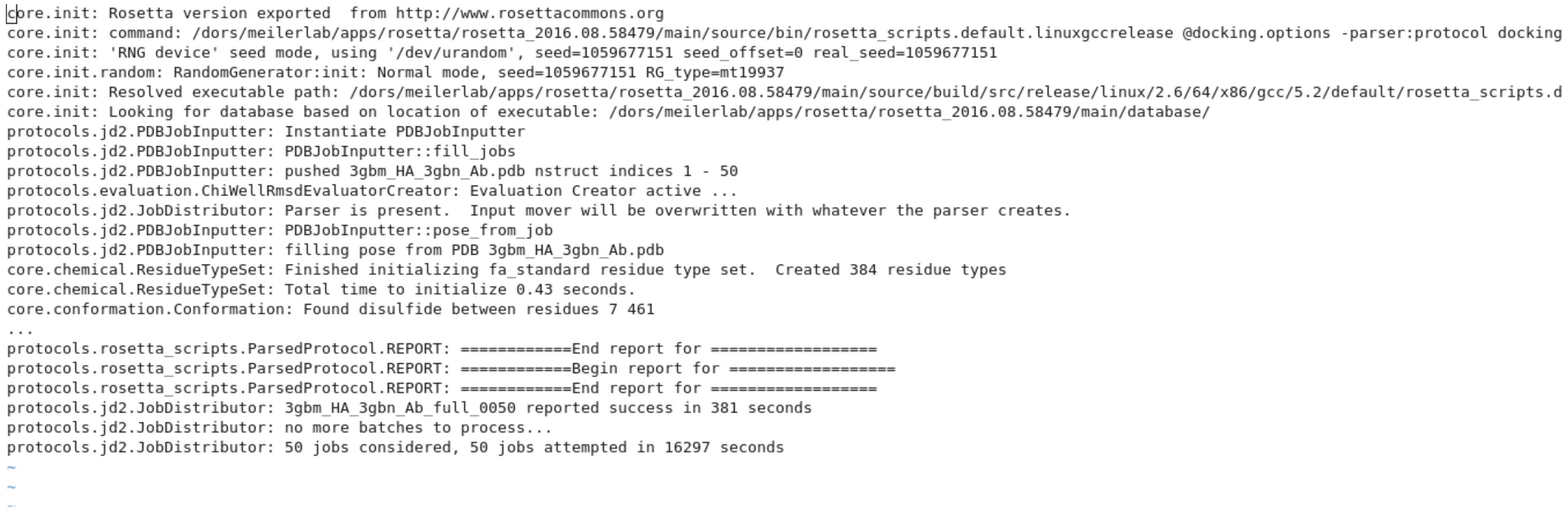
1qys_score.log

- Shows exactly the command line you are running at the beginning
- Which databases are being used, calling protocols, warnings, errors, etc.
- Very useful for debugging to figure out where problems are coming from
- Makes your protocol reproducible!!


I followed the methods in your paper to the last detail... This experiment doesn't work!
APRIL 2015
Well.... When did you do the experiment? On a Friday? Oooh... that's why! It never works on Fridays!
JOURNAL OF MOUSE SCIENCE

Examples of output: Tracer output (log files)

1qys_score.log



```
core.init: Rosetta version exported  from http://www.rosettacommons.org
core.init: command: /dors/meilerlab/apps/rosetta/rosetta_2016.08.58479/main/source/bin/rosetta_scripts.default.linuxgccrelease @docking.options -parser:protocol docking
core.init: 'RNG device' seed mode, using '/dev/urandom', seed=1059677151 seed_offset=0 real_seed=1059677151
core.init.random: RandomGenerator:init: Normal mode, seed=1059677151 RG_type=mt19937
core.init: Resolved executable path: /dors/meilerlab/apps/rosetta/rosetta_2016.08.58479/main/source/build/src/release/linux/2.6/64/x86/gcc/5.2/default/rosetta_scripts.d
core.init: Looking for database based on location of executable: /dors/meilerlab/apps/rosetta/rosetta_2016.08.58479/main/database/
protocols.jd2.PDBJobInputter: Instantiate PDBJobInputter
protocols.jd2.PDBJobInputter: PDBJobInputter::fill_jobs
protocols.jd2.PDBJobInputter: pushed 3gbm_HA_3gbn_Ab.pdb nstruct indices 1 - 50
protocols.evaluation.ChiWellRmsdEvaluatorCreator: Evaluation Creator active ...
protocols.jd2.JobDistributor: Parser is present.  Input mover will be overwritten with whatever the parser creates.
protocols.jd2.PDBJobInputter: PDBJobInputter::pose_from_job
protocols.jd2.PDBJobInputter: filling pose from PDB 3gbm_HA_3gbn_Ab.pdb
core.chemical.ResidueTypeSet: Finished initializing fa_standard residue type set.  Created 384 residue types
core.chemical.ResidueTypeSet: Total time to initialize 0.43 seconds.
core.conformation.Conformation: Found disulfide between residues 7 461
...
protocols.rosetta_scripts.ParsedProtocol.REPORT: ============End report for ==================
protocols.rosetta_scripts.ParsedProtocol.REPORT: ============Begin report for ==================
protocols.rosetta_scripts.ParsedProtocol.REPORT: ============End report for ==================
protocols.jd2.JobDistributor: 3gbm_HA_3gbn_Ab_full_0050 reported success in 381 seconds
protocols.jd2.JobDistributor: no more batches to process...
protocols.jd2.JobDistributor: 50 jobs considered, 50 jobs attempted in 16297 seconds
~
~
~
```

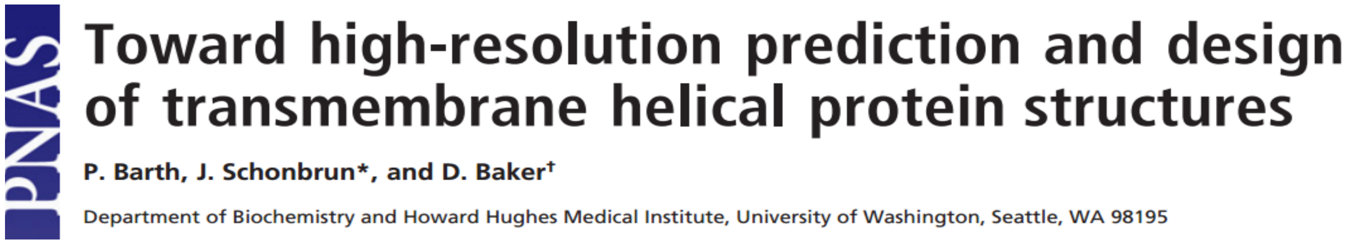
Options to control tracer output *these files can get very long!*:

- Silence certain tracers:
  - -mute core.chemical.ResidueTypeSet
- Change verbosity level (Error/Warning/Info/Debug/Trace)
  - -out:levels all:Warning core.init:Info

Other files: application-specific

- Span File: Defines which residues are in the membrane
- Loops File: Identifies the loop residues for loop closure
- Params File: Custom parameters for small molecules or non-canonical amino acids
- Constraint File: Experimentally derived restraints
- Fragment File: Short protein segments used for comparative modeling and de novo folding
- Res Files: Indicates which residue positions should be designed

Protocols can get complicated…


PNAS

Toward high-resolution prediction and design of transmembrane helical protein structures

P. Barth, J. Schonbrun*, and D. Baker†

Department of Biochemistry and Howard Hughes Medical Institute, University of Washington, Seattle, WA 98195

```
$ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease -database
../../rosetta_database -in:file:fasta ./input_files/1elwA.fasta -
in:file:native ./input_files/1elw.pdb -in:file:frag3
./input_files/aa1elwA03_05.200_v1_3 -in:file:frag9
./input_files/aa1elwA09_05.200_v1_3 -abinitio:relax -relax:fast -
abinitio::increase_cycles 10 -abinitio::rg_reweight 0.5 -
abinitio::rsd_wt_helix 0.5 -abinitio::rsd_wt_loop 0.5 -use_filters
true -psipred_ss2 ./input_files/1elwA.psipred_ss2 -kill_hairpins -
out:file:silent 1elwA_silent.out
-nstruct 10
```

OR

```
$ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease @options.txt
```

Use an options file for your runs

Why?

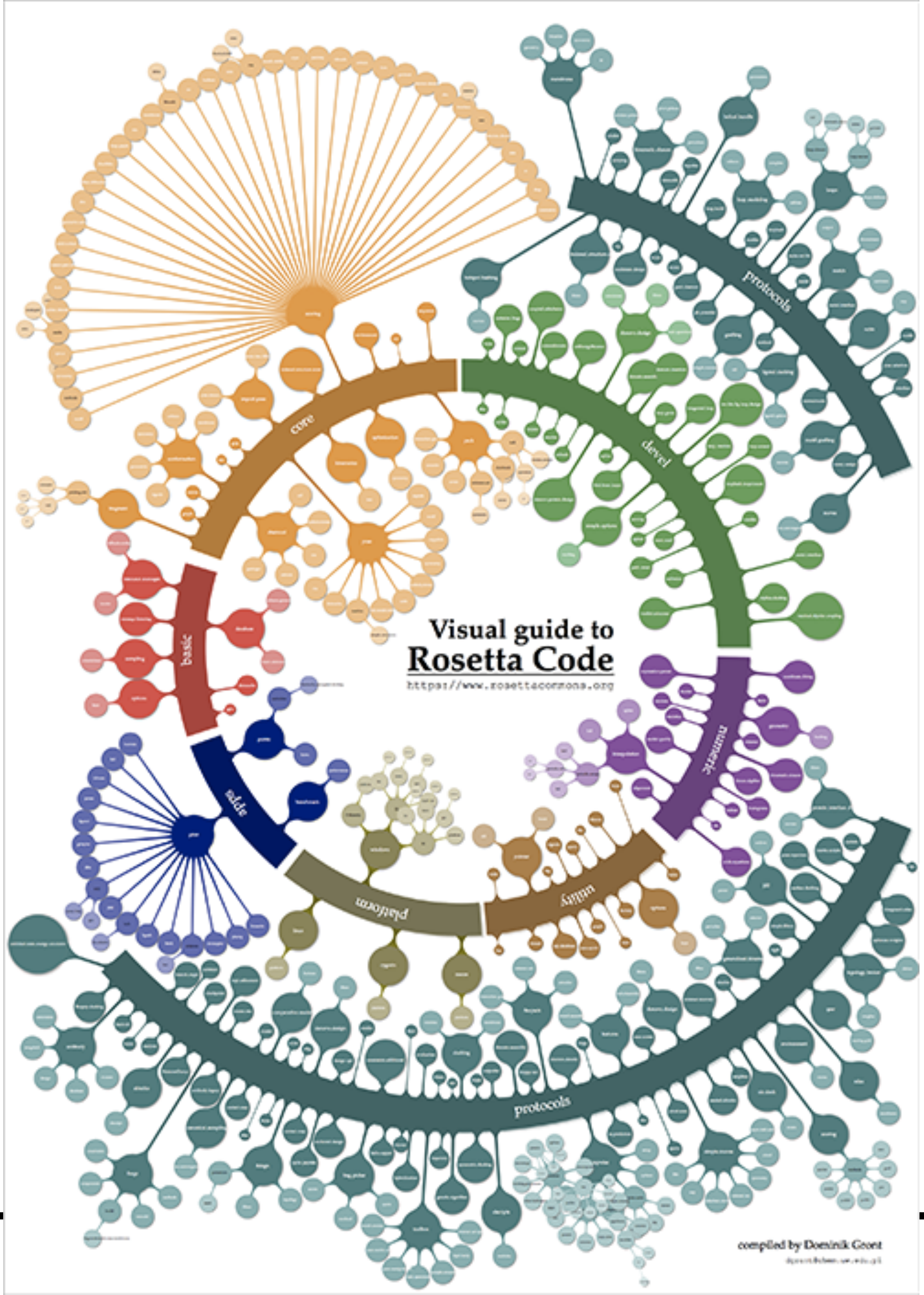
- Easier to read/organize
- Reproducibility!

```
$ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease @options.txt
```

```
-in:file
      -fasta ./input_files/1e1wA.fasta
      -native ./input_files/1e1w.pdb
      -frag3 ./input_files/aa1elwA03_05.200_v1_3
      -frag9 ./input_files/aa1elwA09_05.200_v1_3
-psipred_ss2 ./input_files/1elwA.psipred_ss2
-abinitio:relax
-relax:fast
-abinitio::increase_cycles 10
-abinitio::rg_reweight 0.5
…
-out:file:silent ./output_files/1elwA_silent.out
-nstruct 10
```

Any Questions?

WITH GREAT POWER
COMES GREAT RESPONSIBILITY
makeameme.org


Visual guide to
Rosetta Code
https://www.rosettacommons.org
core
devel
protocols
basic
numeric
platform
utility
protocols
compiled by Dominik Gront

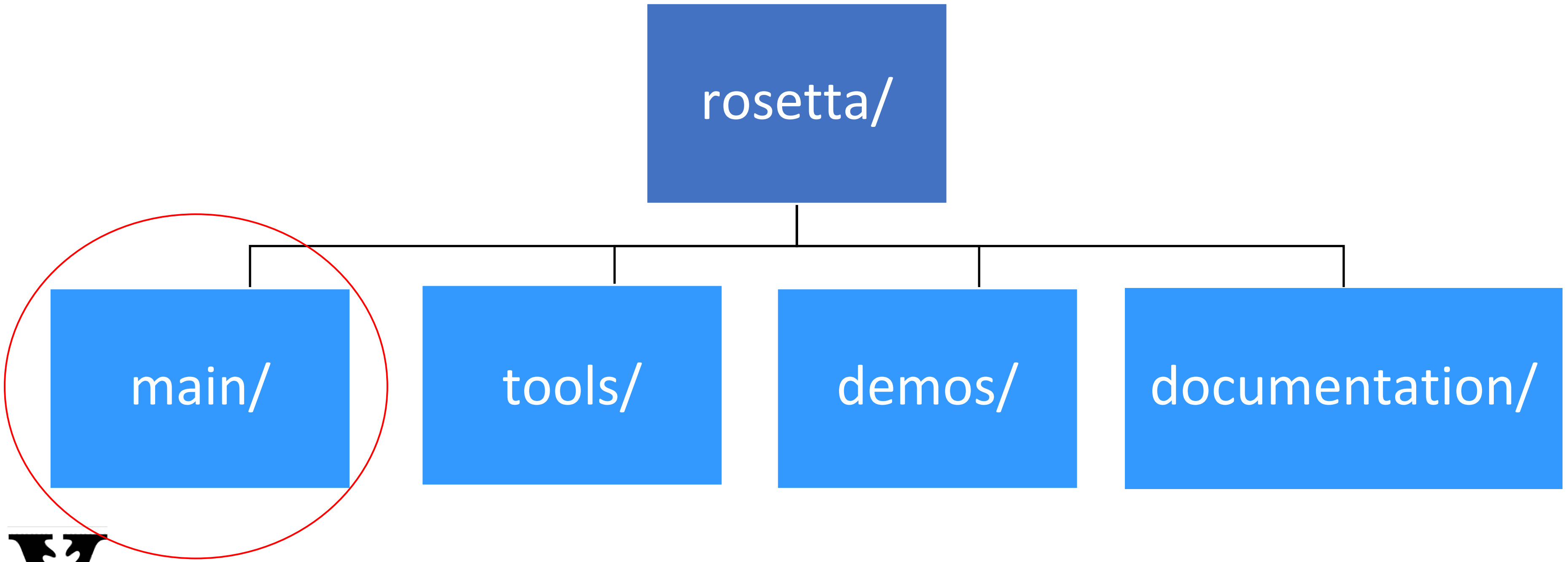
Rosetta Resources for Users:

https://www.rosettacommons.org

- Documentation
- User guides
- Forum
- Software Download
- Tutorials (meilerlab.org)

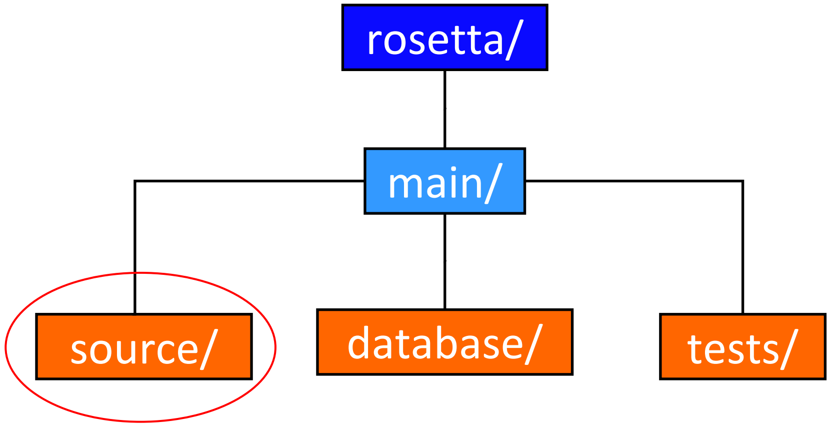
Basic Rosetta Structure

```
cd ~/rosetta_workshop/rosetta/
```



~/rosetta_workshop/rosetta/main/

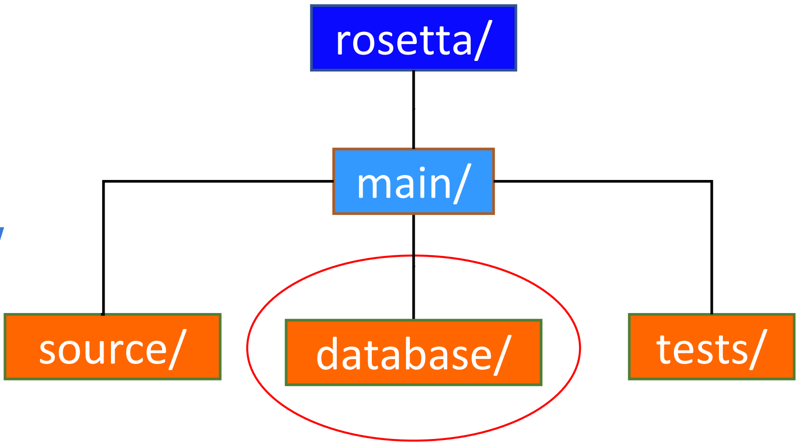
- Rosetta/main/source/bin/
  - Most applications you will run are
  - calling programs within the bin directory
    - rosetta_scripts.default.linuxgccrelease
    - score_jd2.default.linuxgccrelease
    - relax.default.linuxgccrelease
- Rosetta/main/source/scons.py
  - Used for compiling
- Rosetta/main/source/src/
  - This is where all of the code lives
- Rosetta/main/source/scripts/
  - Some useful scripts live here
  - (e.g. params file generation)



~/rosetta_workshop/rosetta/main/database/

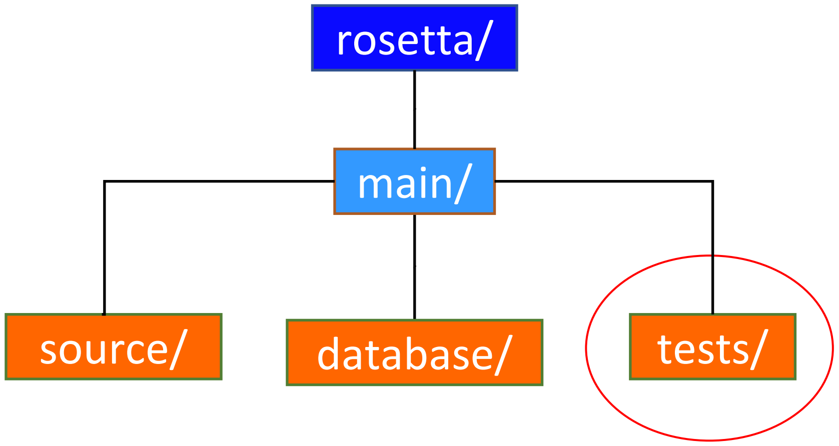
Contains pre-defined information that an application needs — users generally don’t change things here.
Note: Most of the time, applications know where the database is without having to specify it.

- Rosetta/main/database/scoring/
  - Default weights files
  - Rotamer libraries
- Rosetta/main/database/chemical/
  - Residue information--params files
  - Atom sets



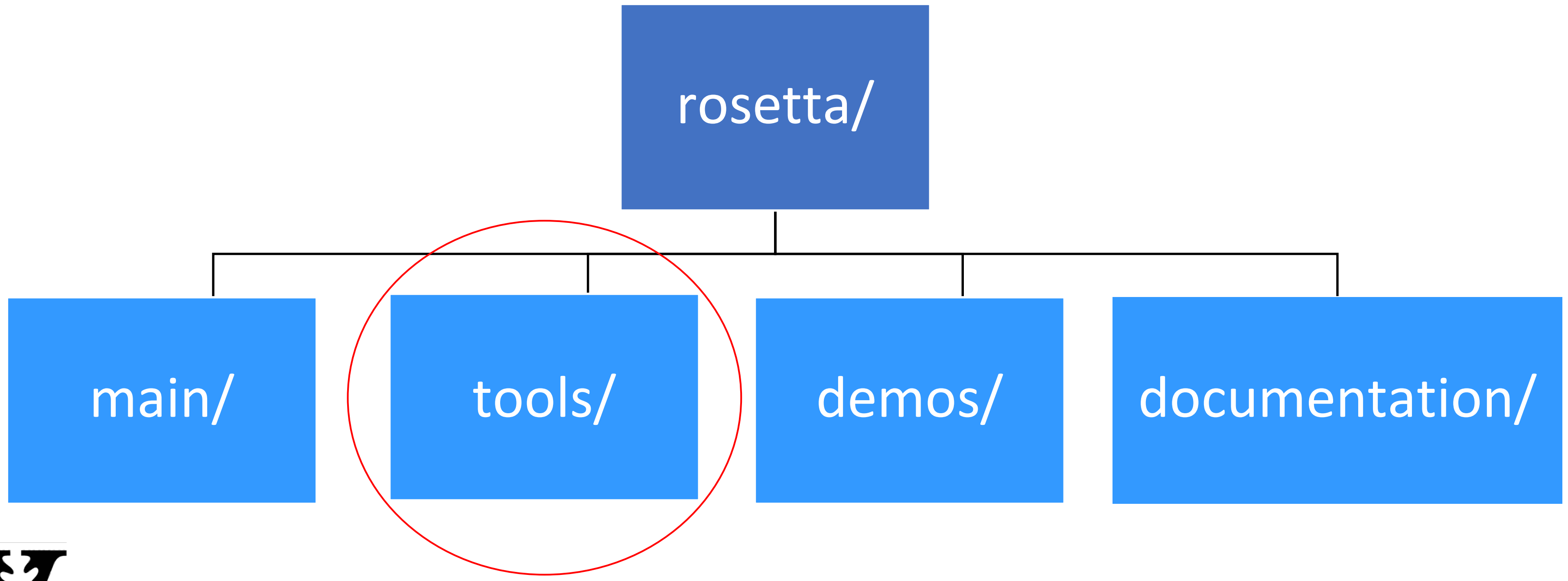
~/rosetta_workshop/rosetta/main/tests/

- Rosetta/main/tests/
  - Tests for Rosetta code (useful for developers only)

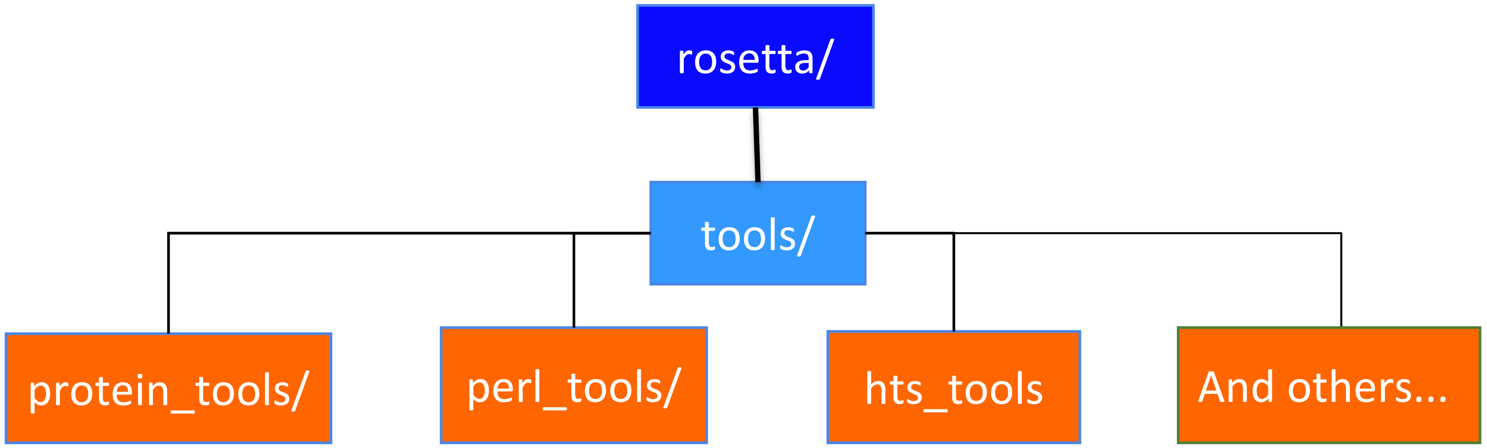


Basic Rosetta Structure

```
cd ~/rosetta_workshop/rosetta/
```

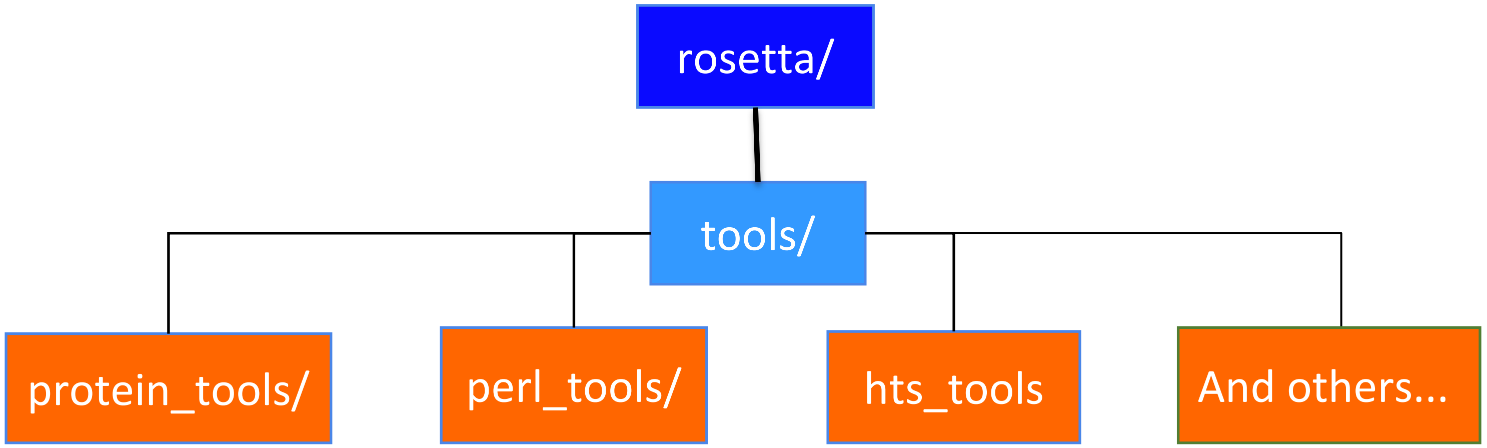


~/rosetta_workshop/rosetta/tools/



- These scripts are incredibly help for smaller, more basic tasks
- Used mainly to setup or analyze runs

~/rosetta_workshop/rosetta/tools/protein_tools/scripts/



Some useful scripts to be aware of:

clean_pdb.py*

- Makes a PDB "Rosetta-proof" and used at the beginning of almost any protocol

pdb_renumber.py

- Renumbering your PDB starting from 1

score_vs_rmsd.py

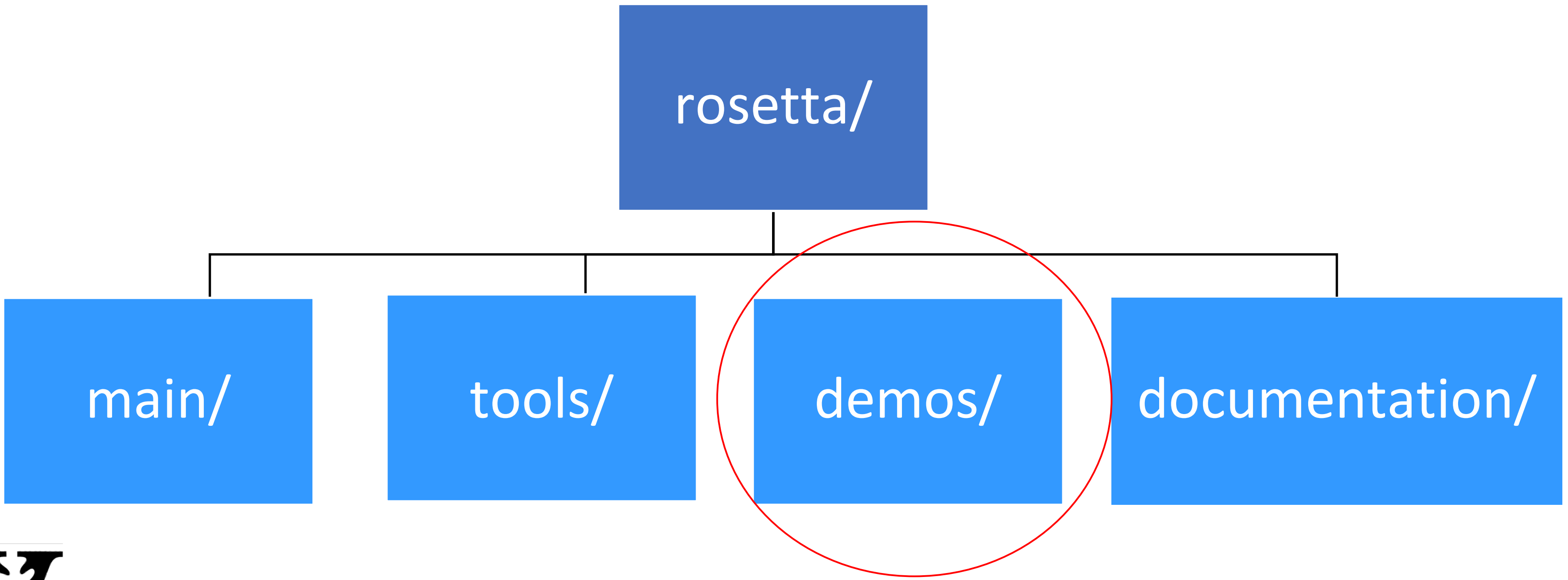
- Setup for score vs. RMSD plots

top_n_percent.py

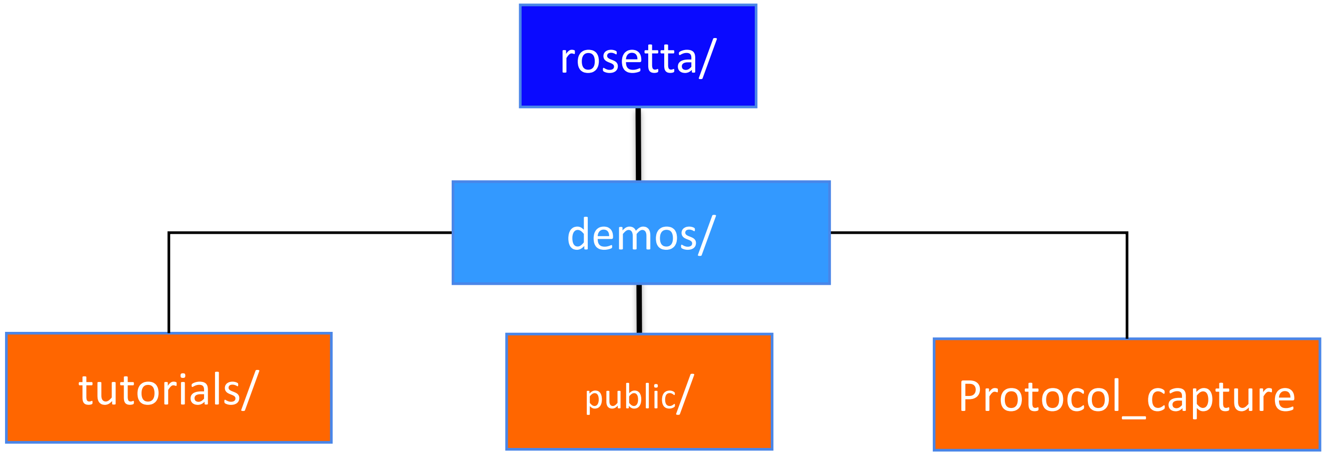
- Extracts tags (protein names) for top given percent of models based on score term

Basic Rosetta Structure

```
cd ~/rosetta_workshop/rosetta/
```



~/rosetta_workshop/rosetta/demos



tutorials/
- Past tutorials

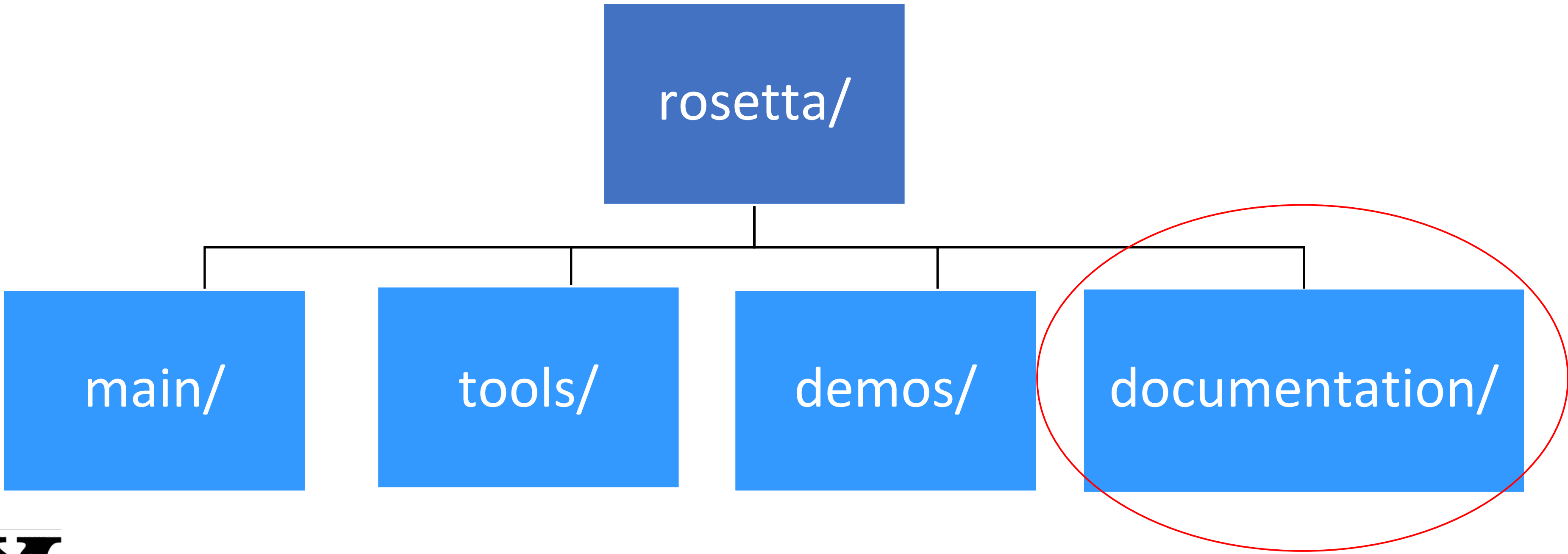
public/
- Protocol examples

protocol_capture/
- protocols associated with a publication

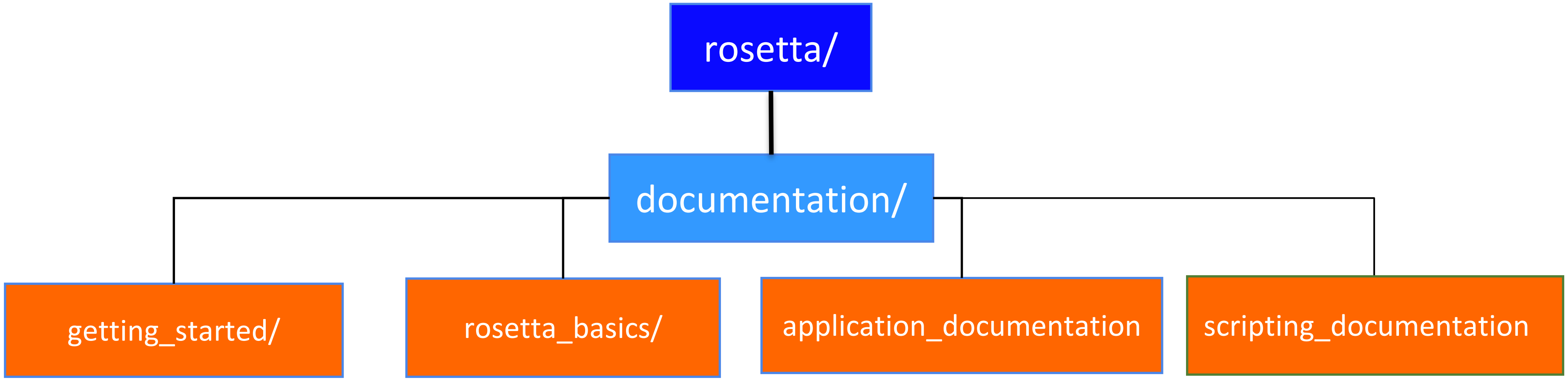
DISCLAIMER: May be out of date, always check Wiki/RosettaCommons/forum for latest information!

Basic Rosetta Structure

```
cd ~/rosetta_workshop/rosetta/
```



~/rosetta_workshop/rosetta/documentation/



Very useful to go through when you're just getting started in Rosetta or any structural biology software

Understanding general Rosetta concepts

- Where to find FAQs (How long does this run take?)
- Options list, file types

General structural biology FAQs

- How do I do X?

Protocols you can use

https://www.rosettacommons.org/docs/latest/Home