

Introduction to the Matcher

Rocco Moretti

(rmorettiase@gmail.com)

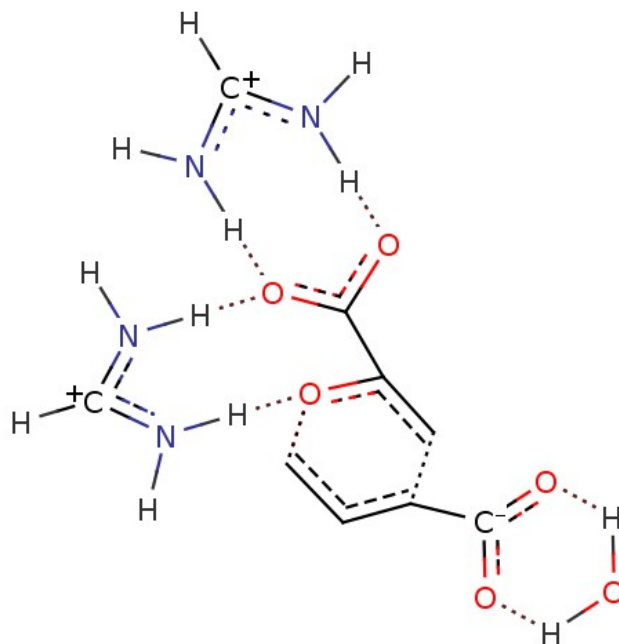
November, 2013

The Matcher

Given a set of geometric constraints for a ligand,
find a scaffold and sequence positions on that scaffold
which satisfy (“match”) those constraints.

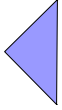
The Matcher

Given a set of geometric constraints for a ligand, find a scaffold and sequence positions on that scaffold which satisfy (“match”) those constraints.



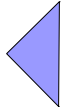
Theozyme

Matching Overview

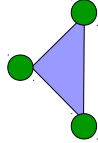


Ligand

Matching Overview

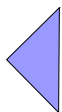


Ligand

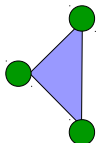


Interactions
(Geometric
Constraints)

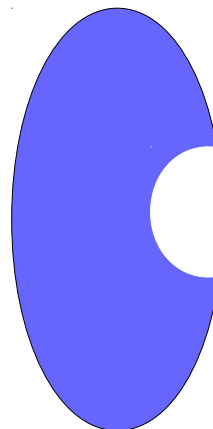
Matching Overview



Ligand

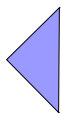


Interactions
(Geometric
Constraints)

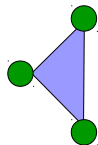


Scaffold

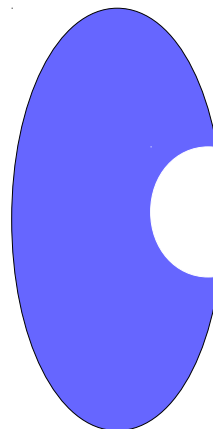
Matching Overview



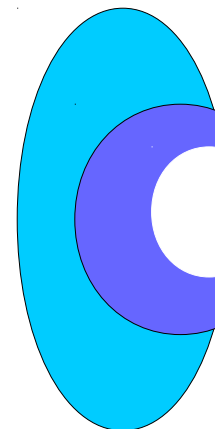
Ligand



Interactions
(Geometric
Constraints)

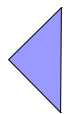


Scaffold

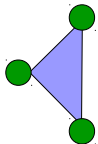


Positions
to Match

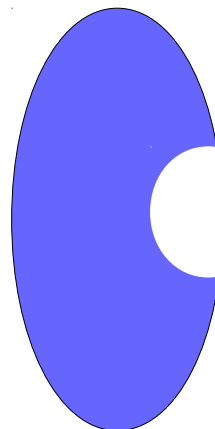
Matching Overview



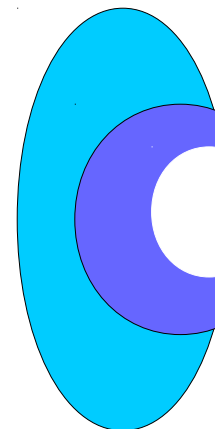
Ligand



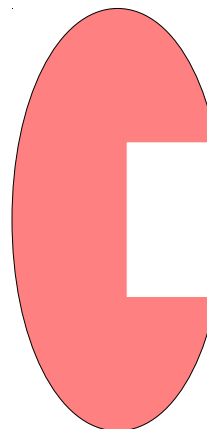
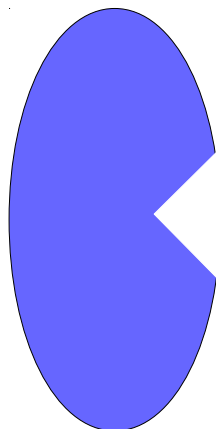
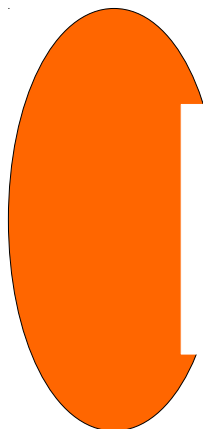
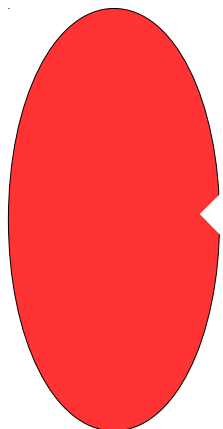
Interactions
(Geometric
Constraints)



Scaffold

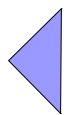


Positions
to Match

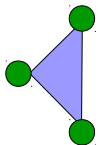


Scaffold Library

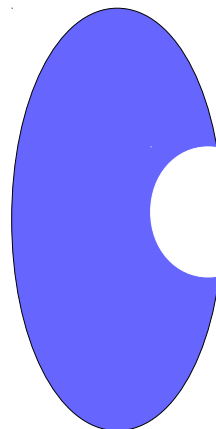
Matching Overview



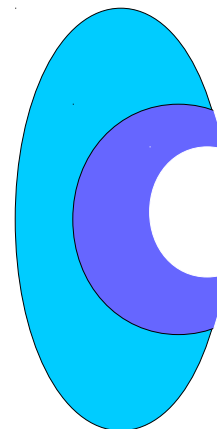
Ligand



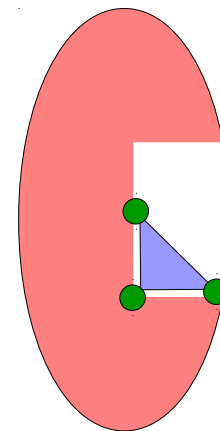
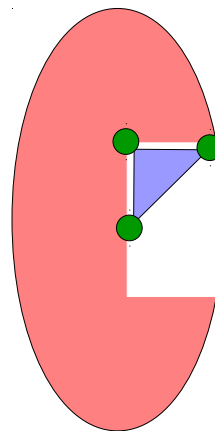
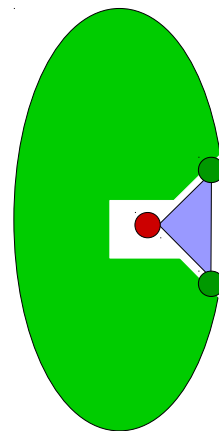
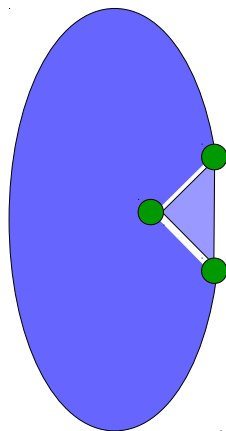
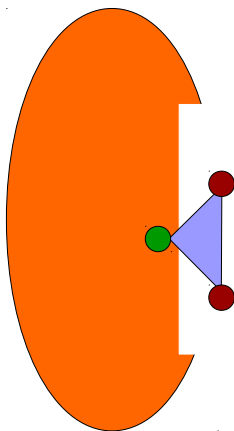
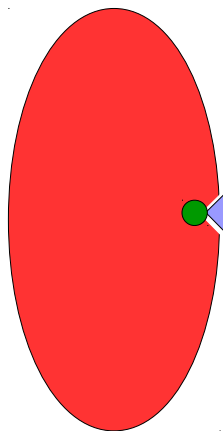
Interactions
(Geometric
Constraints)



Scaffold



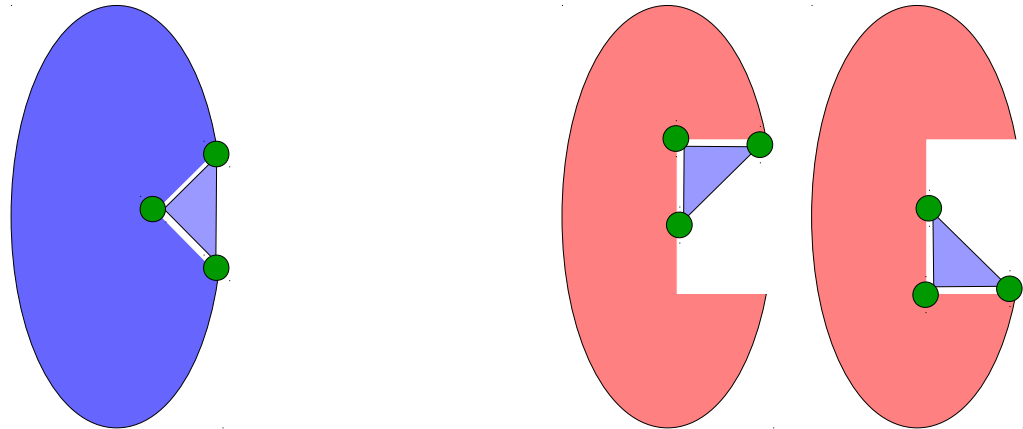
Positions
to Match



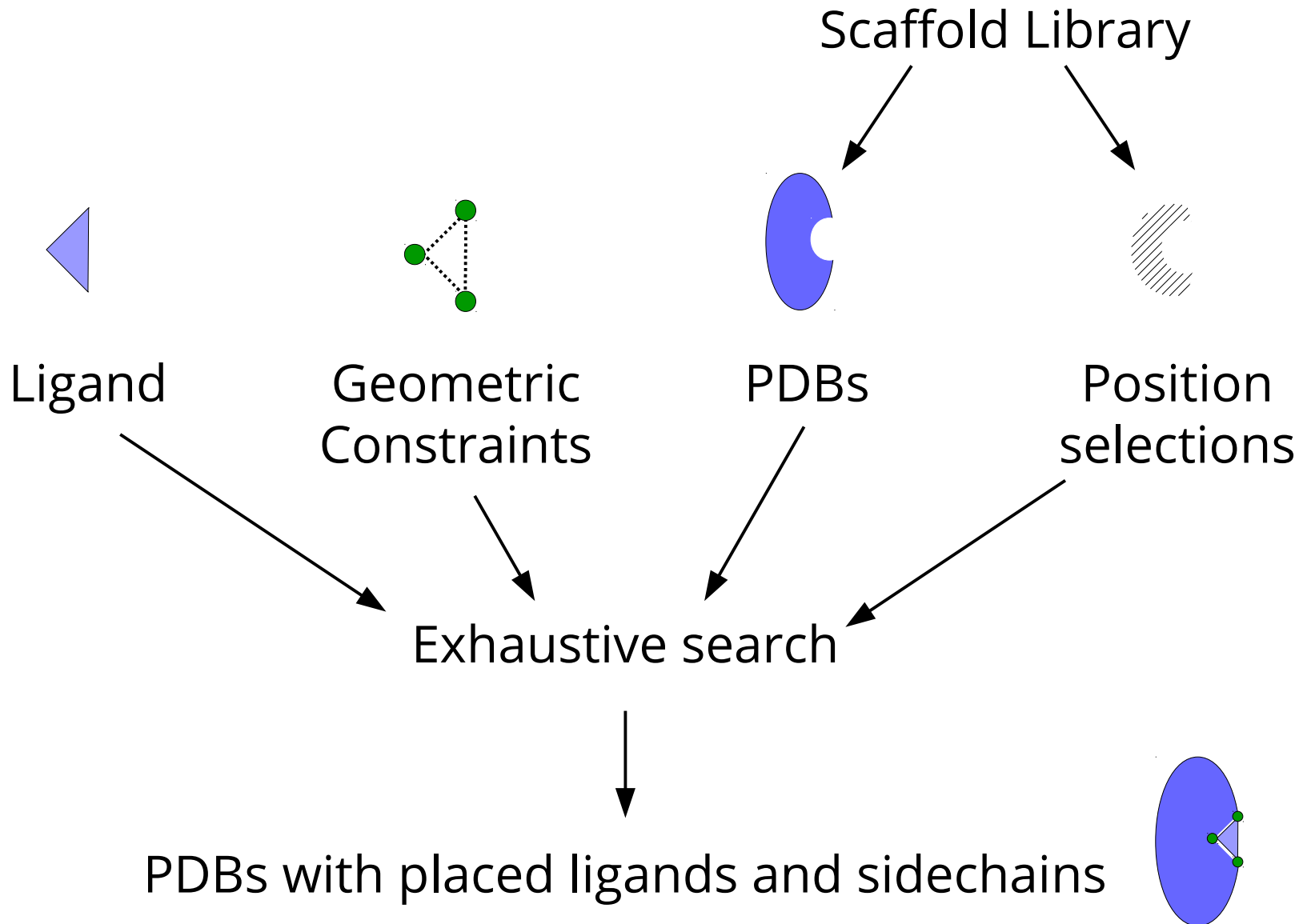
Scaffold Library

Matching Overview

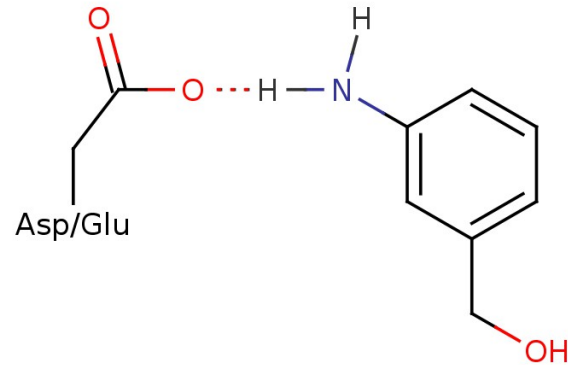
Matcher Output



Overview



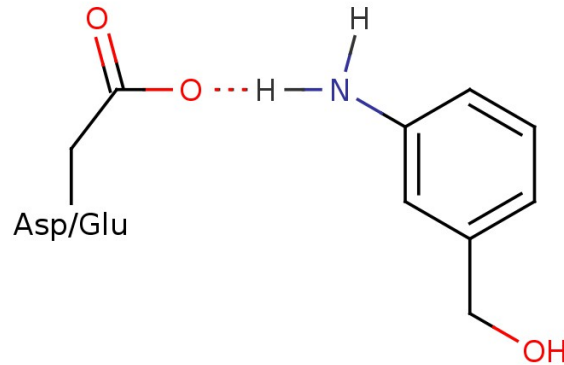
Nomenclature



Upstream
= *Protein*

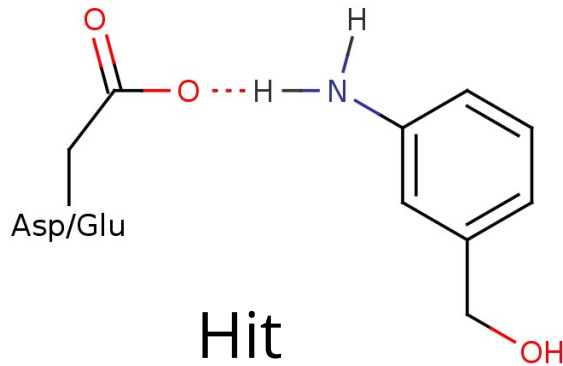
Downstream
= *Ligand*

Nomenclature

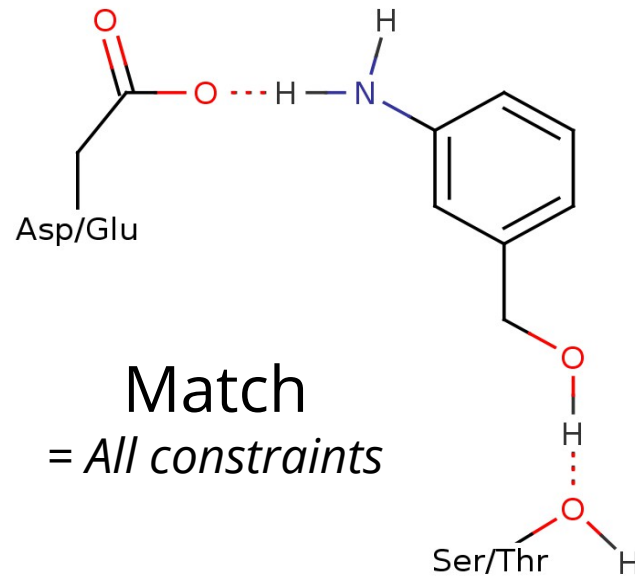


Upstream
= *Protein*

Downstream
= *Ligand*

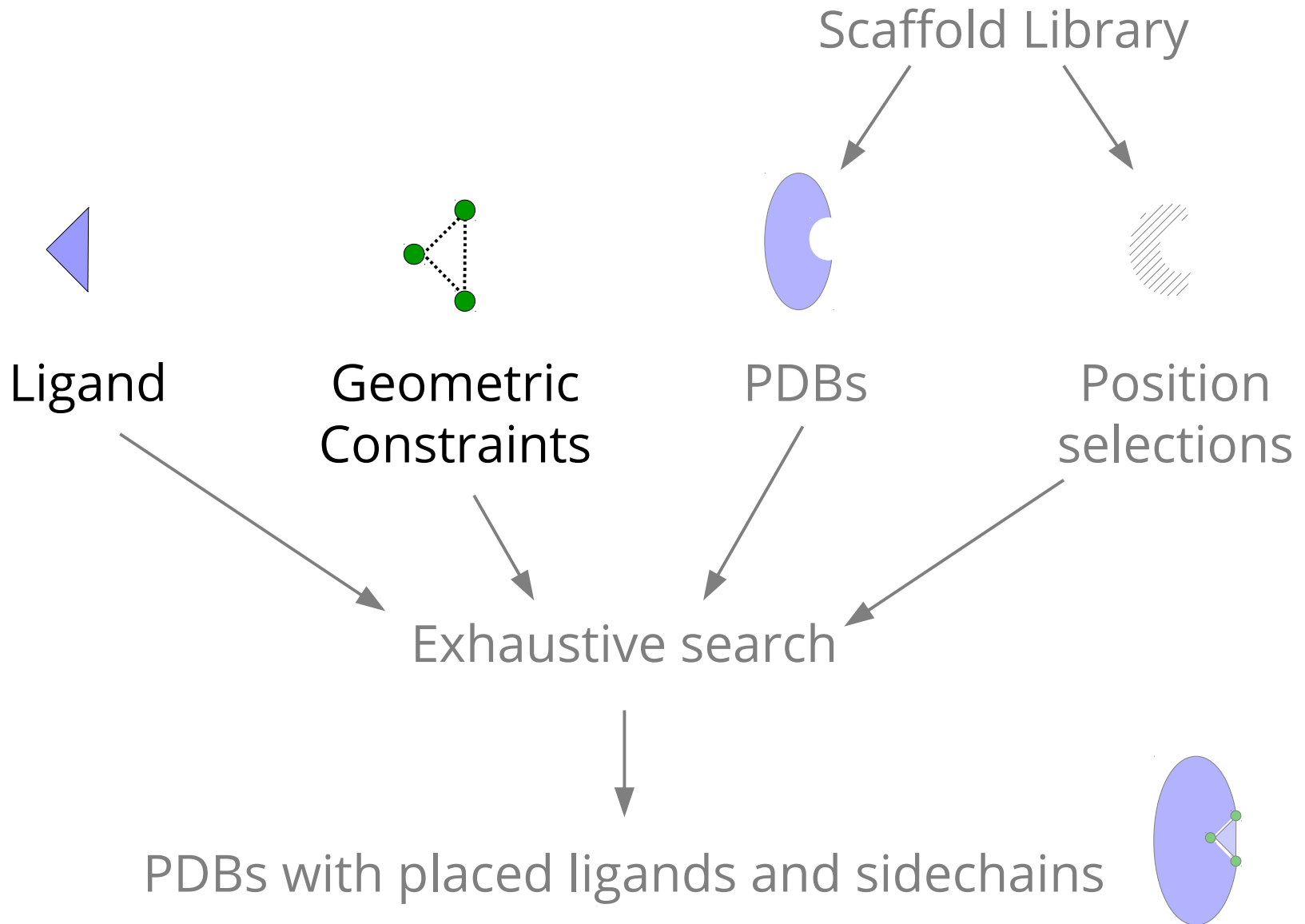


Hit
= *One constraint*



Match
= *All constraints*

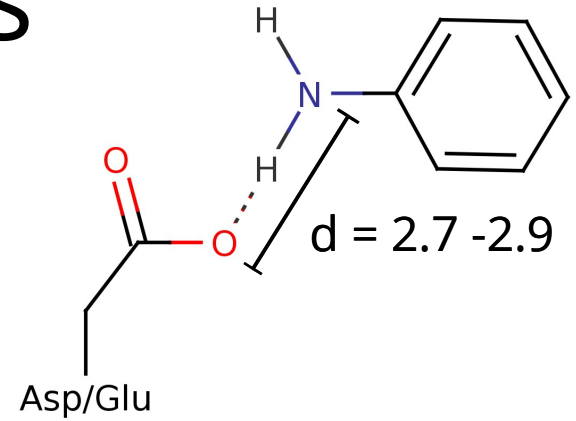
Overview



Constraints

Six degrees of freedom:

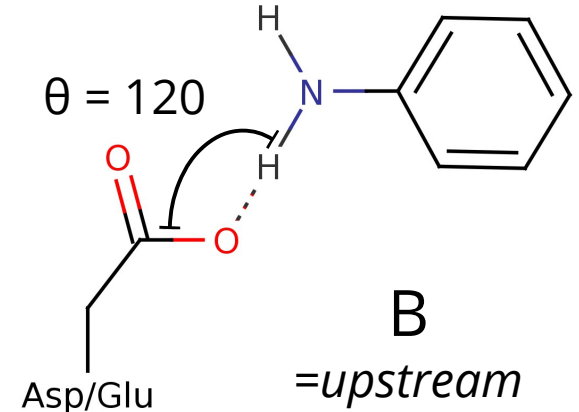
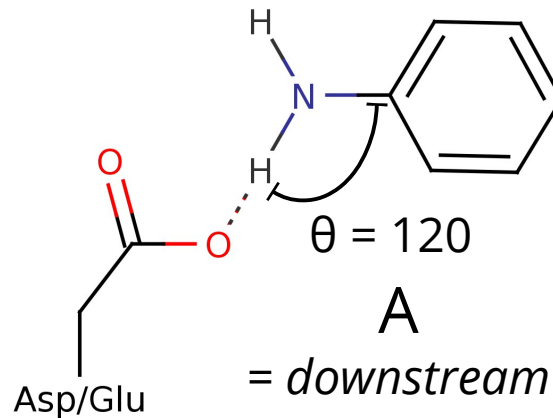
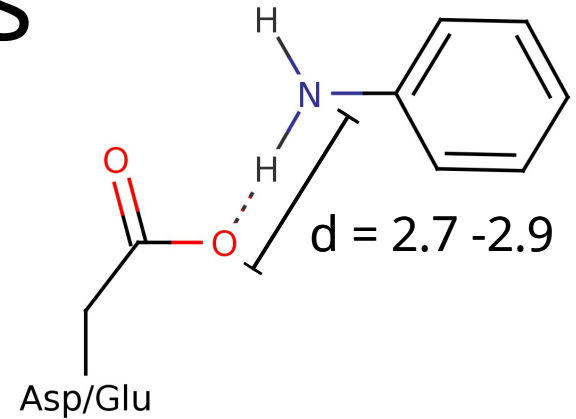
- Distance
- 2 angles
- 3 dihedrals



Constraints

Six degrees of freedom:

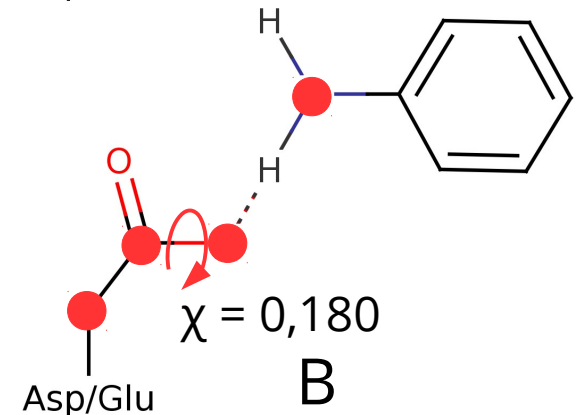
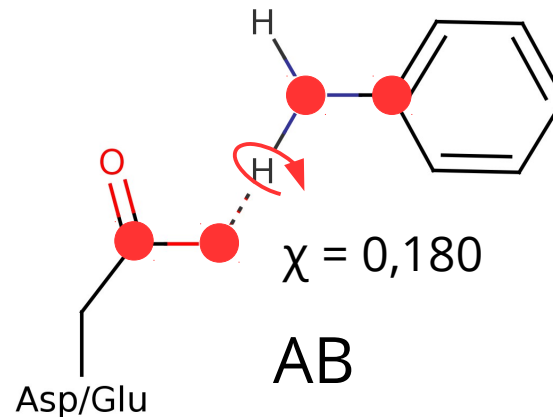
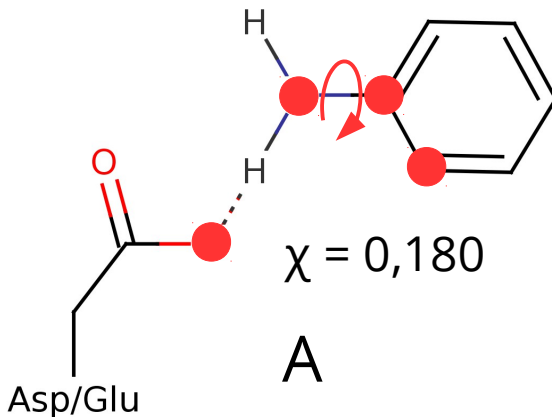
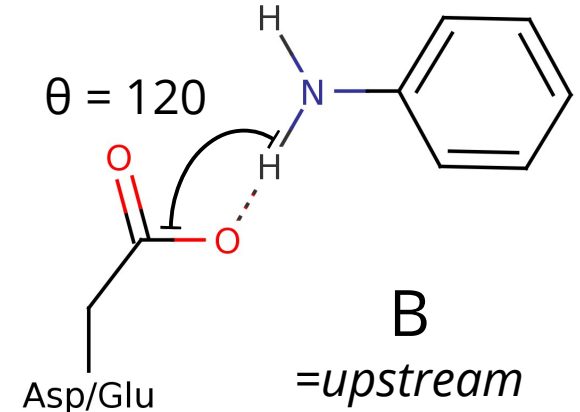
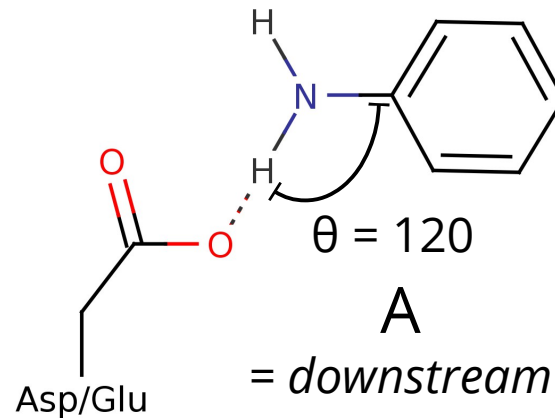
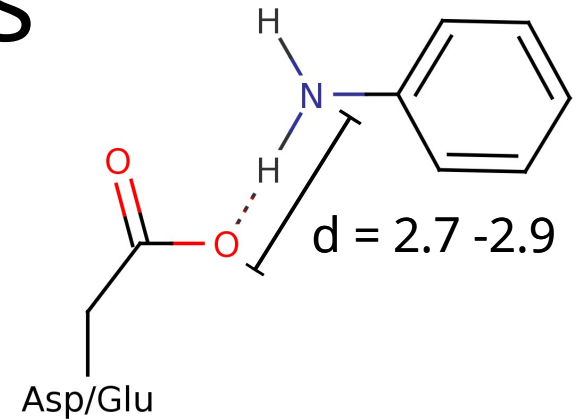
- Distance
- 2 angles
- 3 dihedrals



Constraints

Six degrees of freedom:

- Distance
- 2 angles
- 3 dihedrals



Constraint File Format

Constraint 1

CST::BEGIN

TEMPLATE:: ATOM_MAP: 1 atom_name: N1 C2 C3
TEMPLATE:: ATOM_MAP: 1 residue3: 3MA
TEMPLATE:: ATOM_MAP: 2 atom_type: OOC,
TEMPLATE:: ATOM_MAP: 2 residue1: ED

Atom
Block

CONSTRAINT:: distanceAB: 2.8 0.10 100. 0 0
CONSTRAINT:: angle_A: 120 5.0 25.0 360. 2
CONSTRAINT:: angle_B: 120 5.0 25.0 360. 2
CONSTRAINT:: torsion_A: 0 5.0 10.0 180. 1
CONSTRAINT:: torsion_AB: 0 5.0 10.0 180. 1
CONSTRAINT:: torsion_B: 0 5.0 10.0 180. 1

Geometry
Block

CST::END

Constraint 2

CST::BEGIN

TEMPLATE:: ATOM_MAP: 1 atom_type: OOC
TEMPLATE:: ATOM_MAP: 1 residue1: ED
TEMPLATE:: ATOM_MAP: 2 atom_type: OH
TEMPLATE:: ATOM_MAP: 2 residue1: ST

CONSTRAINT:: distanceAB: 2.8 0.1 100. 0 0
CONSTRAINT:: angle_A: 120. 10.0 60.0 360. 2
CONSTRAINT:: angle_B: 120. 10.0 60.0 360. 2

ALGORITHM_INFO:: match
SECONDARY_MATCH: UPSTREAM_CST 1
ALGORITHM_INFO::END

CST::END

Algorithm
Block

Constraint File – Atoms

PDB atom names
(Distance, angle, dihedral)

```
TEMPLATE::    ATOM_MAP: 1 atom_name: N1 C2 C3
```

PDB code for residue

```
TEMPLATE::    ATOM_MAP: 1 residue3: 3MA
```

Rosetta atom type

```
TEMPLATE::    ATOM_MAP: 2 atom_type: OOC,
```

1 letter amino acid code

```
TEMPLATE::    ATOM_MAP: 2 residue1: ED
```

Constraint File - Geometry

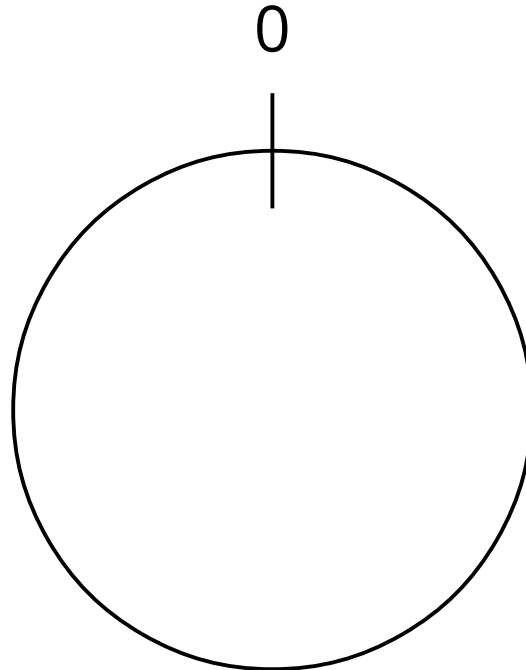
	<i>Value</i>	<i>Sampling range</i>	<i>Weight (ignored)</i>	<i>Periodicity</i>	<i>Sampling level</i>
CONSTRAINT:: distanceAB:	2.8	0.10	100.	0	0
CONSTRAINT:: angle_A:	120	5.0	25.0	360.	2
CONSTRAINT:: angle_B:	120	5.0	25.0	360.	2
CONSTRAINT:: torsion_A:	0	5.0	10.0	180.	1
CONSTRAINT:: torsion_AB:	0	5.0	10.0	180.	1
CONSTRAINT:: torsion_B:	0	5.0	10.0	180.	1

A = downstream

B = upstream

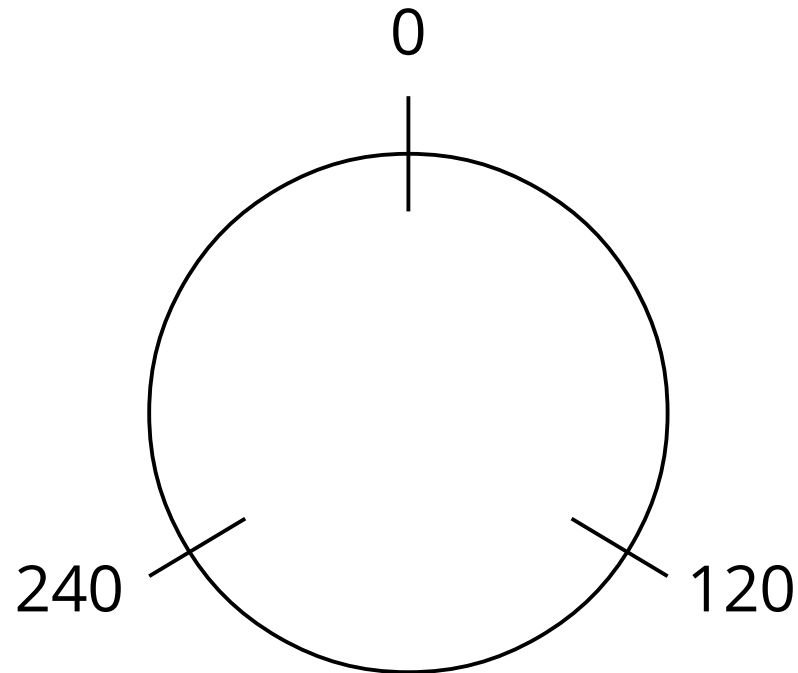
Sampling

CONSTRAINT:: torsion_A: 0 10.0 50.0 120. 1



Sampling

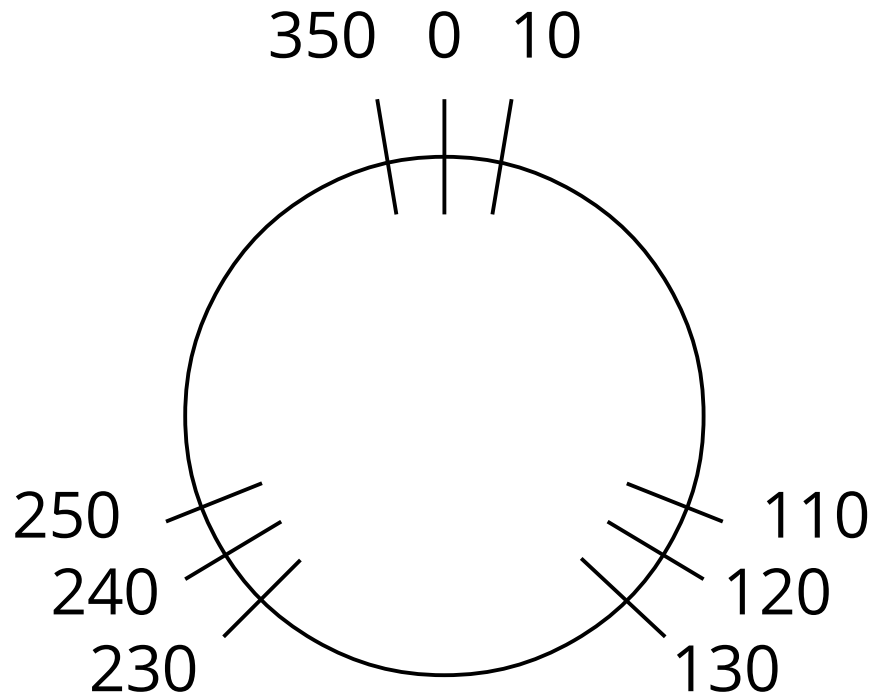
```
CONSTRAINT::  torsion_A:      0   10.0   50.0  120.  1
```



Sampling

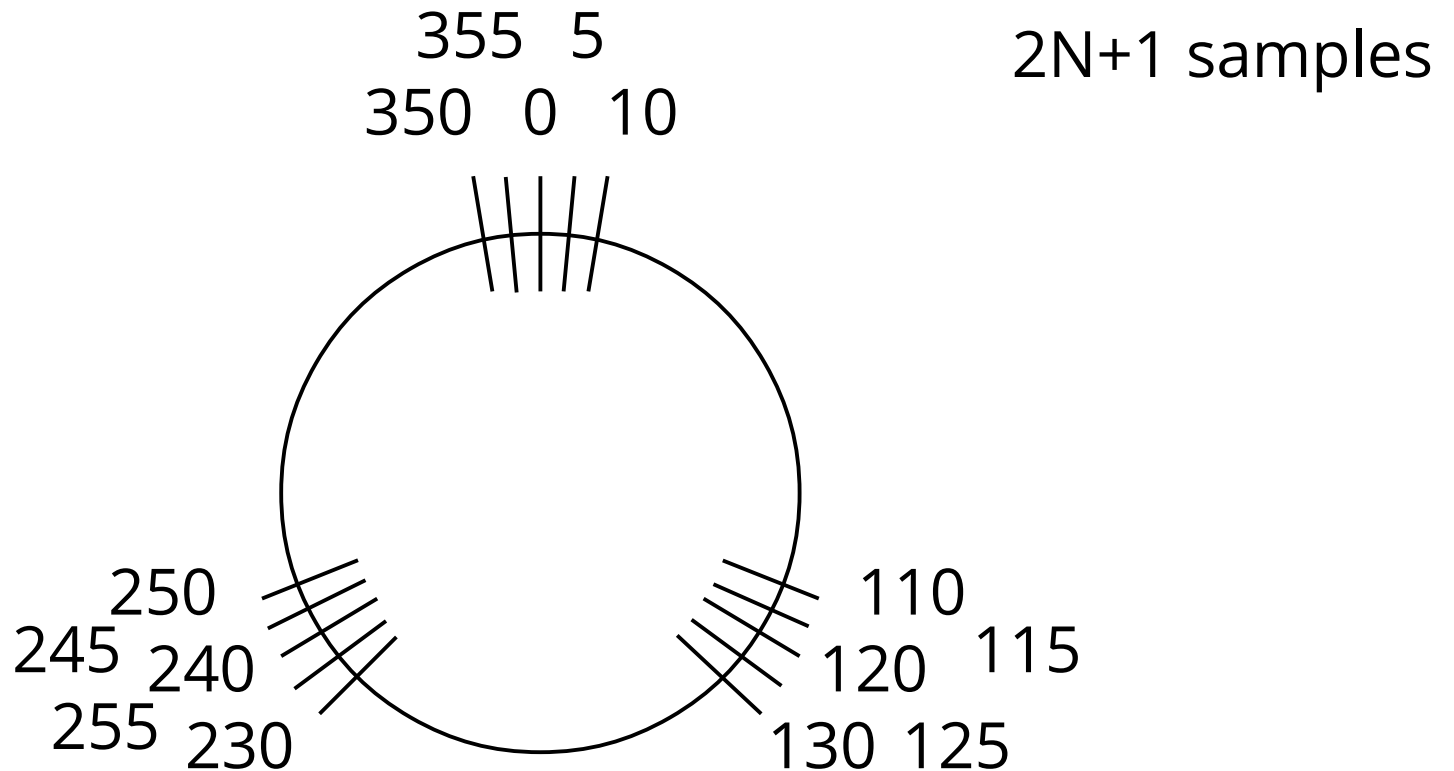
CONSTRAINT:: torsion_A: 0 10.0 50.0 120. 1

$2N+1$ samples



Sampling

CONSTRAINT:: torsion_A: 0 10.0 50.0 120. 2



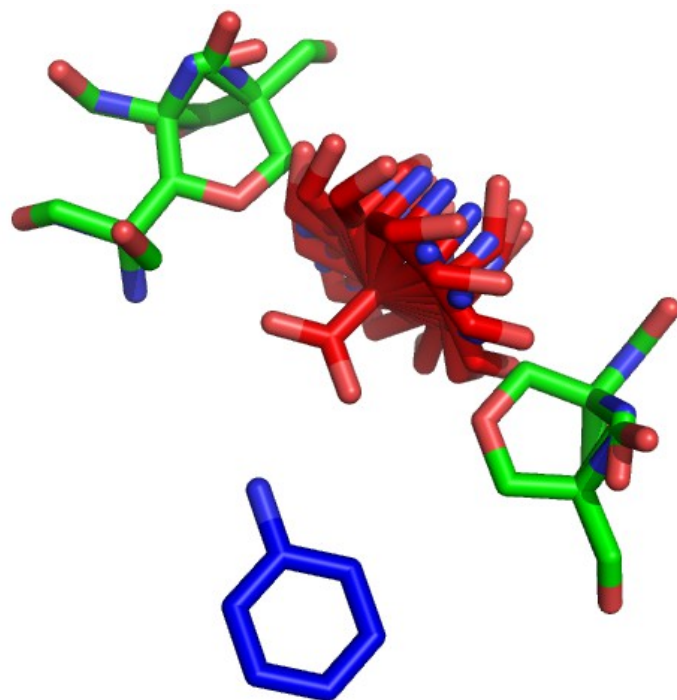
ConstraintFileToTheozyme

CstfileToTheozymePDB.linuxgccrelease

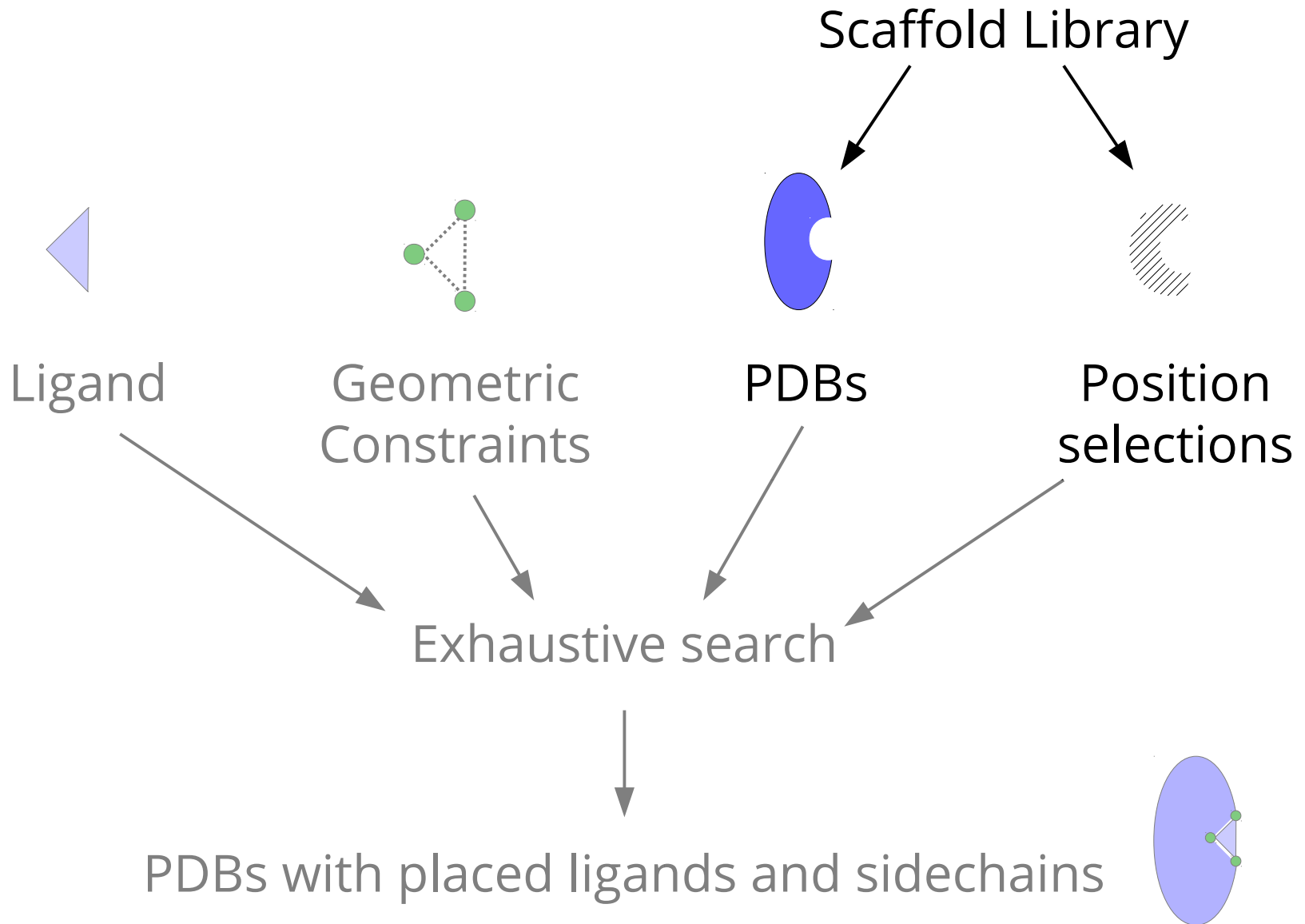
-database~/Rosetta/main/database

-extra_res_faligand.params

-match:geometric_constraint_filereduced.cst



Overview



Scaffold

Typically run against a set of proteins:

- Several dozen select structures (e.g. homologs)
- Several hundred privileged scaffolds (e.g. thermostable)
- Several thousand proteins (e.g. all ligand binders)
- The complete PDB (if you have the computational power)

Pocket selection:

- Limit position to match

Two formats:

- General `-match:scaffold_active_site_residues`
Space separated (pose numbered) scaffold positions
- Constraint specific
`-match:scaffold_active_site_residues_for_geomcst`
One line specifying the number of constraints
One line for each constraint

Automatic scaffold preparation

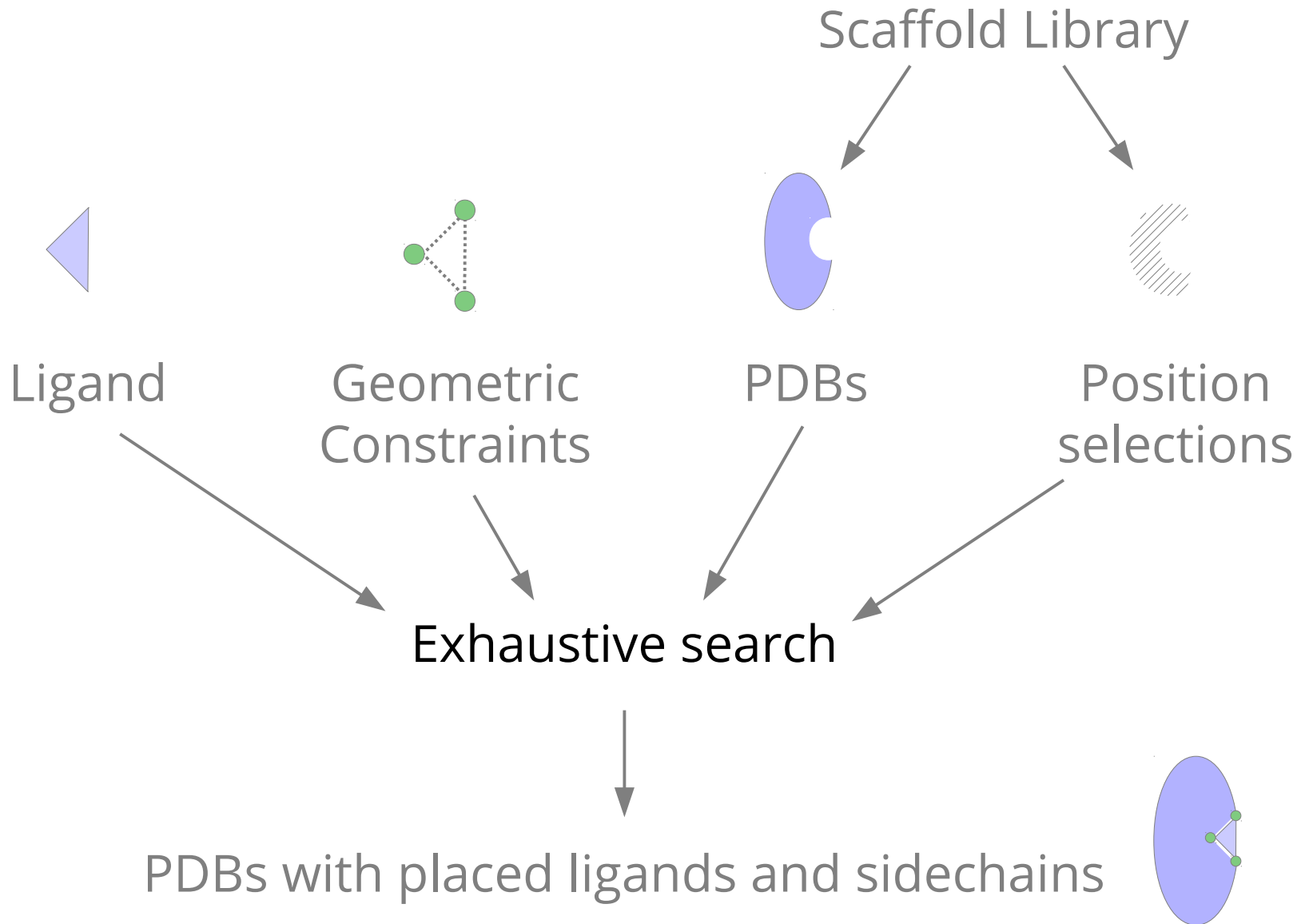
Generate pos files for PDB with existing ligand in pocket:

```
gen_lig_grids.linuxgccrelease -s nohet.pdb ligand.pdb  
-ignore_unrecognized_res -grid_delta 0.5  
-grid_lig_cutoff 4.0 -grid_bb_cutoff 2.25  
-grid_active_res_cutoff 5.0
```

Generate pos files for apo PDBs
(autodetect pocket using RosettaHoles):

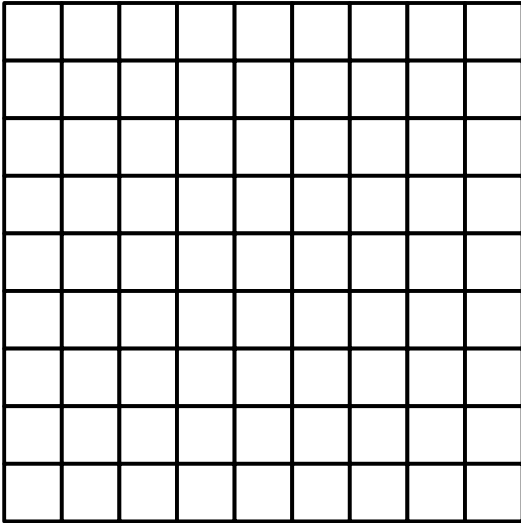
```
gen_apo_grids.linuxgccrelease -s scaffold.pdb  
-ignore_unrecognized_res -packstat:surface_accessibility  
-packstat:cavity_burial_probe_radius 3.0  
-packstat:cluster_min_volume 30  
-packstat:min_cluster_overlap 1.0  
-packstat:min_cav_ball_radius 1.0  
-packstat:min_surface_accessibility 1.4
```

Overview



Hashmap based hit storage

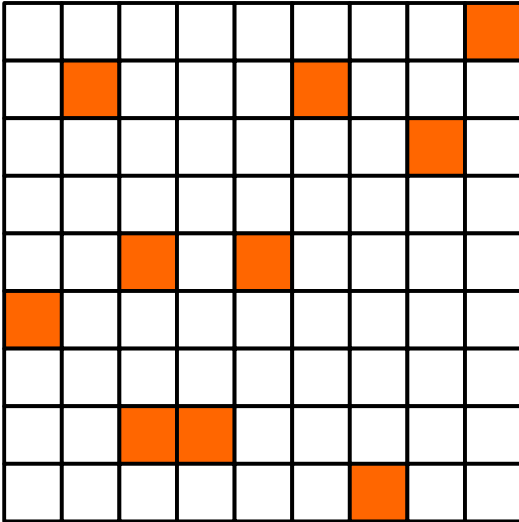
Constraint #1 hits



Position/Orientation
of the ligand

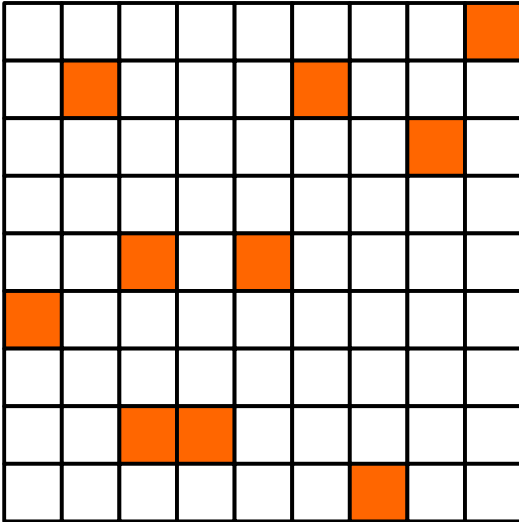
HashMap based hit storage

Constraint #1 hits

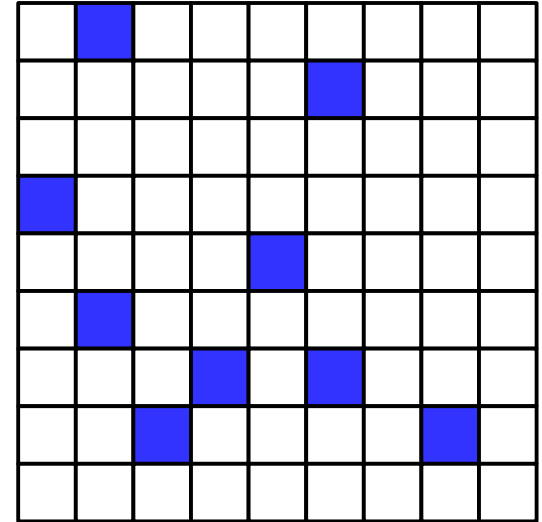


HashMap based hit storage

Constraint #1 hits

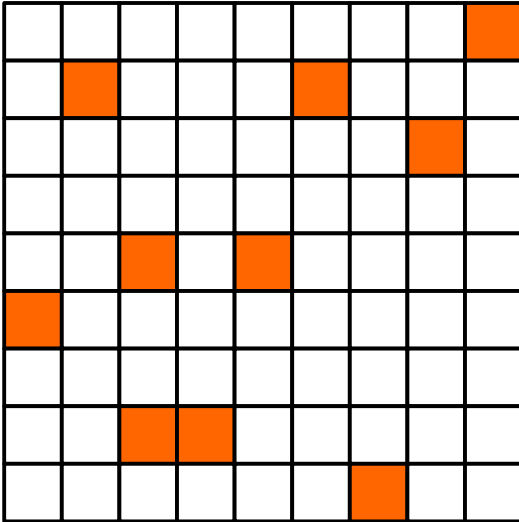


Constraint #2 hits

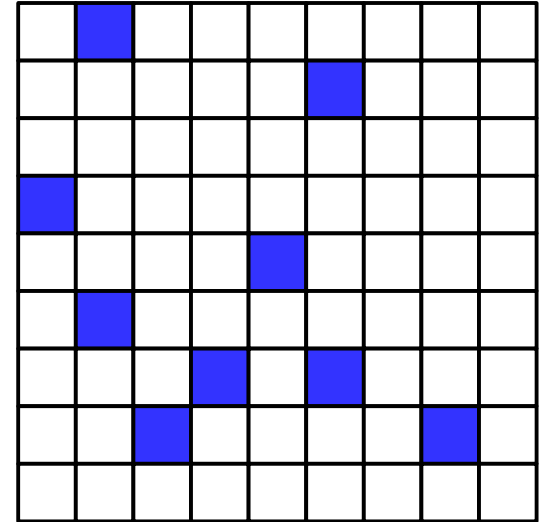


Hashmap based hit storage

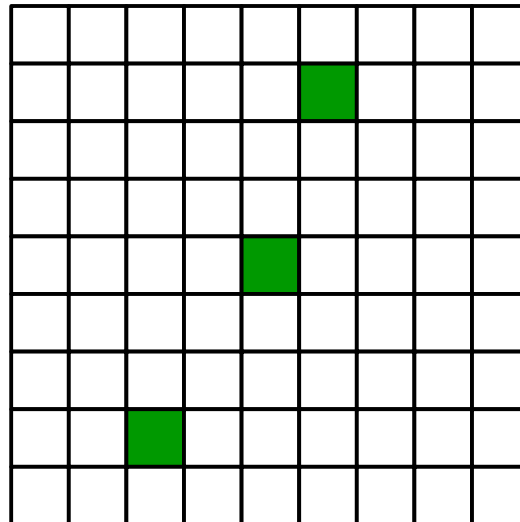
Constraint #1 hits



Constraint #2 hits



Matches

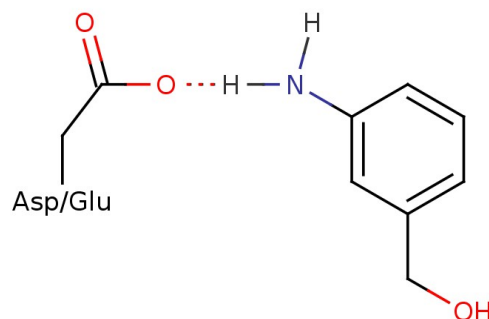


But in 6 dimensions, with identity and rotamer information ...

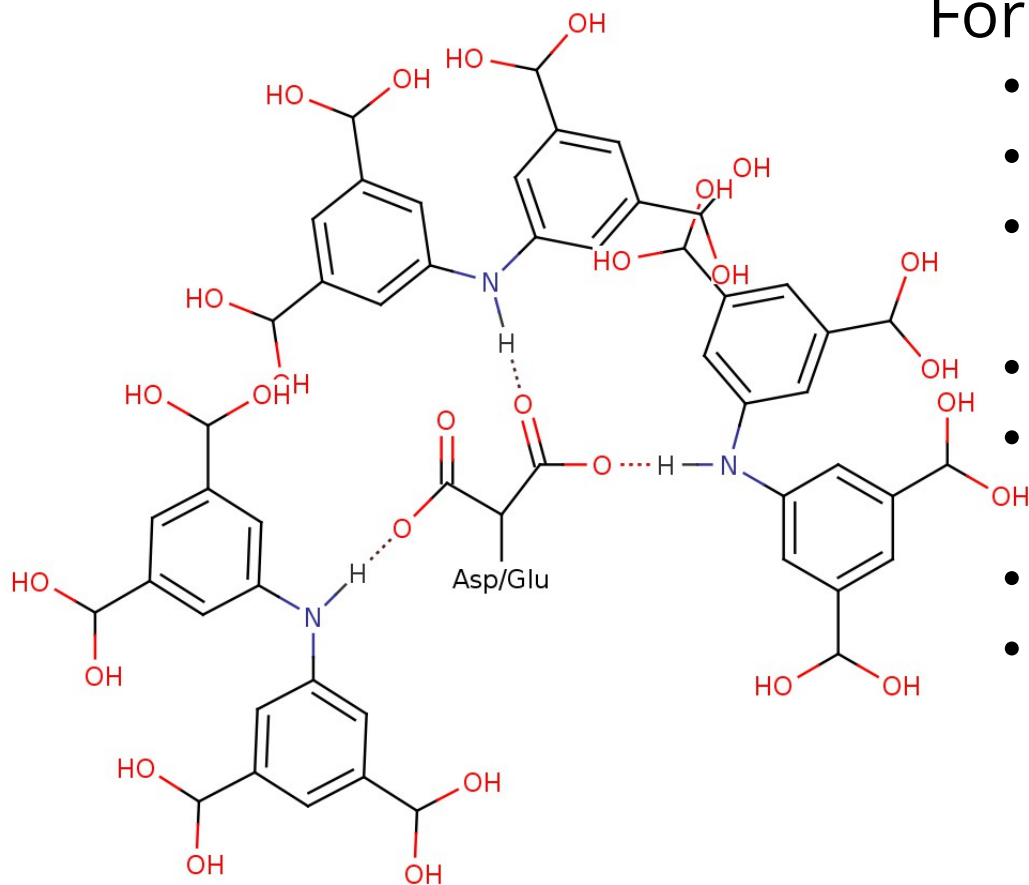
Classic Matching

For each potential position:

- Build protein rotamers
- Build ligand rotamers
- Enumeratively combine pairs with geometry variations
- Filter on backbone clash
- Compute position/orientation signature for ligand
- Store hit in hashmap
- Filter based on hashmap collisions with previous constraints



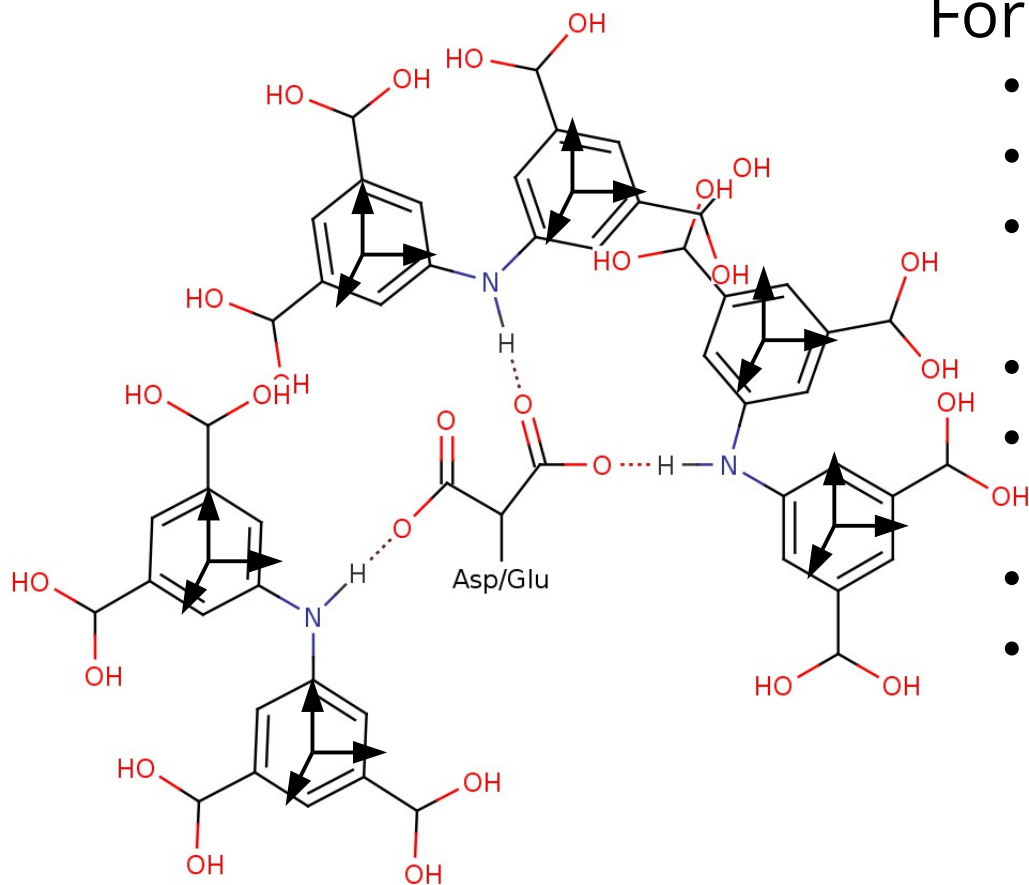
Classic Matching



For each potential position:

- Build protein rotamers
- Build ligand rotamers
- Enumeratively combine pairs with geometry variations
- Filter on backbone clash
- Compute position/orientation signature for ligand
- Store hit in hashmap
- Filter based on hashmap collisions with previous constraints

Classic Matching



For each potential position:

- Build protein rotamers
- Build ligand rotamers
- Enumeratively combine pairs with geometry variations
- Filter on backbone clash
- Compute position/orientation signature for ligand
- Store hit in hashmap
- Filter based on hashmap collisions with previous constraints

Matching Algorithm

Classic Matching

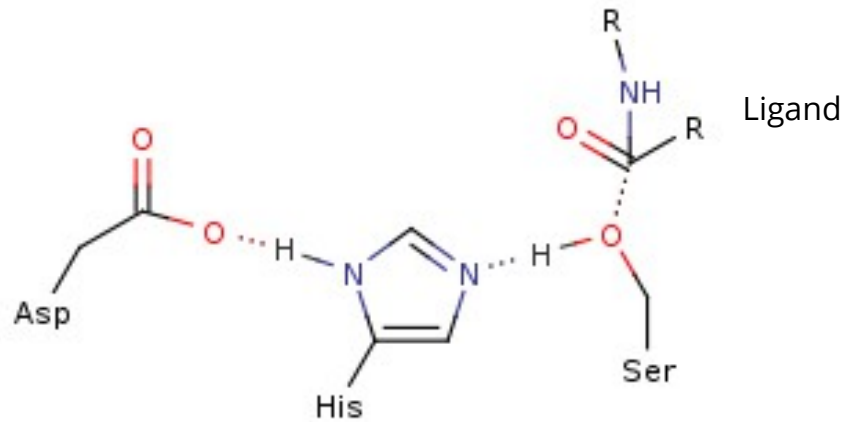
- Doesn't need pre-existing hit
- Must involve ligand
- Must specify all six degrees of freedom
- Fast of large number of potential matches

Secondary Matching

- Can omit degrees of freedom
- Can be protein-protein interaction
- Needs pre-existing hit (can't be first constraint)
- Fast for small number of potential matches

Secondary Matching

Allows matching protein-protein interactions

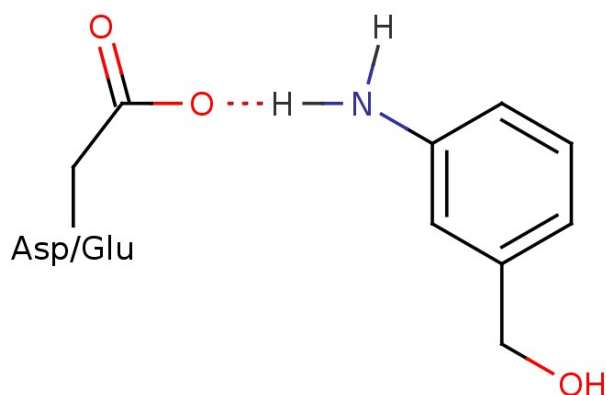


Catalytic Triad

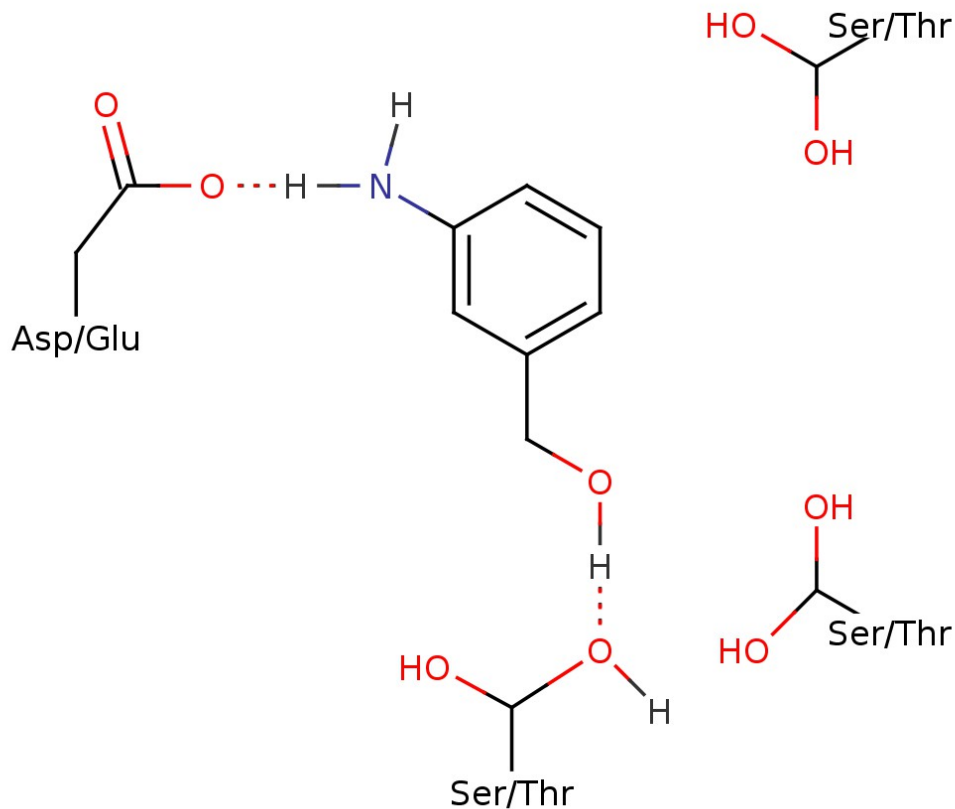
Secondary Matching

For each hit in hashmap:

- Build single ligand position
- Build protein rotamers
- Consider pairwise
- Compute constraint satisfaction
- If not satisfied, remove from hashmap

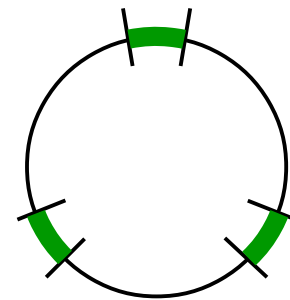


Secondary Matching



For each hit in hashmap:

- Build single ligand position
- Build protein rotamers
- Consider pairwise
- Compute constraint satisfaction
- If not satisfied, remove from hashmap



Secondary "Sampling"

Constraint File – Algorithm

```
ALGORITHM_INFO:: match
```

```
SECONDARY_MATCH: DOWNSTREAM
```

Secondary match
to ligand

```
ALGORITHM_INFO::END
```

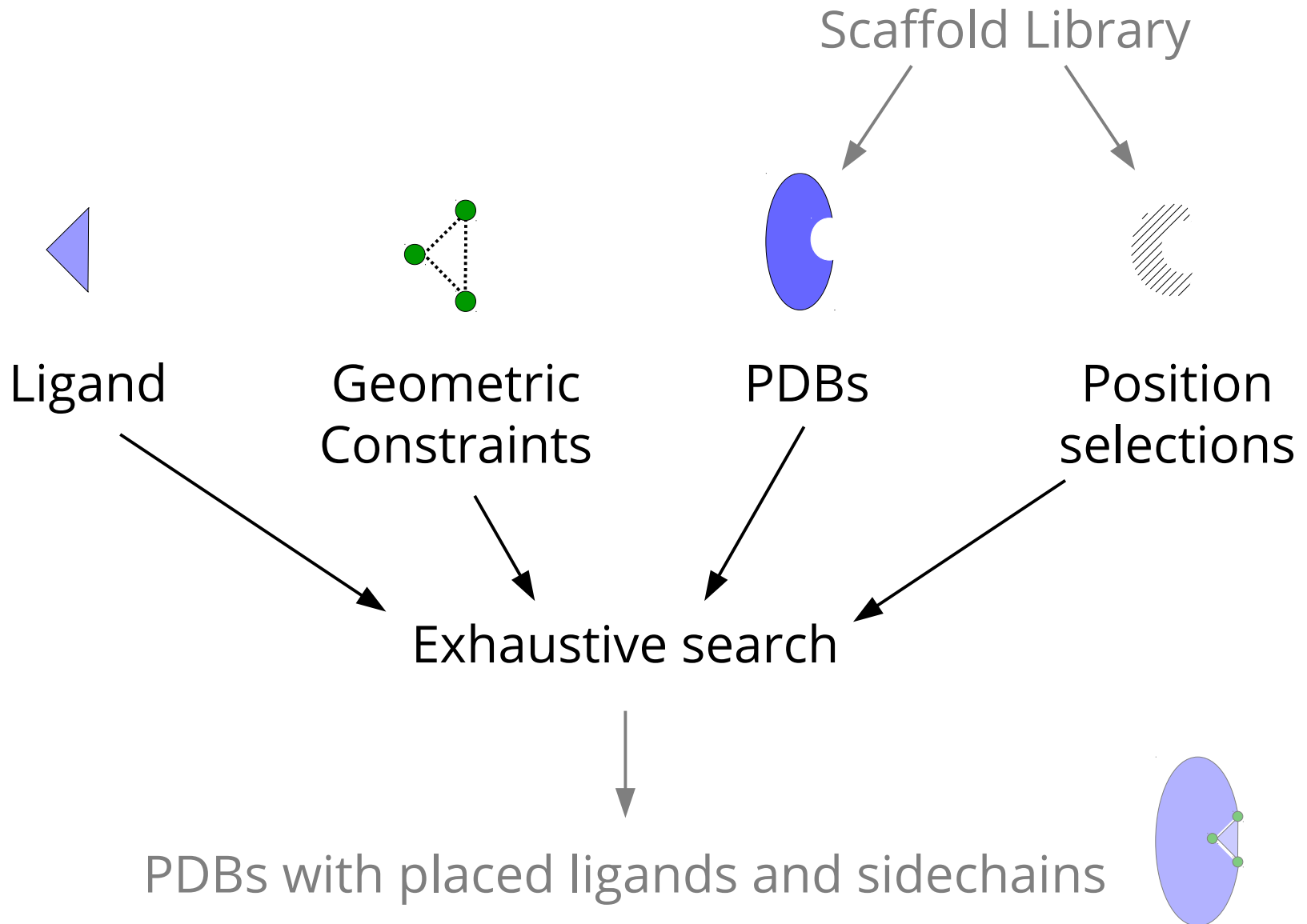
```
ALGORITHM_INFO:: match
```

Secondary match to
upstream residue of
given constraint block

```
SECONDARY_MATCH: UPSTREAM_CST 1
```

```
ALGORITHM_INFO::END
```

Overview



Running the matcher

Required

```
match.linuxgccrelease  
-s scaffold.pdb          # Scaffold PDB (without ligand)  
-match:scaffold_active_site_residues scaffold.pos # posfile  
-match:lig_name LG1      # PDB name of the ligand to match  
-extra_res_fa LG1.params # params file for ligand  
-match:geometric_constraint_file LG1.cst      # constraint file
```

Recommended

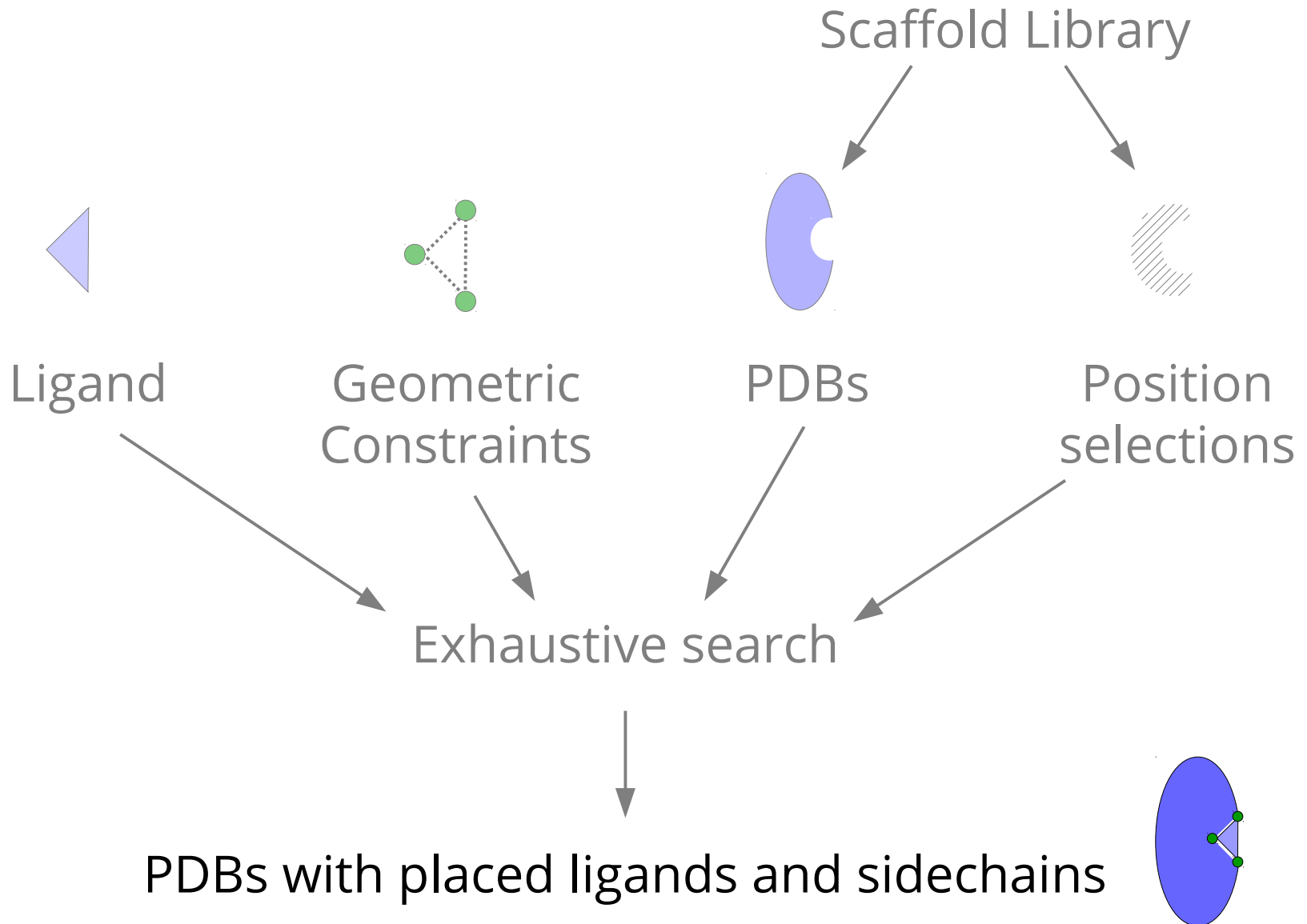
```
-ex1    # sidechain sampling flags  
-ex2    # (Standard packer flags apply)  
-use_input_sc
```

Optional

```
-match:output_format CloudPDB # Output format (CloudPDB default)  
  
-euclid_bin_size 1.0 # Fine tune hashing bins  
-euler_bin_size 10.0 # (defaults should be fine)  
  
-bump_tolerance 0.1 # Backbone clash threshold (default is fine)
```

WARNING: Can be memory intensive!

Overview



Output

➡ Option 1: Separate PDBs for each match

`-match:output_format PDB`

LOTS of PDBs.

➡ Default: Cloud PDB

Compact representation: Multi-model PDB
with one example match with “related” matches as
rotamers.

`-parser_read_cloud_pdb`

Allows “jd2” protocols to read in CloudPDBs

PDB Constraint Header

Constraint Block

REMARK	666	MATCH	TEMPLATE	X	D2N	0	MATCH	MOTIF	A	CYS	10	1	1
REMARK	666	MATCH	TEMPLATE	X	D2N	0	MATCH	MOTIF	A	SER	98	3	1
REMARK	666	MATCH	TEMPLATE	X	D2N	0	MATCH	MOTIF	A	HIS	144	2	1
REMARK	666	MATCH	TEMPLATE	X	D2N	0	MATCH	MOTIF	A	TRP	100	5	1

Residue for A
(0 means
every position)

Residue for B

-run:preserve_header

References

New algorithms and an in silico benchmark for computational enzyme design.

Zanghellini *et al.* (2006) Protein Sci. 15(12):2785-94.
<http://dx.doi.org/10.1110/ps.062353106>

De novo enzyme design using Rosetta3.

Richter *et al.* (2011) PLoS One. 6(5):e19230.
<http://dx.doi.org/10.1371/journal.pone.0019230>

Matcher Documentation (Rosetta 3.5)

https://www.rosettacommons.org/manuals/archive/rosetta3.5_user_guide/d7/dfc/match.html

Constraint File Documentation (Rosetta 3.5)

https://www.rosettacommons.org/manuals/archive/rosetta3.5_user_guide/d5/dd4/match_cstfile_format.html

Enzyme design demo:

[Rosetta/demos/public/enzyme_design/](https://www.rosettacommons.org/demos/public/enzyme_design/)