

# VIRTUAL COMBINATORIAL SCREENING FOR PDZ- BINDING PEPTIDES

BRYAN CRAMPTON

# THE GOAL

- **We have:** Peptides known to bind to PDZ domain of a particular protein (GIPC, CAL, PSD-95)
- **We want:** Compounds with higher binding affinity or other desired chemical properties, either:
  1. Modified versions of current peptides
  2. New core peptides (with or without modifications)

# THE GOAL

- **We have:** Peptides known to bind to PDZ domain of a particular protein (GIPC, CAL, PSD-95)
- **We want:** Compounds with higher binding affinity or other desired chemical properties, either:
  1. **Modified versions of current peptides**
  2. New core peptides (with or without modifications)

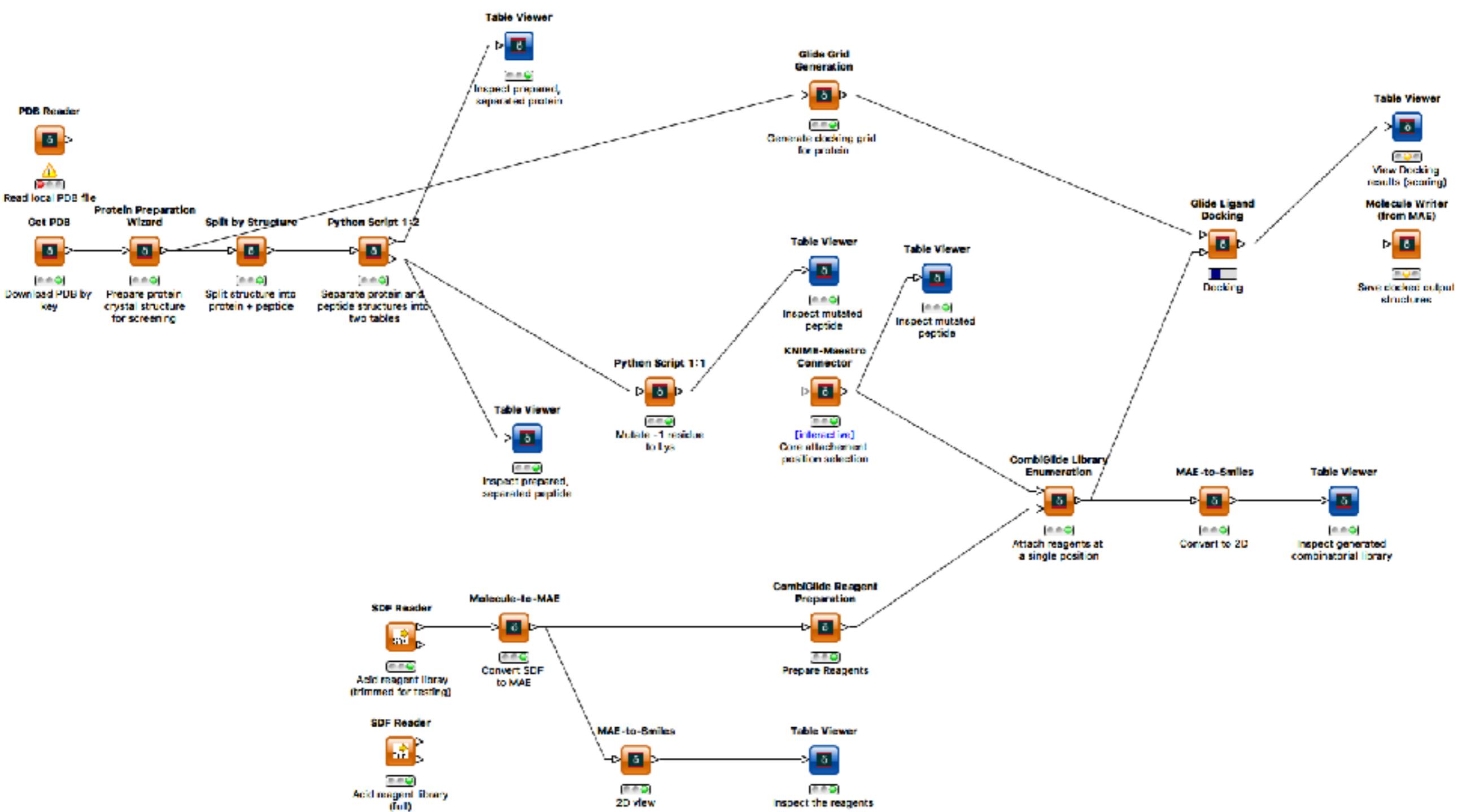
# INITIAL APPROACH

- Input crystal structure of protein-peptide complex
- Modify -1 residue of peptide to **Lysine**—core molecule
- Combinatorially attach acids from library (purchasable) to the terminal amino group of Lys<sub>-1</sub> on core
- Dock each modified peptide compound, searching for higher binding affinity than core molecule

# BACKGROUND

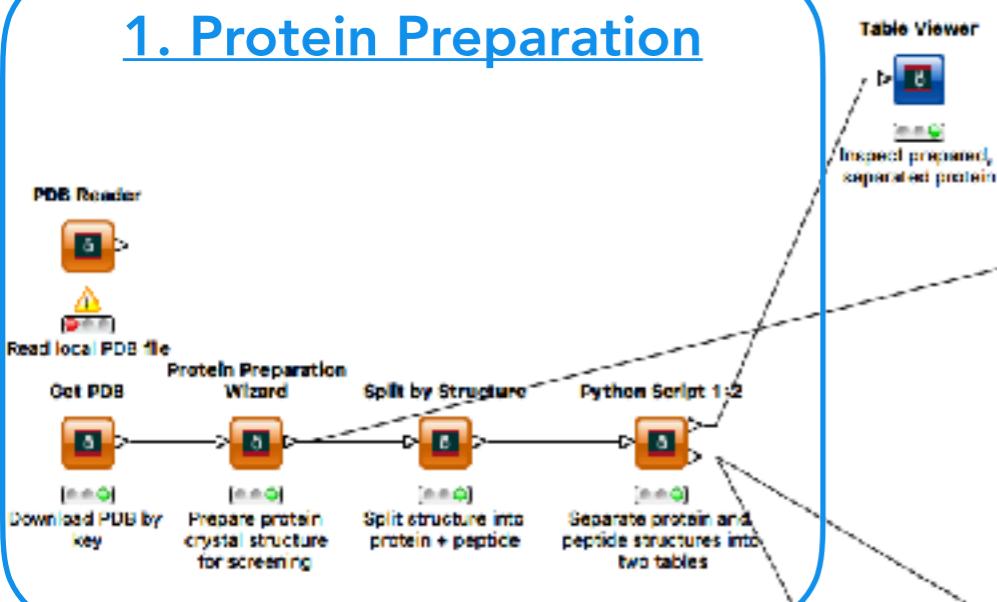
- **Schrödinger**: Small-Molecule Drug Discovery Suite
  - **ProteinPrep**: *Protein structure refinement and fixes*
    - Add hydrogens, add missing side chains, remove unneeded / incorrectly placed waters, assign bond orders, add disulfides, optimize hydroxyl, Asn, and His states
  - **LigPrep**: *Ligand structure refinement and conformer generation*
    - Add hydrogens, ionization, tautomers, stereoisomers (if desired), etc.
  - **Glide**: *Grid generation and docking protocol. HTVS, SP, XP*
  - **CombiGlide**: *Combinatorial library enumeration, preparation, and docking*
  - **KNIME**: *Open-source automation / data mining software with Schrödinger plugins*
    - Setup workflow trees to control data flow, reduce error, and save time

# THE WORKFLOW (KNIME)



# KEY COMPONENTS

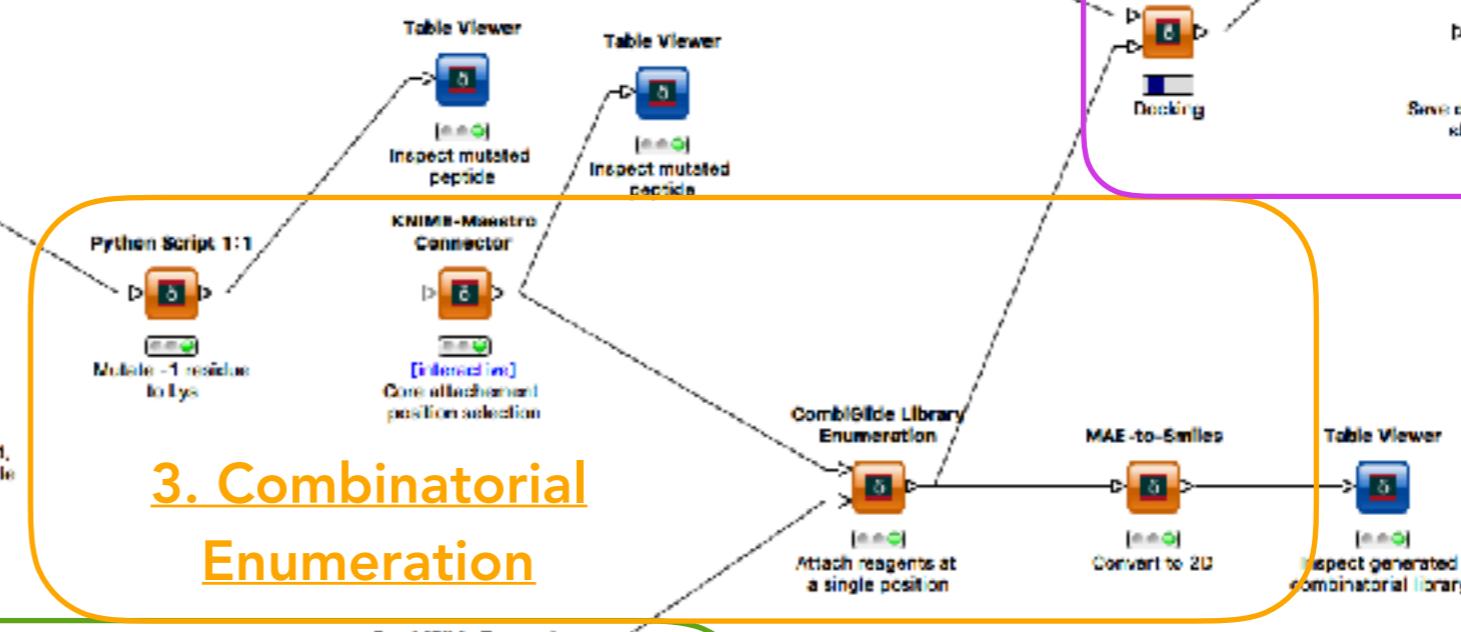
## 1. Protein Preparation



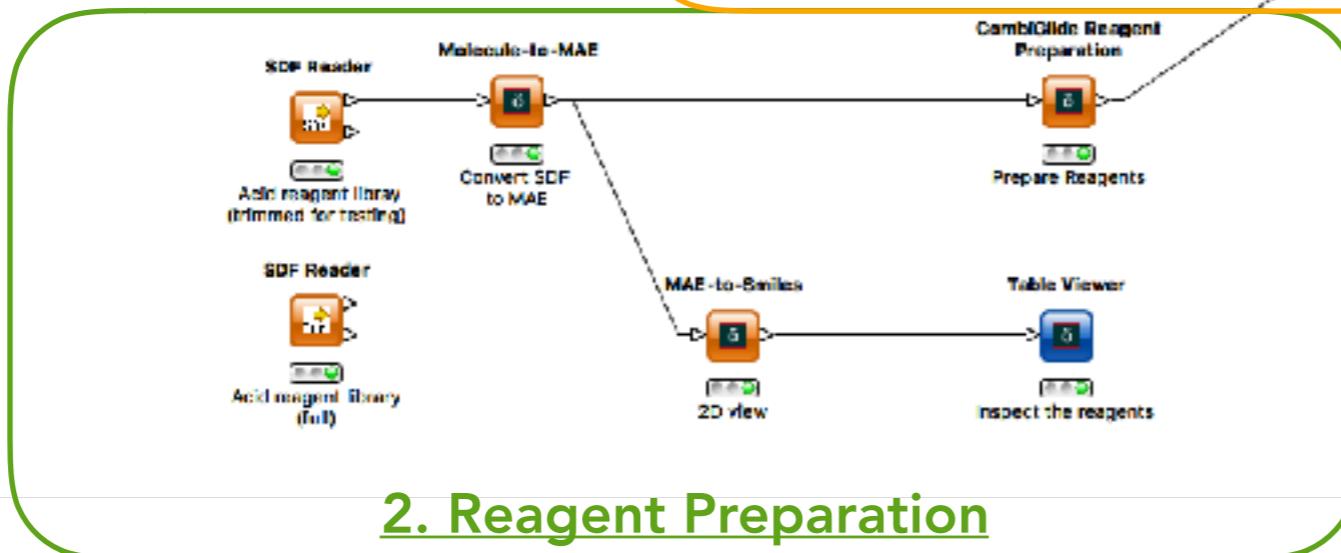
## 4. Glide Receptor Grid Preparation



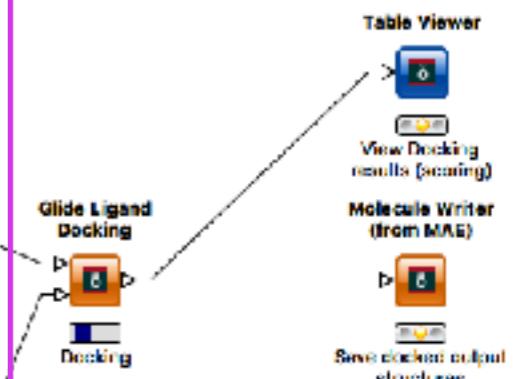
## 3. Combinatorial Enumeration



## 2. Reagent Preparation

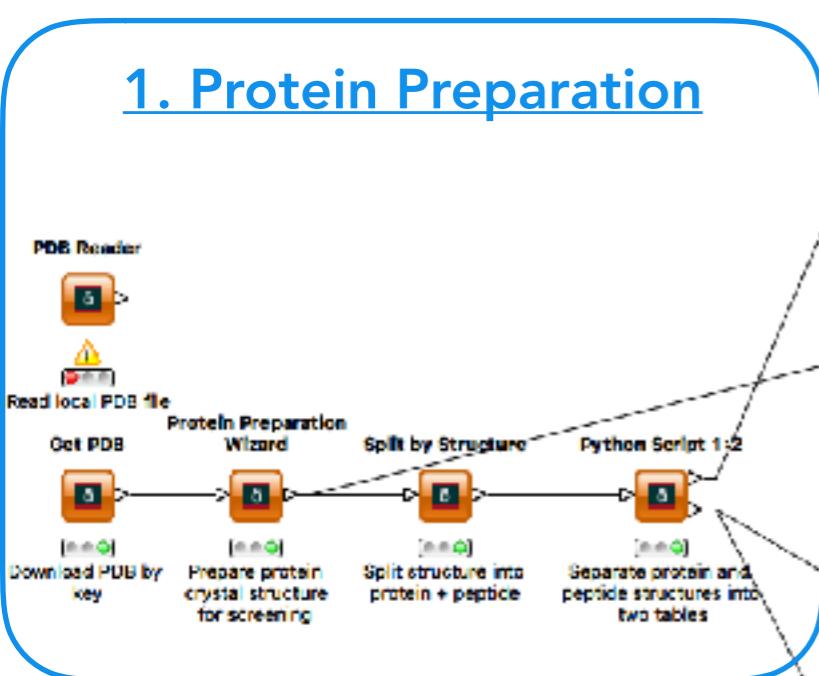


## 5. Combinatorial Docking & Scoring



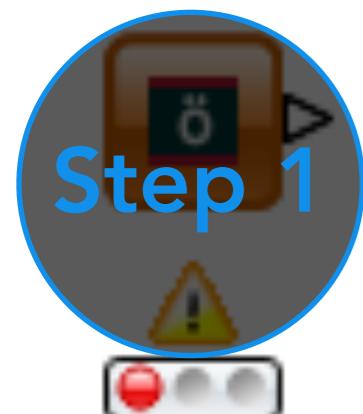
# COMPONENT 1: PROTEIN PREPARATION

## 1. Protein Preparation



# COMPONENT 1: PROTEIN PREPARATION

## PDB Reader



Read local PDB file

## Protein Preparation Wizard



Download PDB by key



Prepare protein crystal structure for screening

## Split by Structure



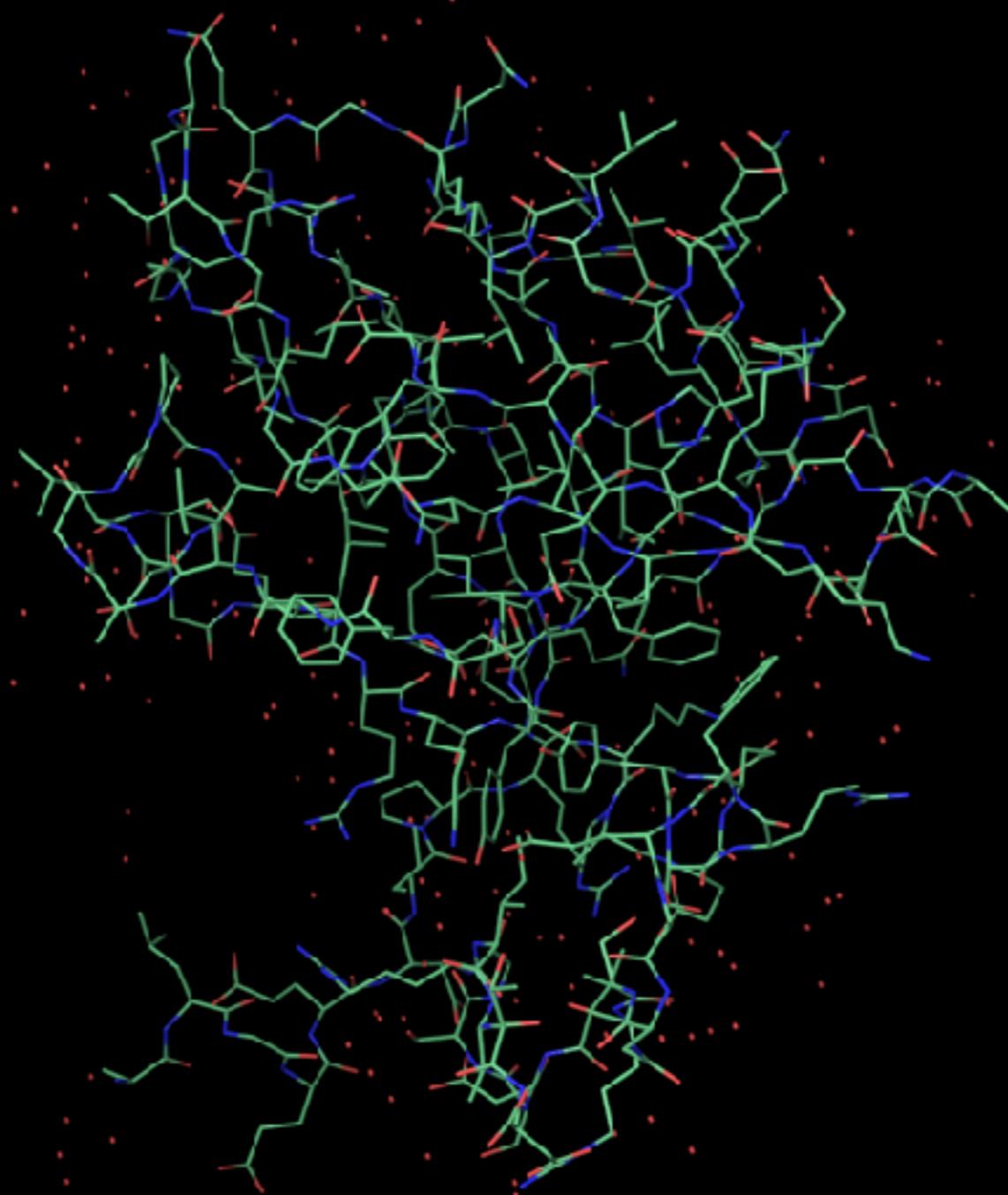
Split structure into protein + peptide

## Python Script 1:2



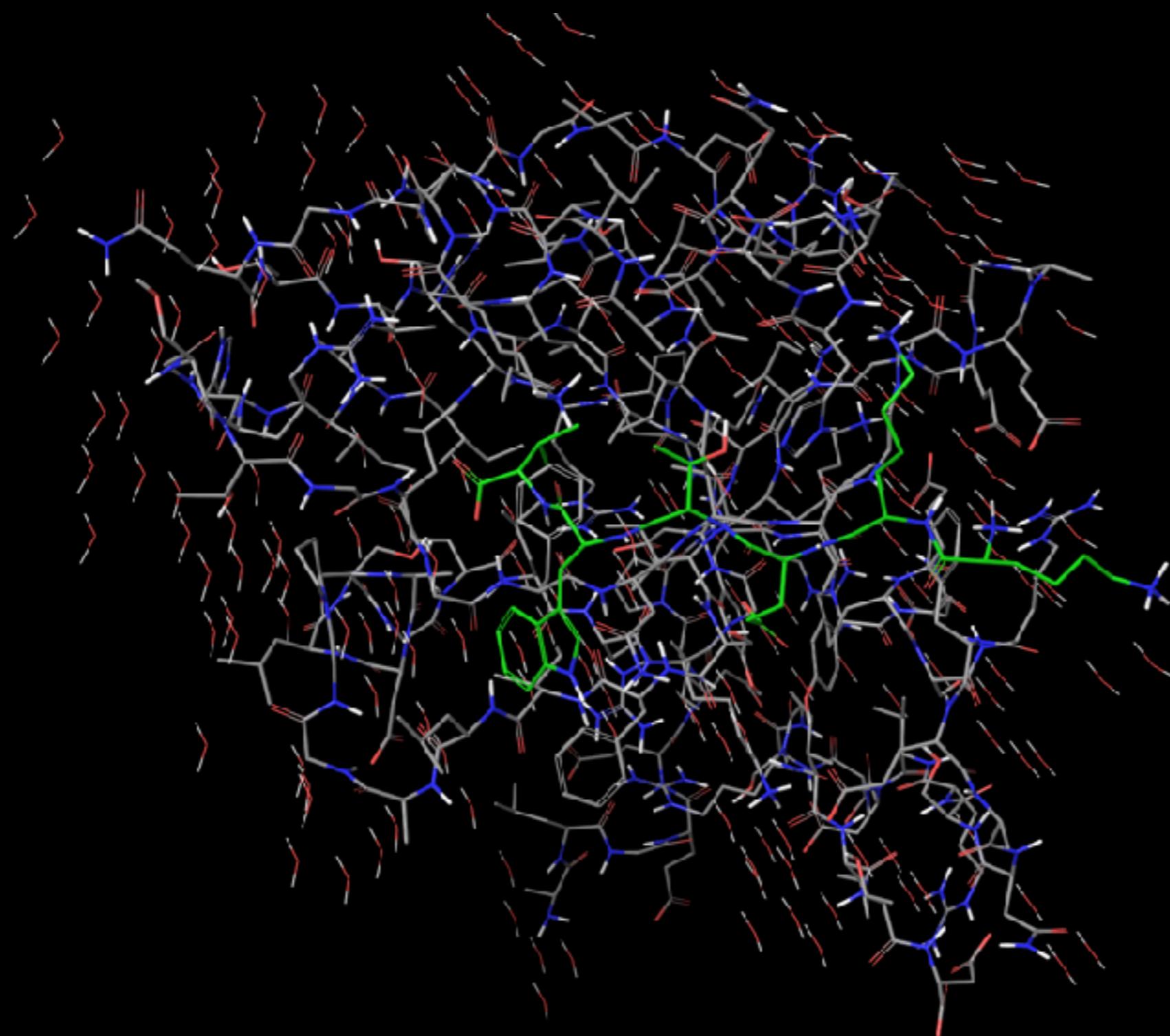
Separate protein and peptide structures into two tables

# COMPONENT 1: PROTEIN PREPARATION



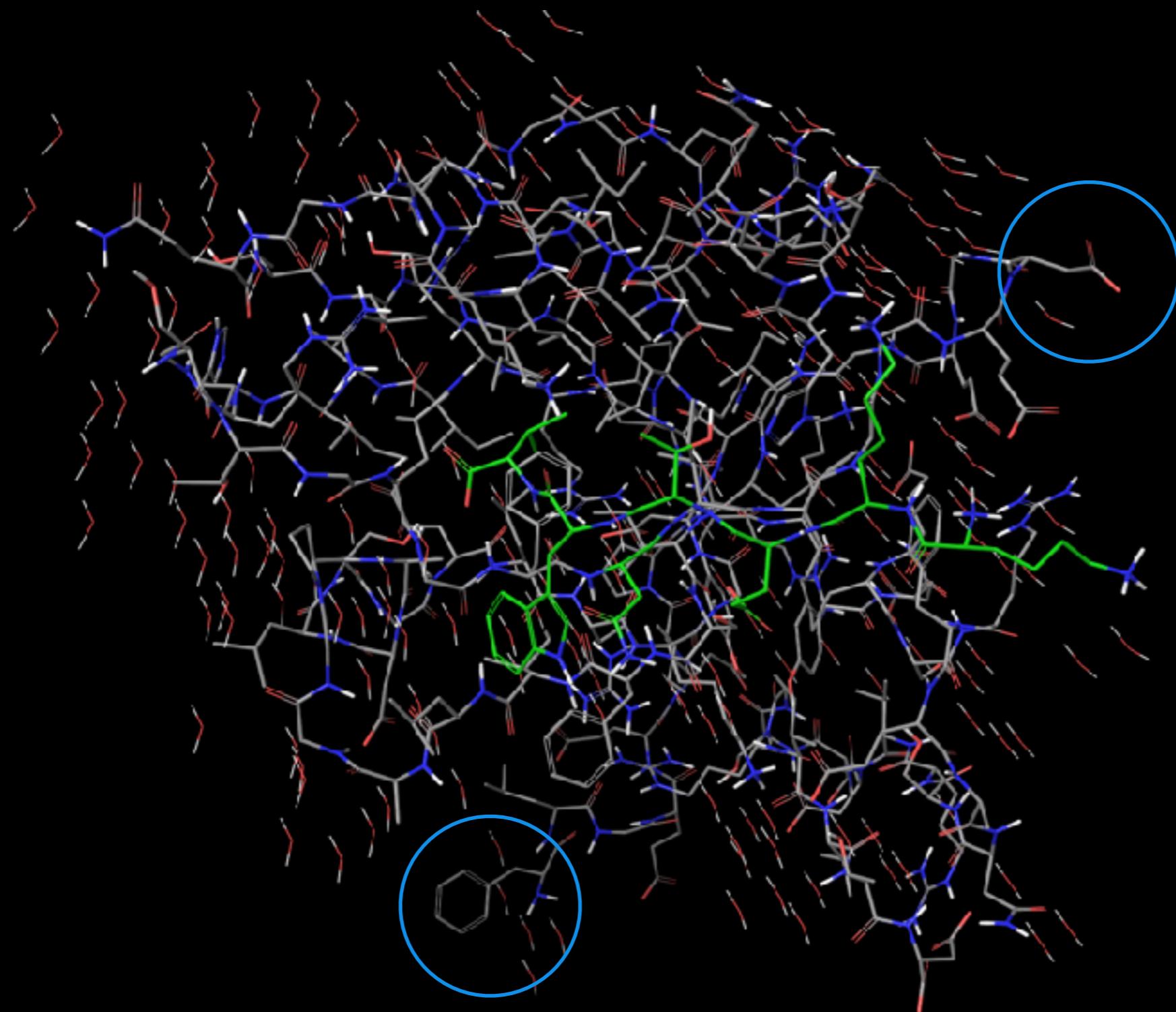
Step 1: Download structure from PDB directly or read from file

# COMPONENT 1: PROTEIN PREPARATION



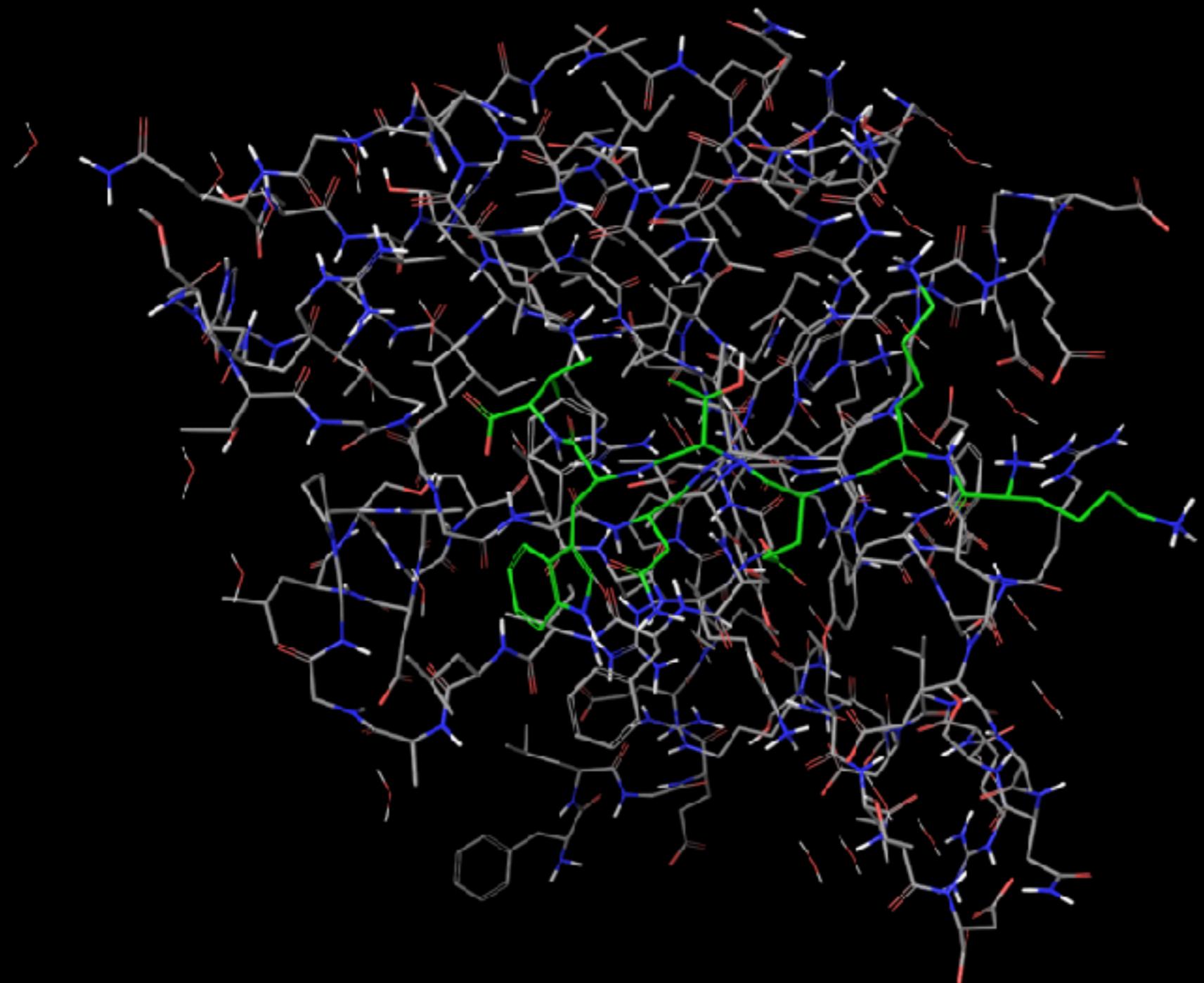
Step 2A: Protein Preparation — Add hydrogens

# COMPONENT 1: PROTEIN PREPARATION



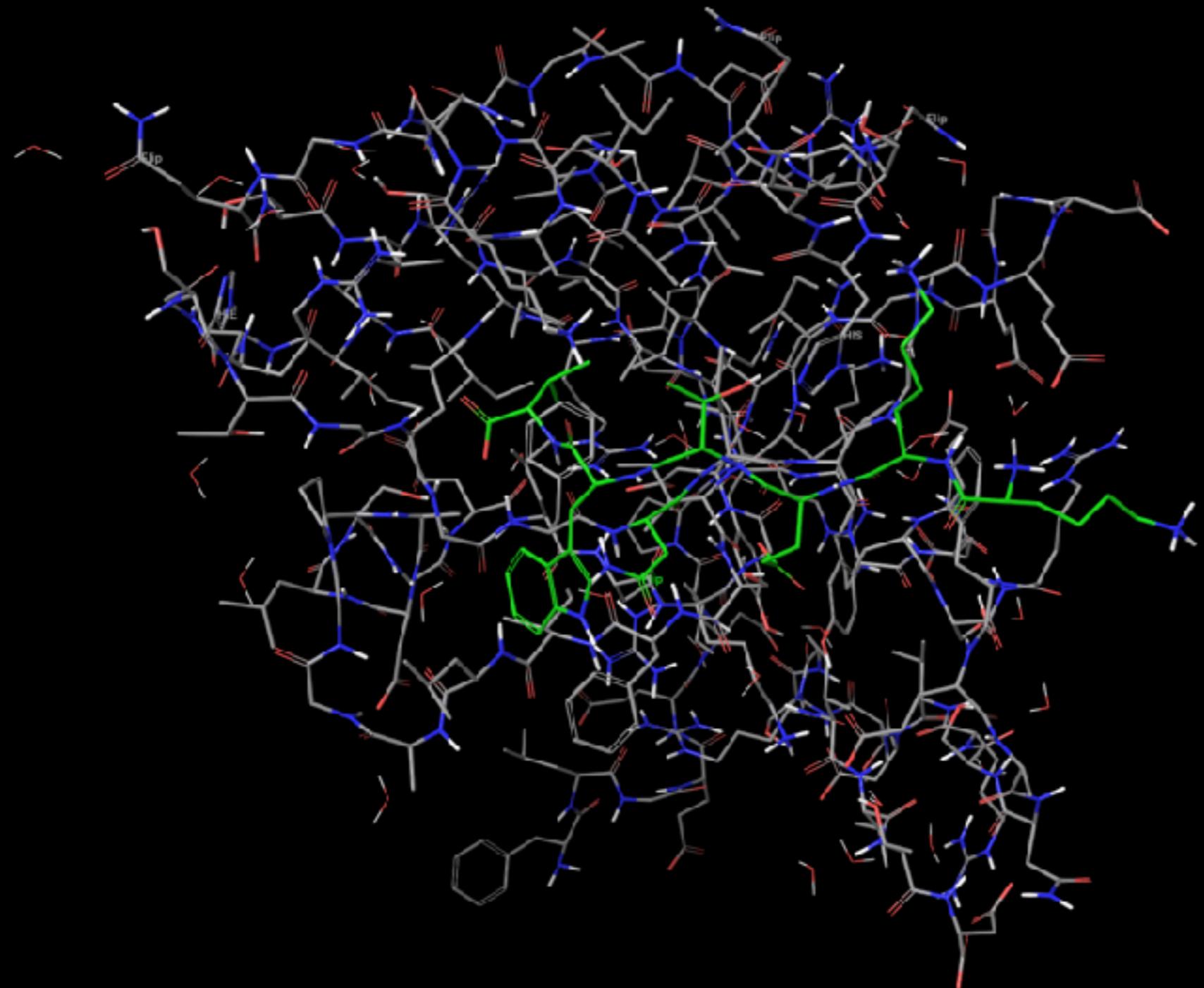
Step 2B: Protein Preparation — Add missing side chains

# COMPONENT 1: PROTEIN PREPARATION



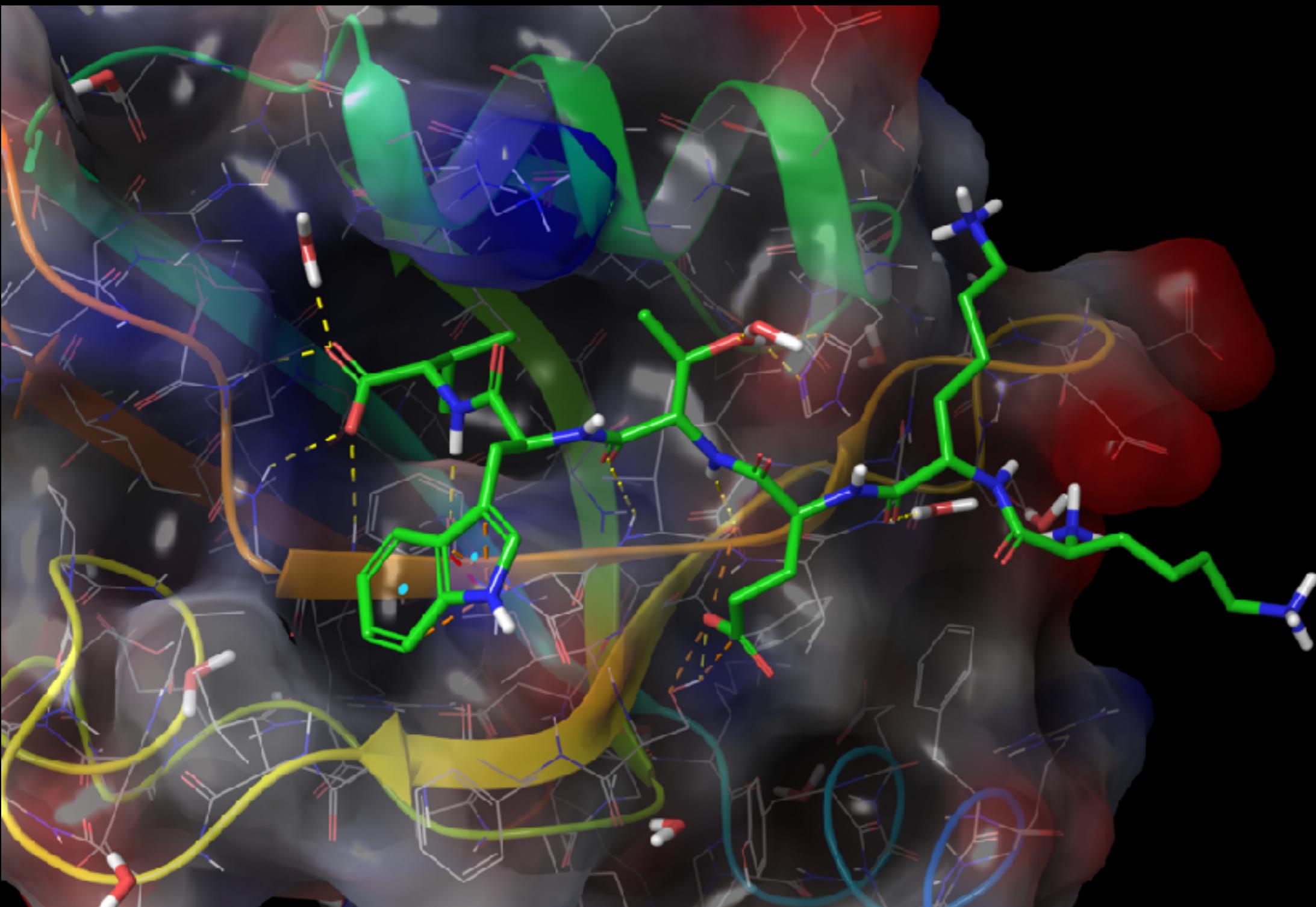
Step 2C: Protein Preparation — Remove extraneous H<sub>2</sub>O

# COMPONENT 1: PROTEIN PREPARATION



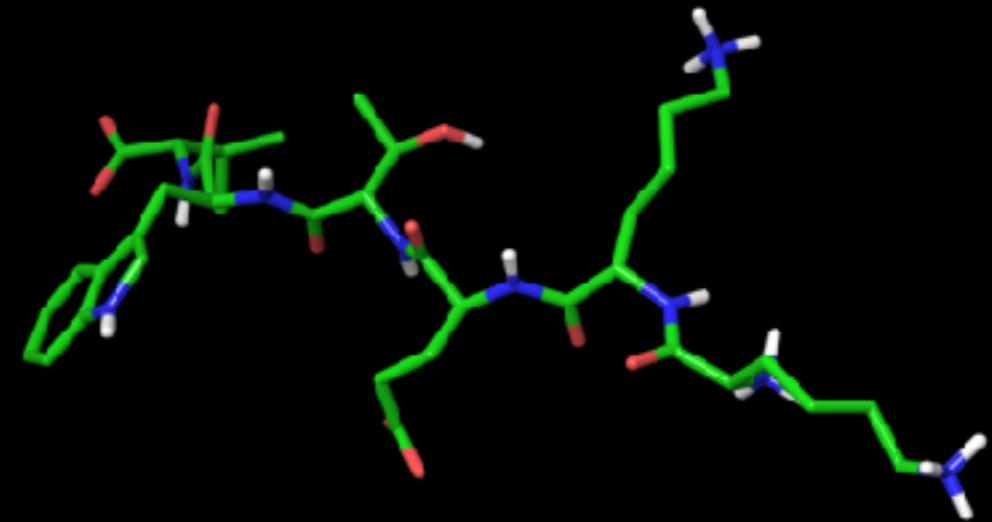
Step 2D: Protein Preparation — Optimize H-bond network (flips, charge, etc.)

# COMPONENT 1: PROTEIN PREPARATION



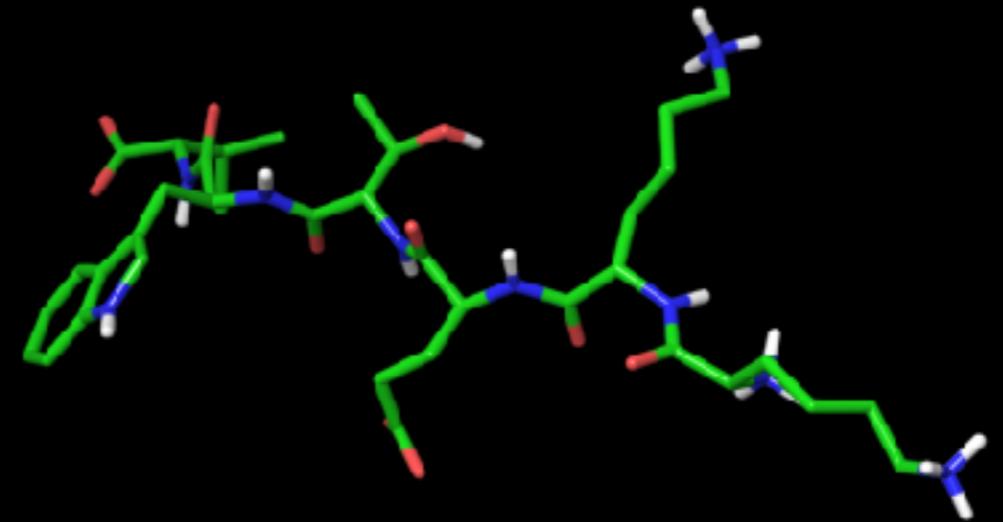
Step 2E: Protein Preparation — Visualize intermolecular interactions & surface

# COMPONENT 1: PROTEIN PREPARATION



Step 3: Split Structure — Separate peptide structure from protein (and H<sub>2</sub>O)

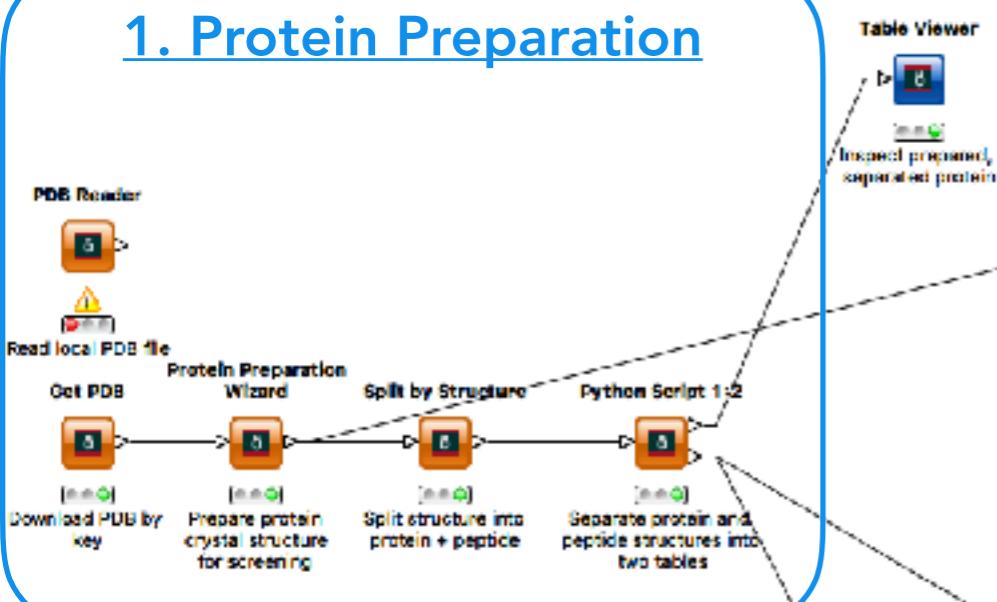
# COMPONENT 1: PROTEIN PREPARATION



Step 4: Structure Distinguishing—python script determines based on size

# KEY COMPONENTS

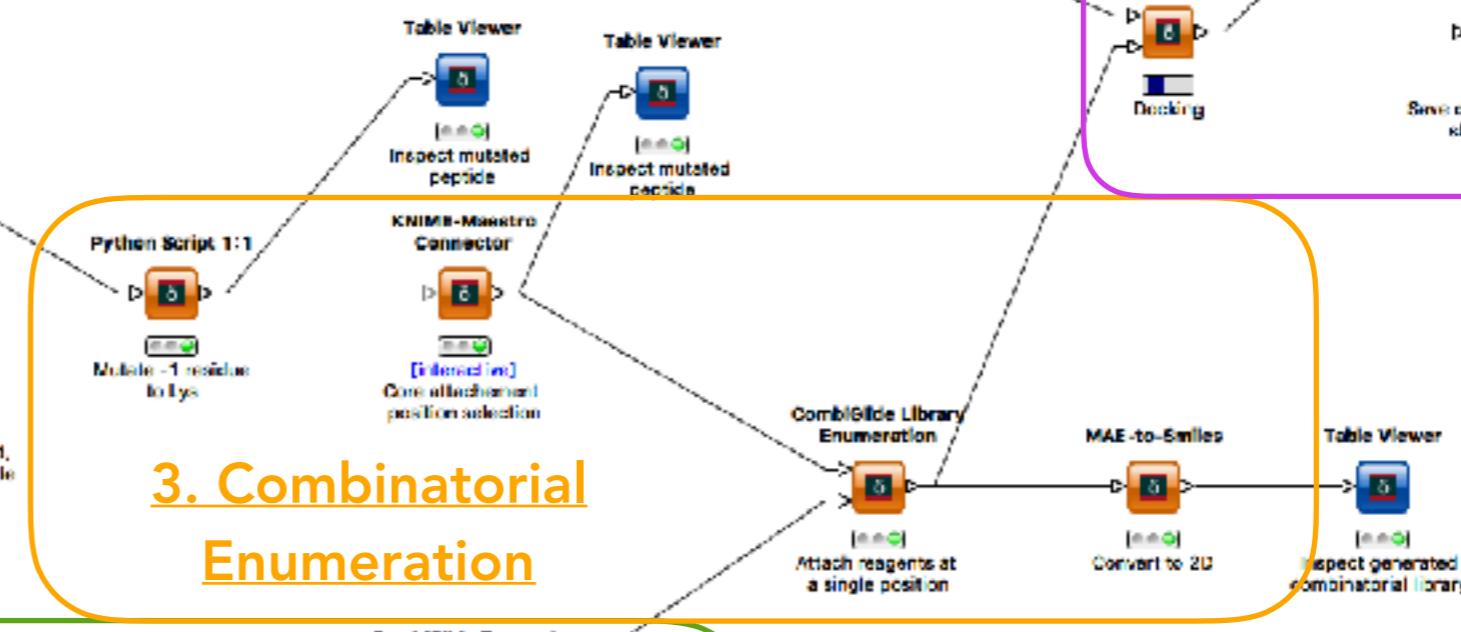
## 1. Protein Preparation



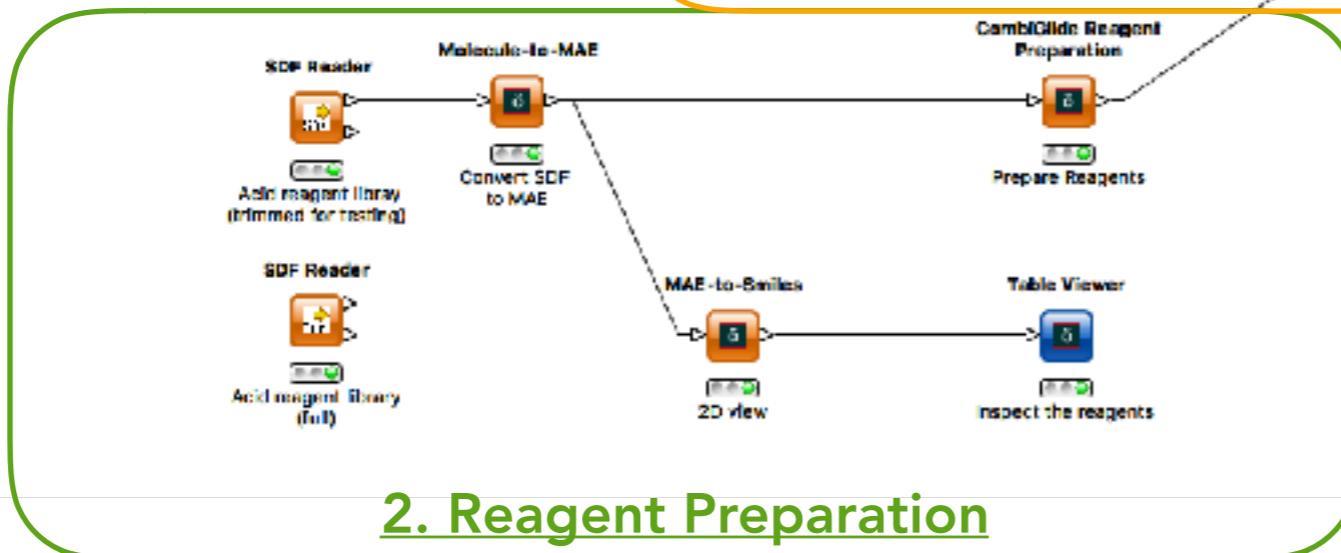
## 4. Glide Receptor Grid Preparation



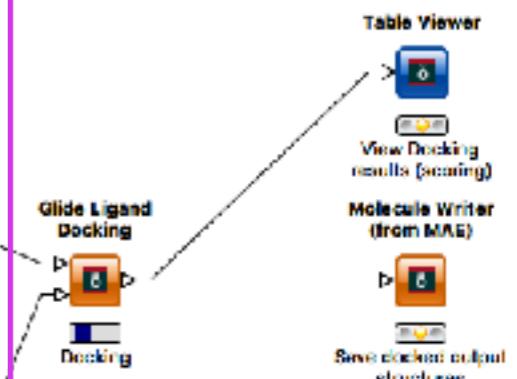
## 3. Combinatorial Enumeration



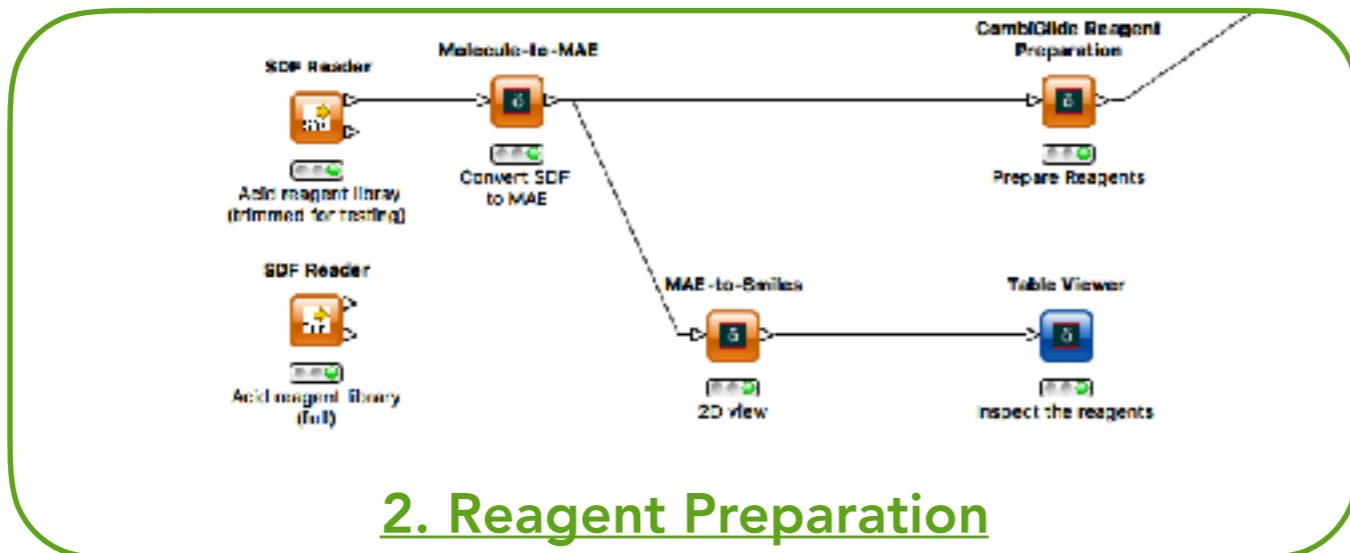
## 2. Reagent Preparation



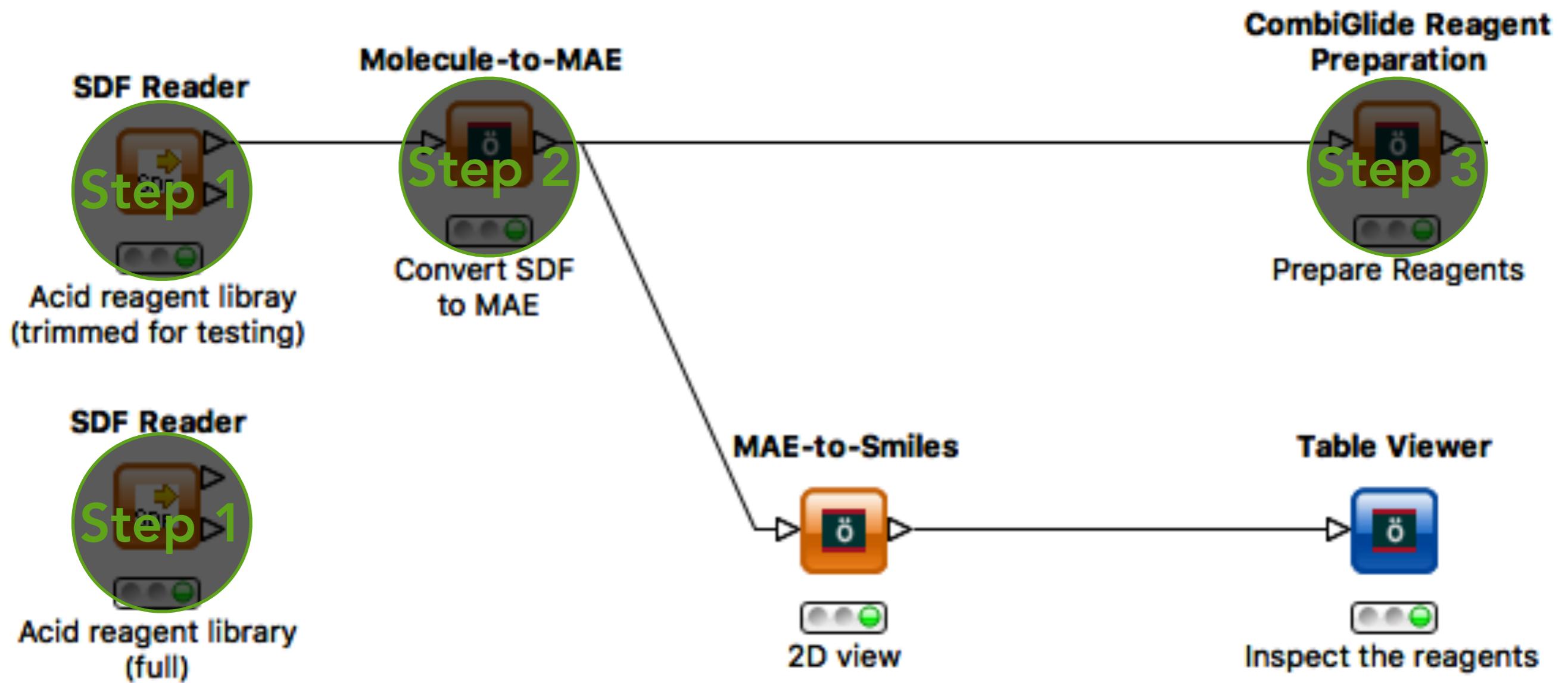
## 5. Combinatorial Docking & Scoring



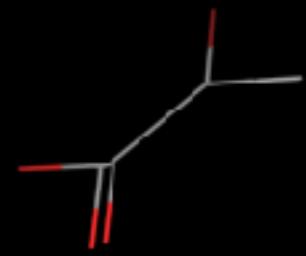
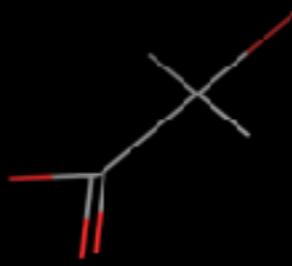
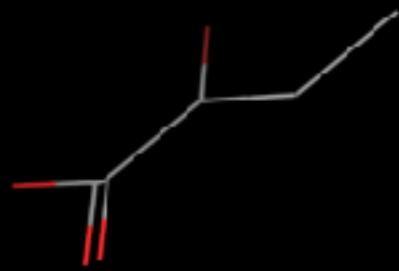
# COMPONENT 2: REAGENT PREPARATION



# COMPONENT 2: REAGENT PREPARATION



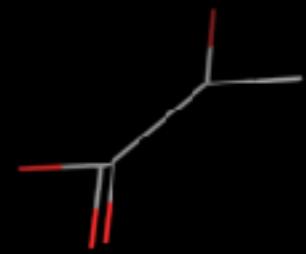
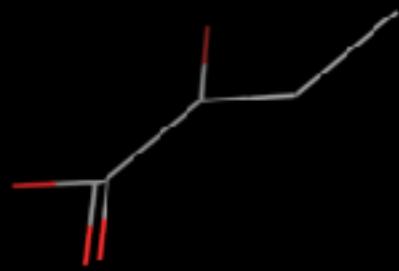
## COMPONENT 2: REAGENT PREPARATION



Trimmed acid library displayed (>> 100k acid compounds in actual)

Step 1: SDF Reader—Load acid library from SDF file (2D chemical formula)

## COMPONENT 2: REAGENT PREPARATION



Trimmed acid library displayed (>> 100k acid compounds in actual)

Step 2: Molecule-to-MAE—Convert SDF format to maestro format (python)

# COMPONENT 2: REAGENT PREPARATION

Reagent Preparation

Use structures from: Workspace (included entries)

Input file:

Create reagent titles from:

SD molecule name  SD property:

Functional group to identify (black atoms are kept, red atoms are lost):  
[Alk = alkyl; Ar = aryl or heteraryl; Vi = vinyl; R = Alk, Ar, or Vi; A= any atom]


Selected functional group: Carboxylic\_Acid\_C\_O

Treatment of multiple occurrences: Produce a structure only if all are equivalent

Ionize, using pH: 7.0

Generate tautomers

Use Epik for ionization and tautomerization

Include metal-binding states

Generate stereoisomers: 10 per reagent

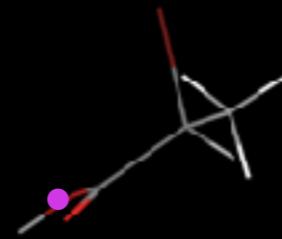
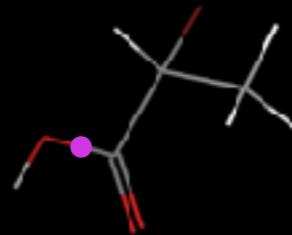
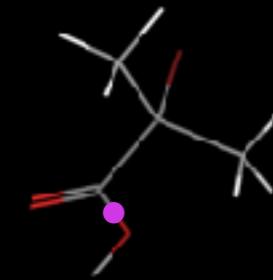
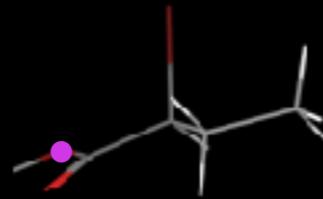
Generate low-energy ring conformations: 1 per reagent

Job name: cg\_C\_Acid\_C\_O\_2

Combicle: Reagent Preparation, Hohlsheimheat

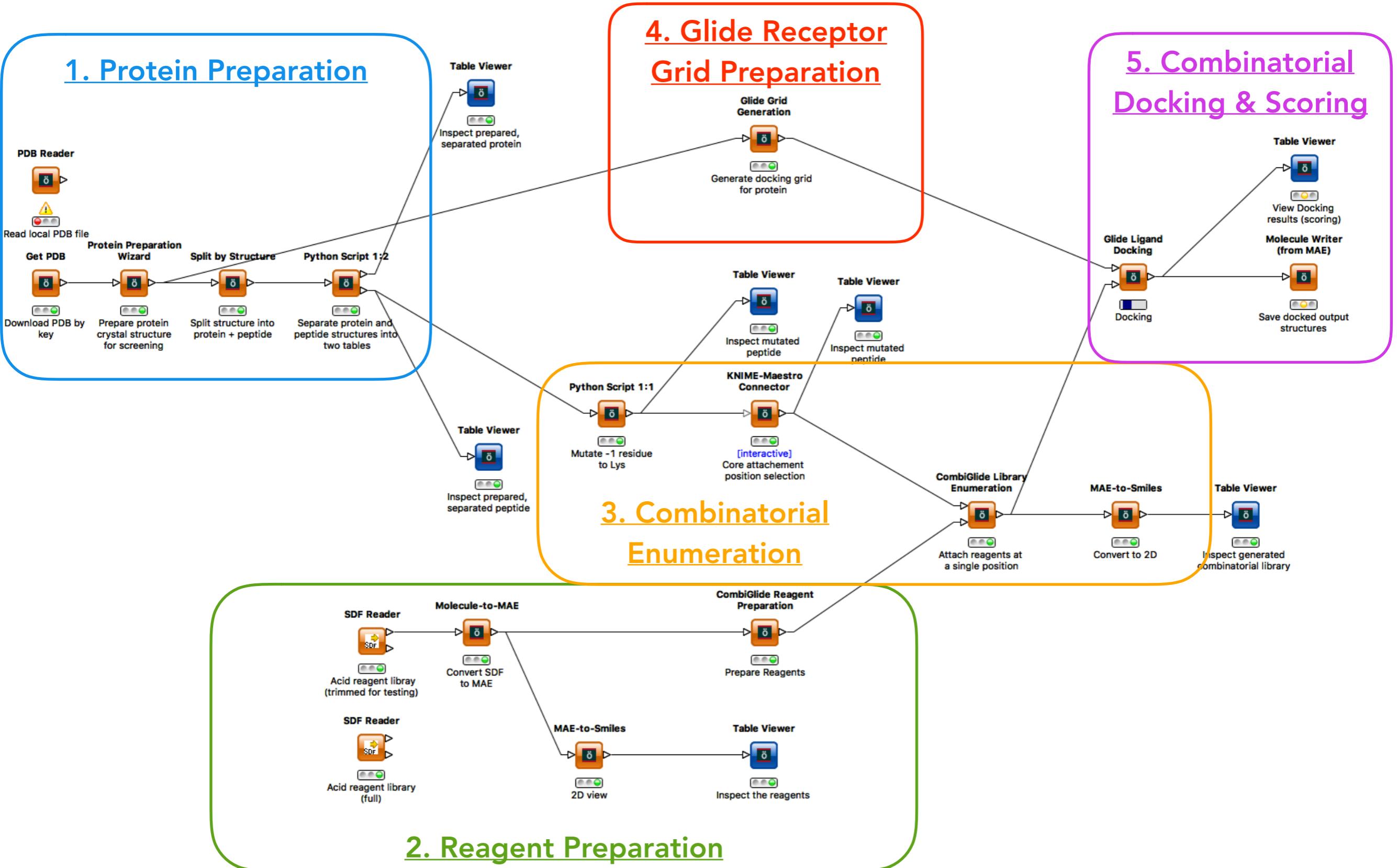
Step 3: Reagent Preparation—Specify broken bond, generate 3D conformers

## COMPONENT 2: REAGENT PREPARATION

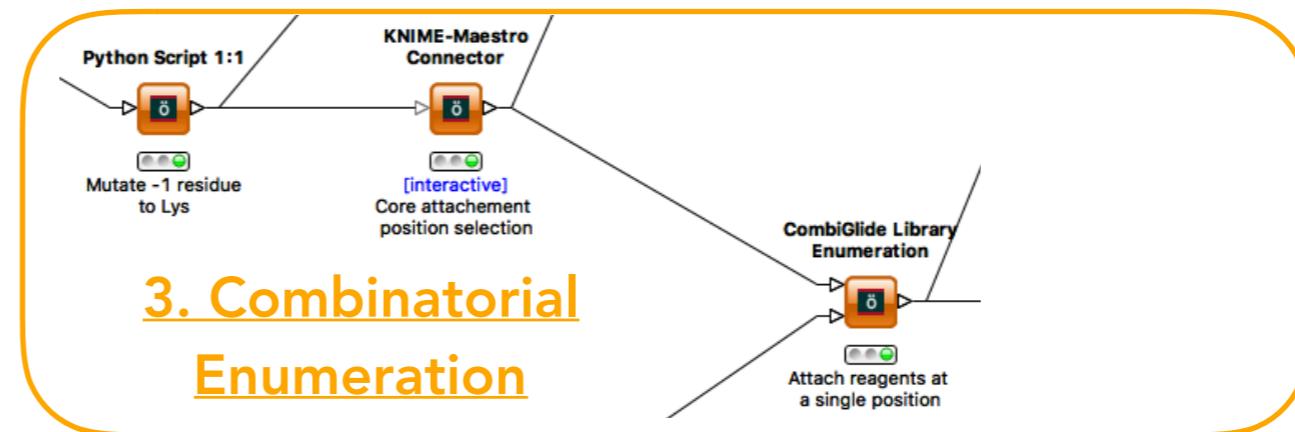


Step 3: Reagent Preparation—Specify broken bond, generate 3D conformers

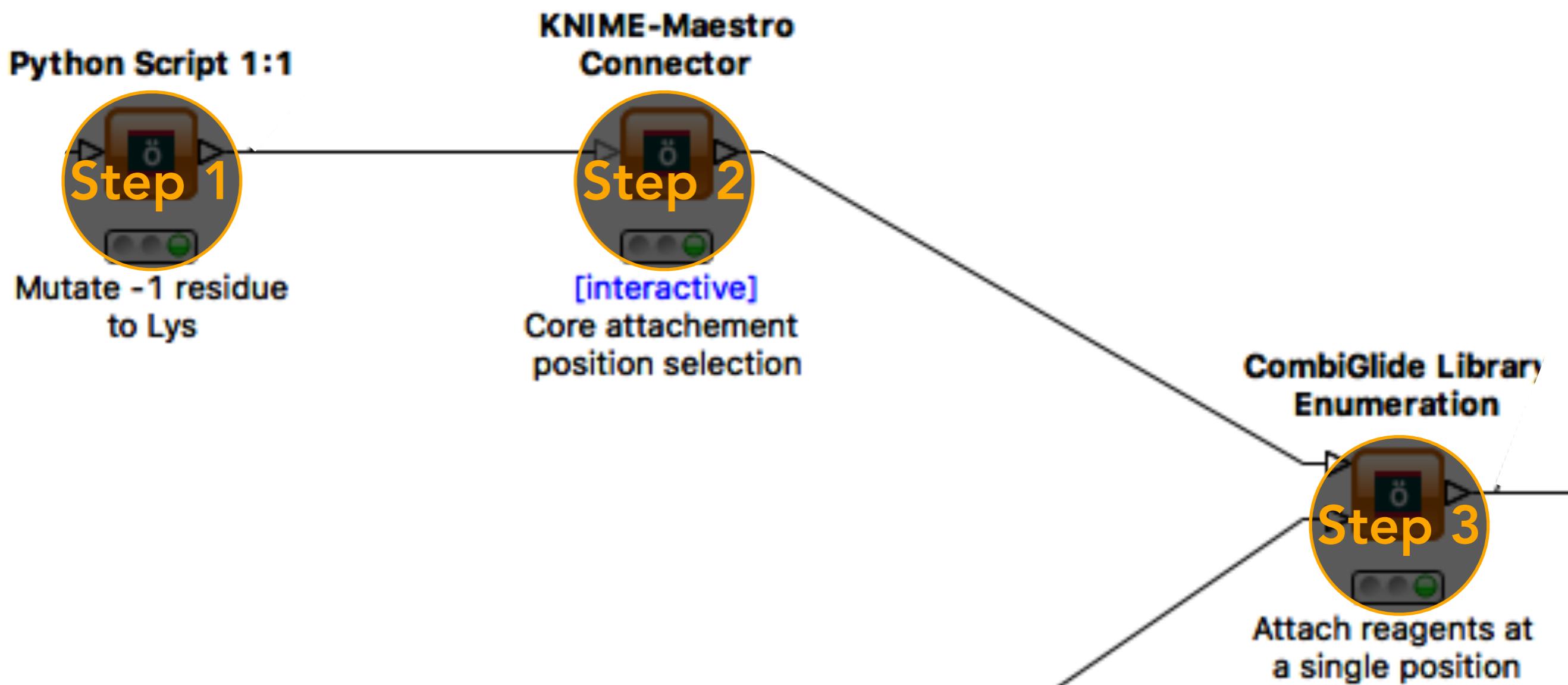
# KEY COMPONENTS



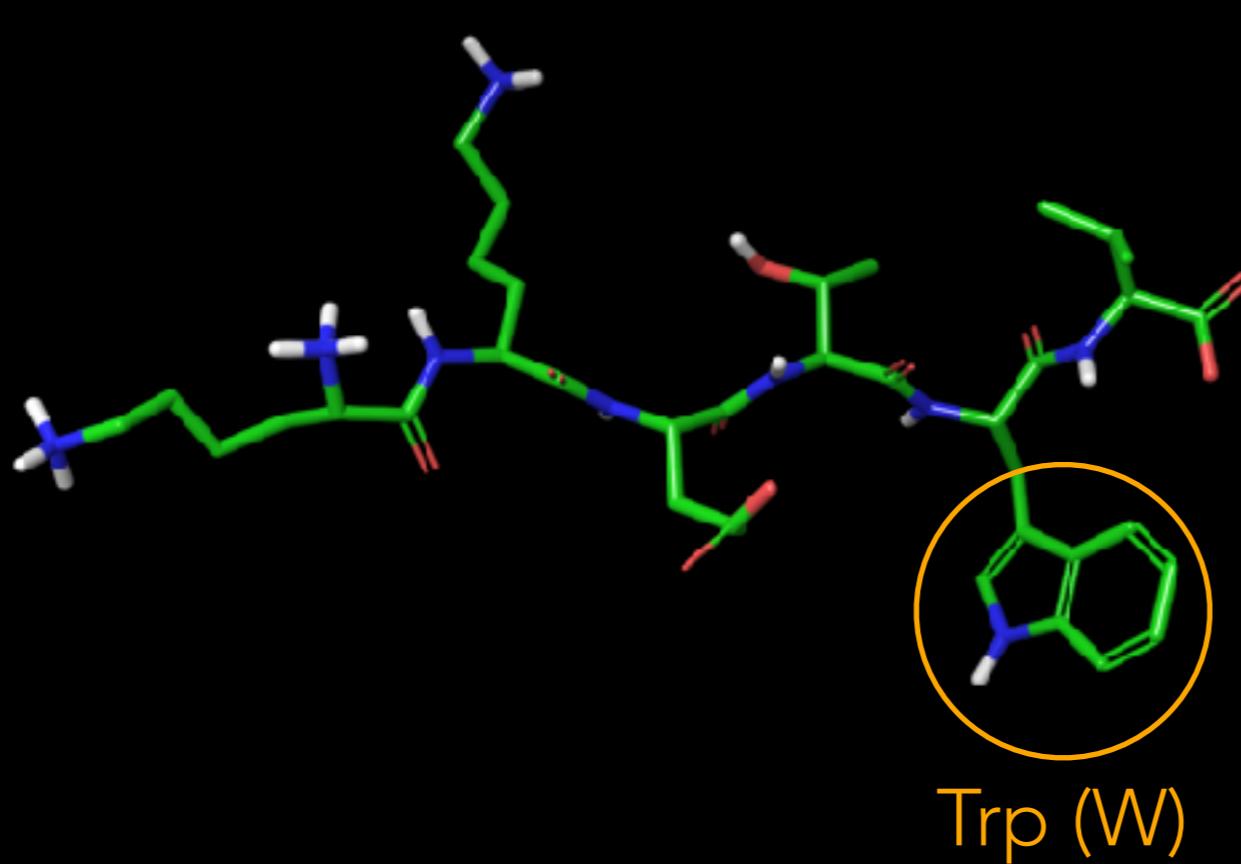
# COMPONENT 3: COMBINATORIAL ENUMERATION



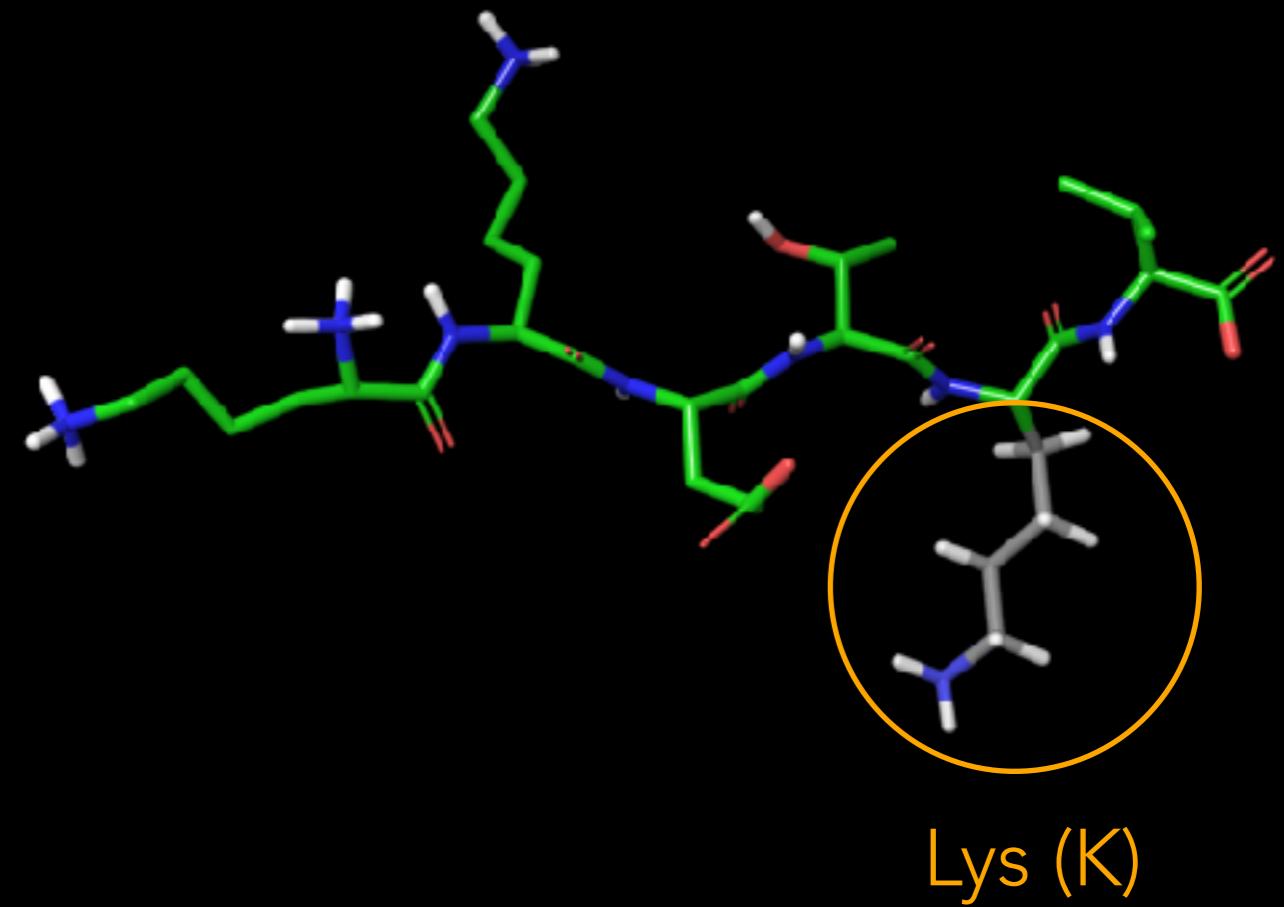
# COMPONENT 3: COMBINATORIAL ENUMERATION



# COMPONENT 3: COMBINATORIAL ENUMERATION



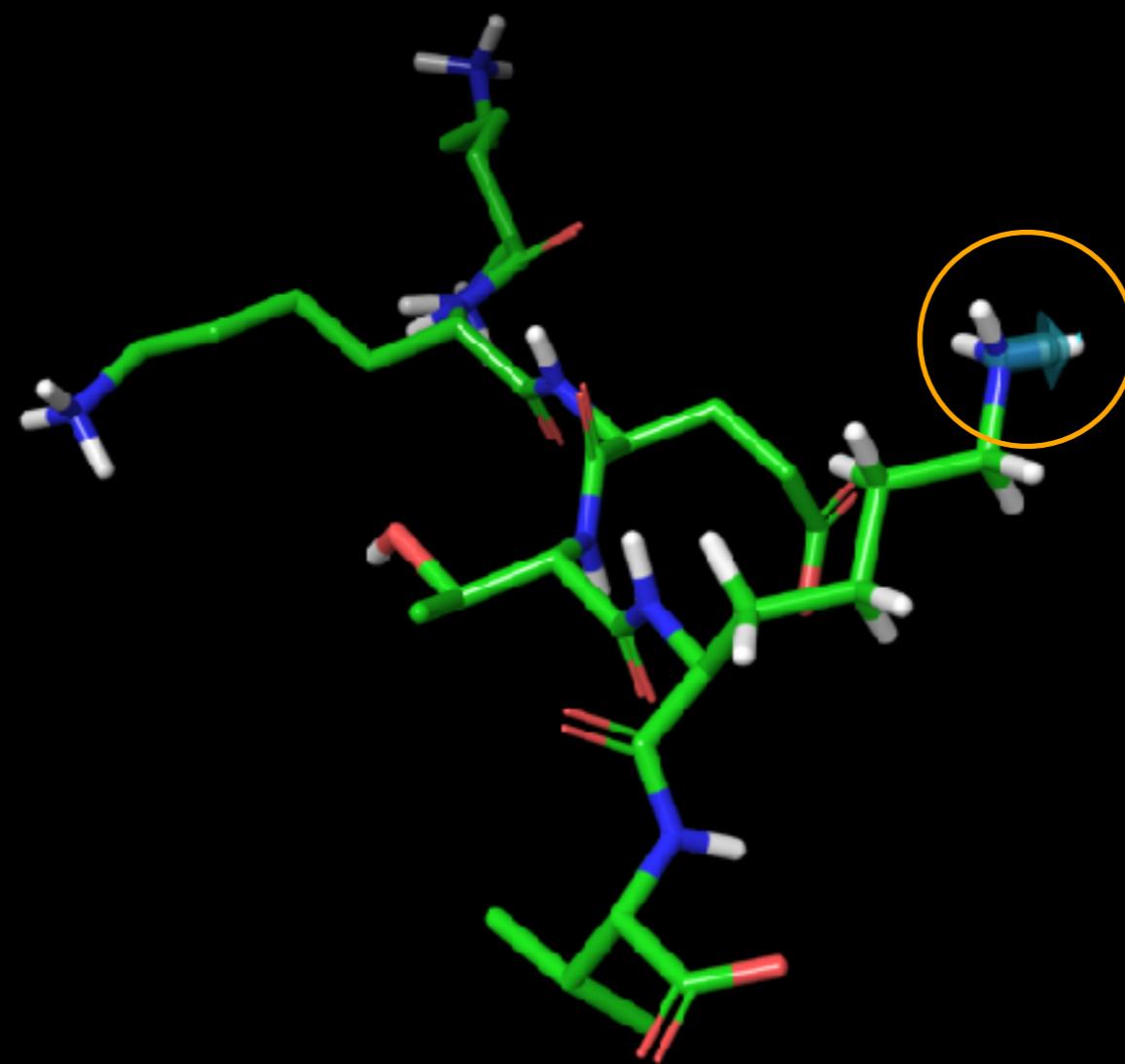
Trp (W)



Lys (K)

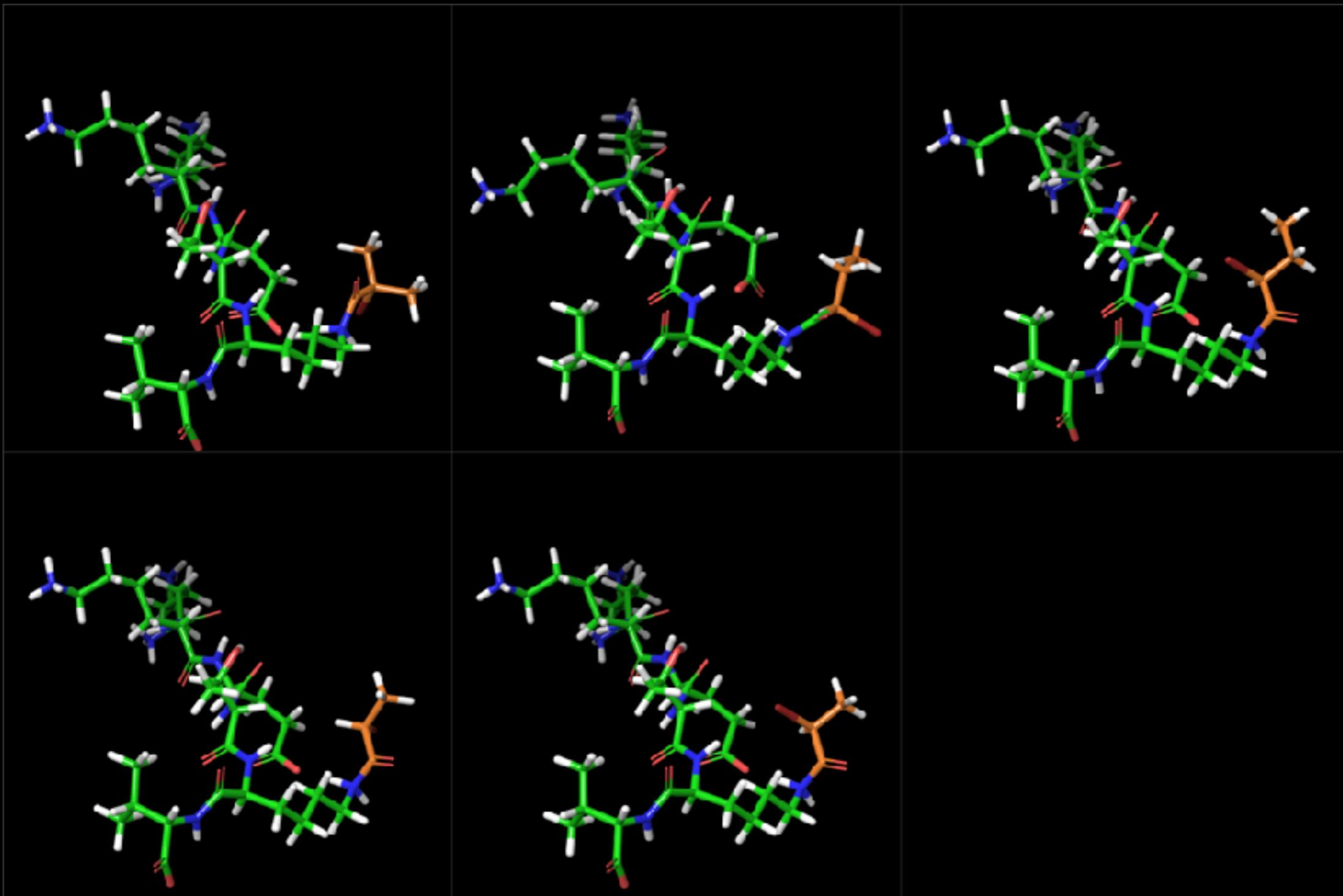
Step 1: Mutate Residue—python script finds and modifies -1 residue —> Lys

# COMPONENT 3: COMBINATORIAL ENUMERATION



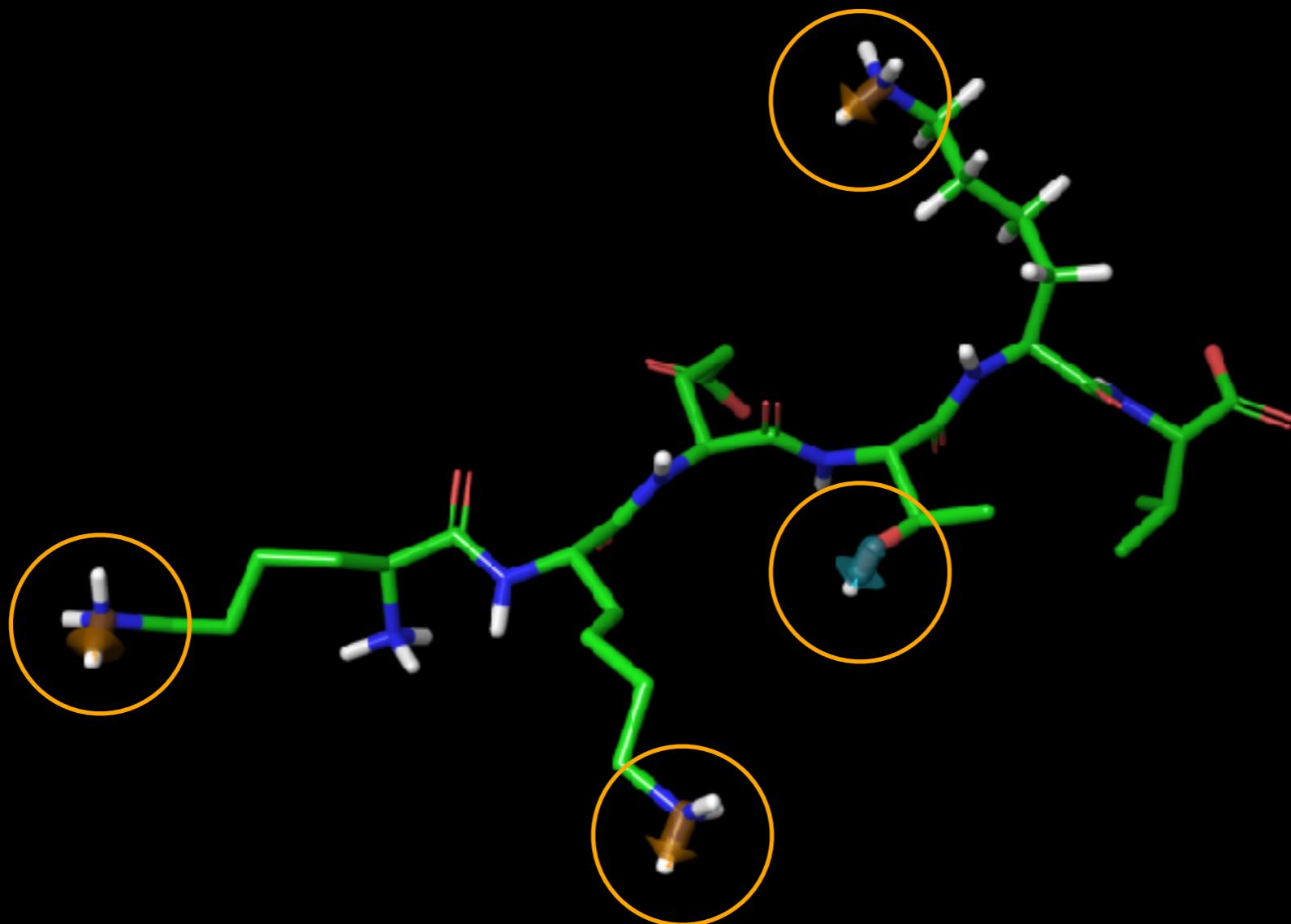
Step 2: Core Attachment Site—interactive GUI (for now) to choose location

# COMPONENT 3: COMBINATORIAL ENUMERATION

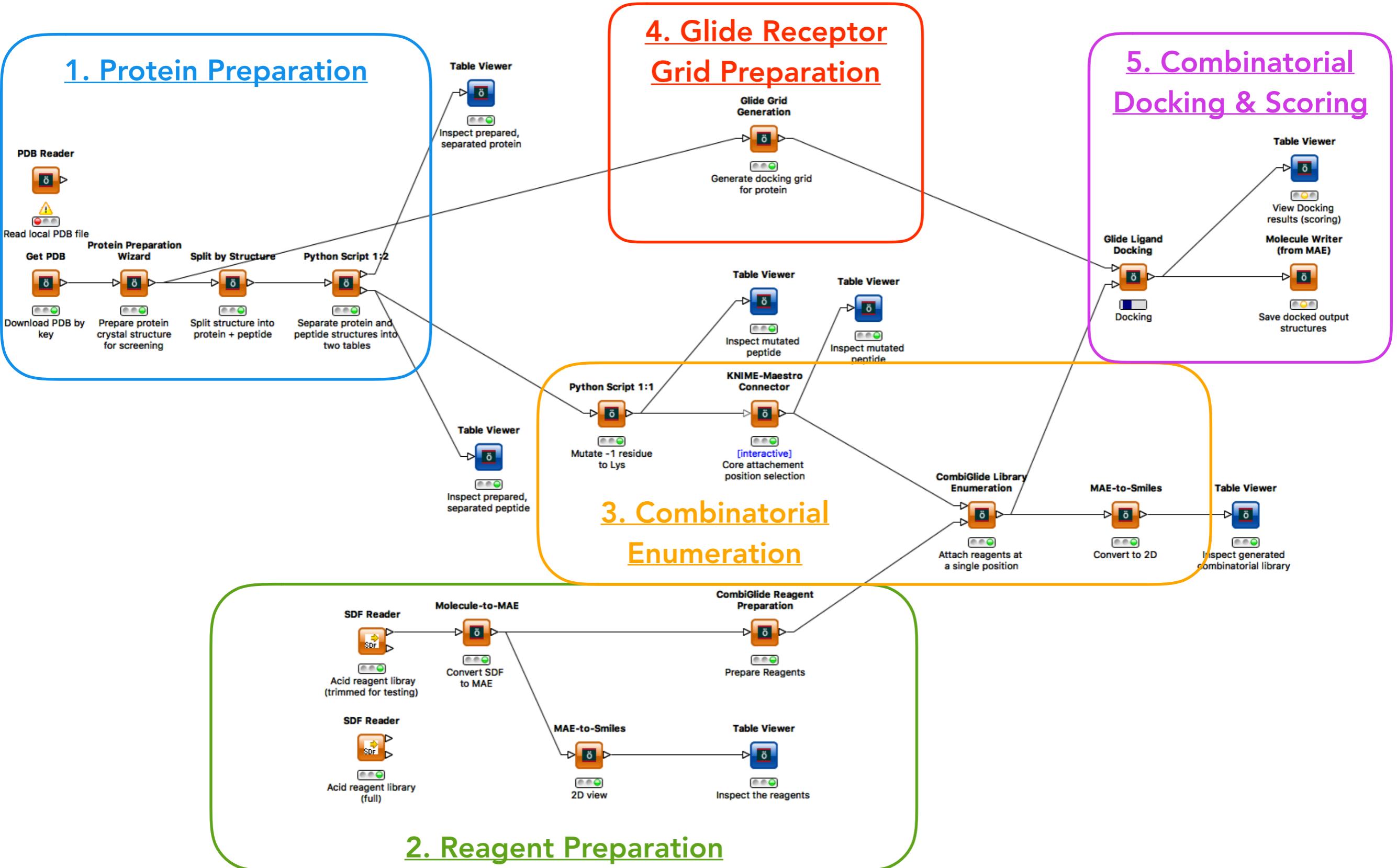


Step 3: Combinatorial Enumeration—Create library by coupling all fragments with core attachment site

# SIDE NOTE: REAL COMBINATORIAL ENUMERATION



# KEY COMPONENTS



# COMPONENT 4: GLIDE RECEPTOR GRID GENERATION

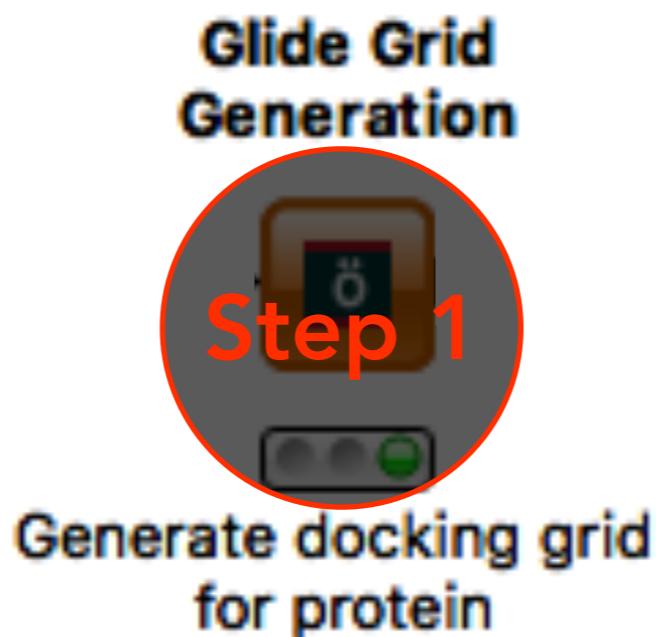
## 4. Glide Receptor Grid Preparation

Glide Grid Generation

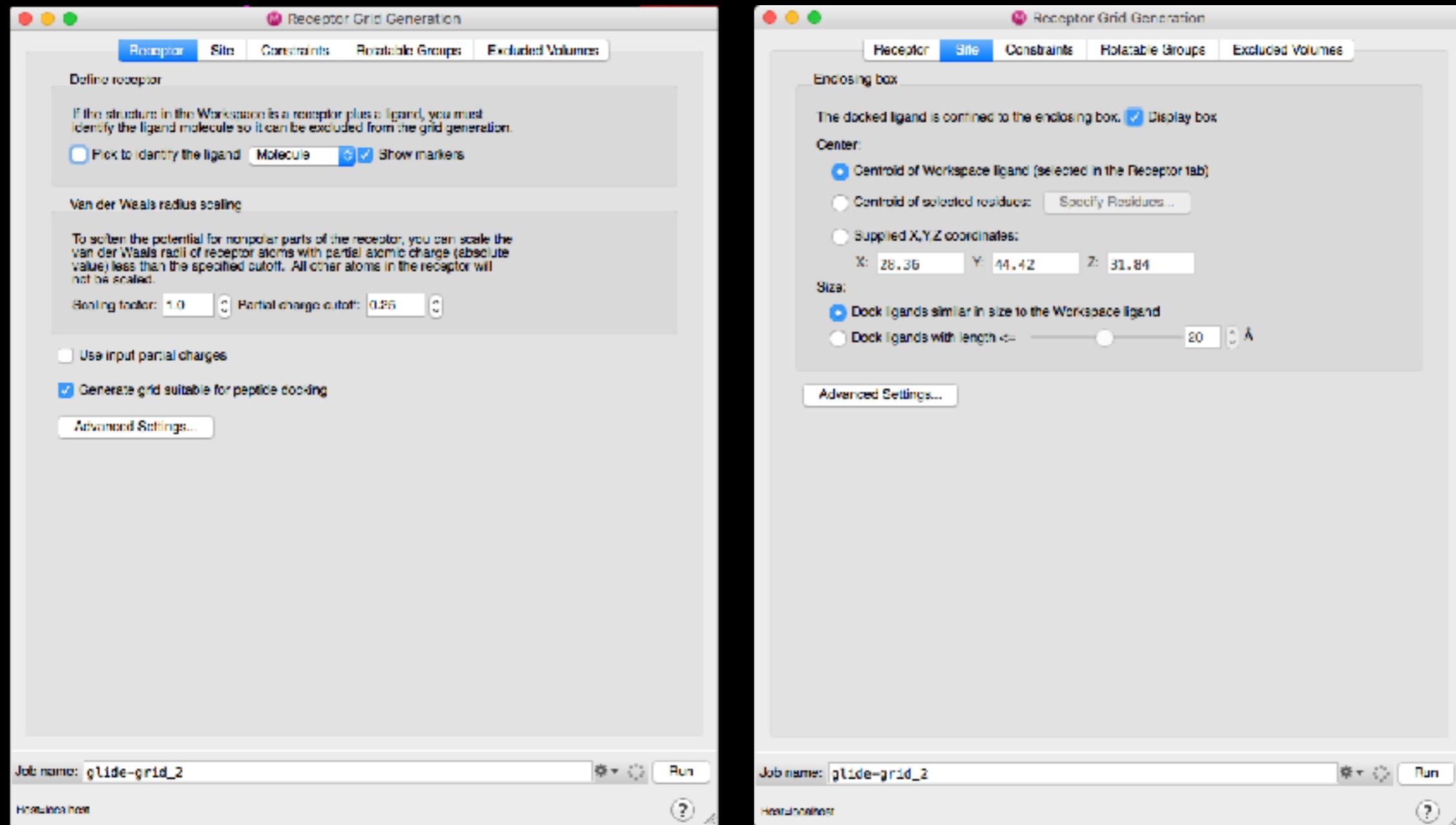


Generate docking grid  
for protein

# COMPONENT 4: GLIDE RECEPTOR GRID GENERATION

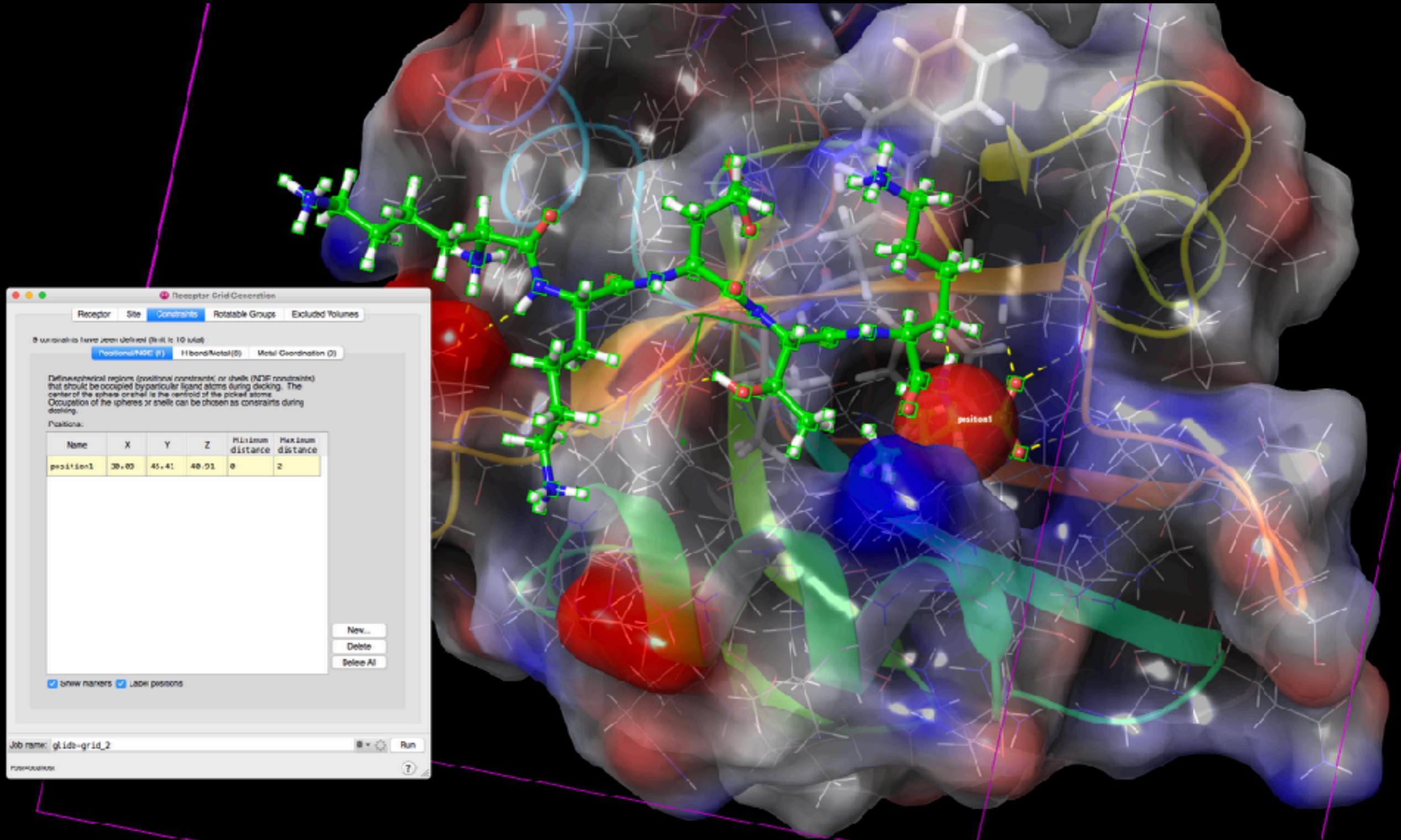


# COMPONENT 4: GLIDE RECEPTOR GRID GENERATION



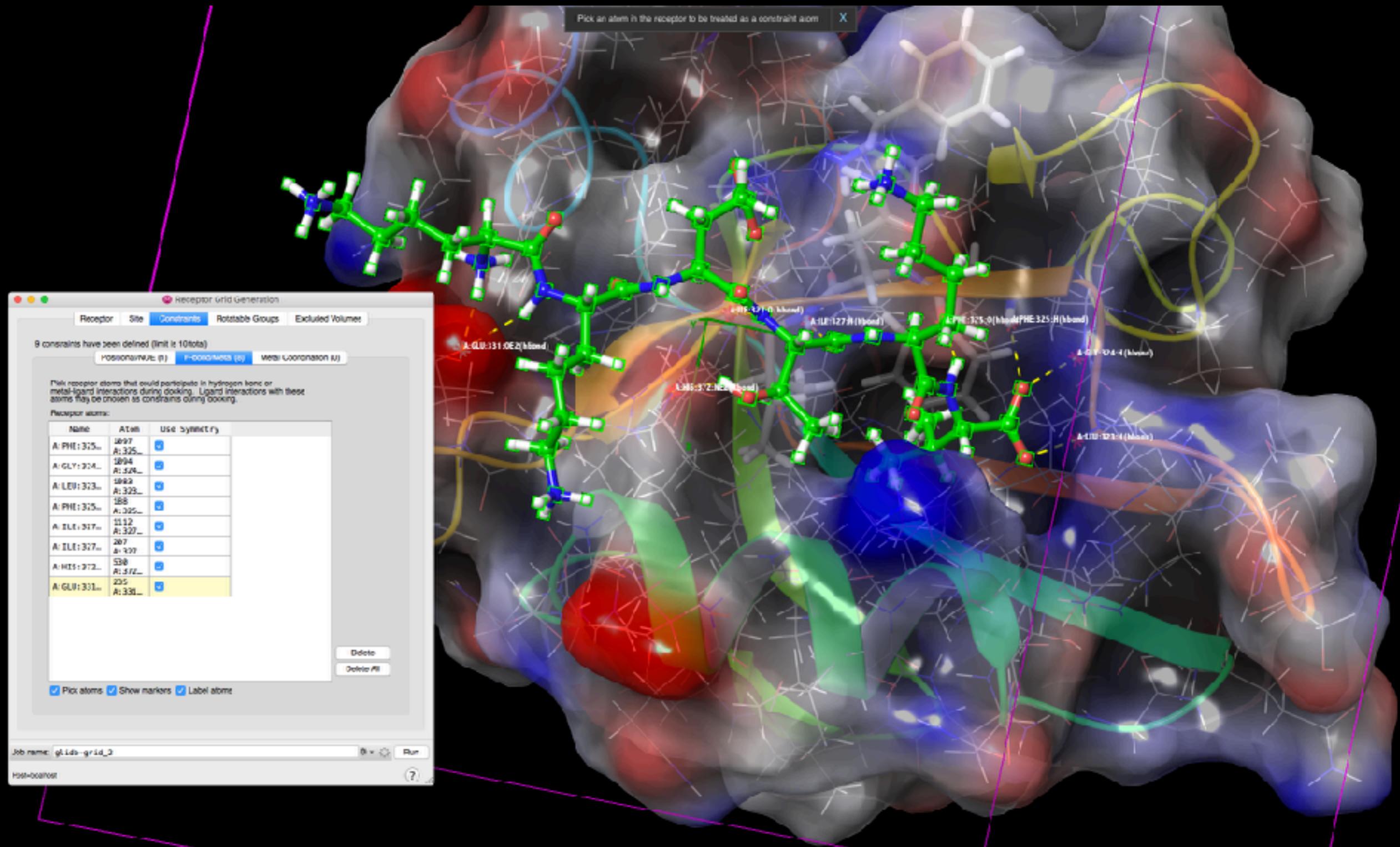
Step 1A: Glide Receptor Grid Generation—Once per protein (with extreme care!)

# COMPONENT 4: GLIDE RECEPTOR GRID GENERATION



Step 1B: Glide Receptor Grid Generation—constraints (positional)

# COMPONENT 4: GLIDE RECEPTOR GRID GENERATION



Step 1C: Glide Receptor Grid Generation—constraints (H-bonds)

# COMPONENT 4: GLIDE RECEPTOR GRID GENERATION

The figure displays two side-by-side windows of the 'Receptor Grid Generation' software.

**Left Window (Rotatable Groups Tab):**

- Title Bar:** Receptor Grid Generation
- Tab Bar:** Receptor, Site, Constraints, Rotatable Groups (selected), Excluded Volumes
- Text:** Select receptor hydroxyl and thiol groups for which to allow rotation:
- Table:** A list of atoms and residues with checkboxes for 'Allow rotation'. One row is highlighted in yellow.

Allow rotation	Atom	Residue
<input type="checkbox"/>	1195 H	A:339 SER
<input type="checkbox"/>	1269 H	A:358 SER
<input type="checkbox"/>	1357 H	A:361 SER
<input type="checkbox"/>	1428 H	A:371 SER
<input type="checkbox"/>	1627 H	A:397 TYR
<input type="checkbox"/>	1632 H	A:398 SER
<input type="checkbox"/>	1676 H	A:404 SER
<input type="checkbox"/>	1752 H	A:414 THR
- Buttons:** Deselect All, Pick groups
- Job Name:** glide-grid\_2
- Host:** localhost
- Run Button:** Run

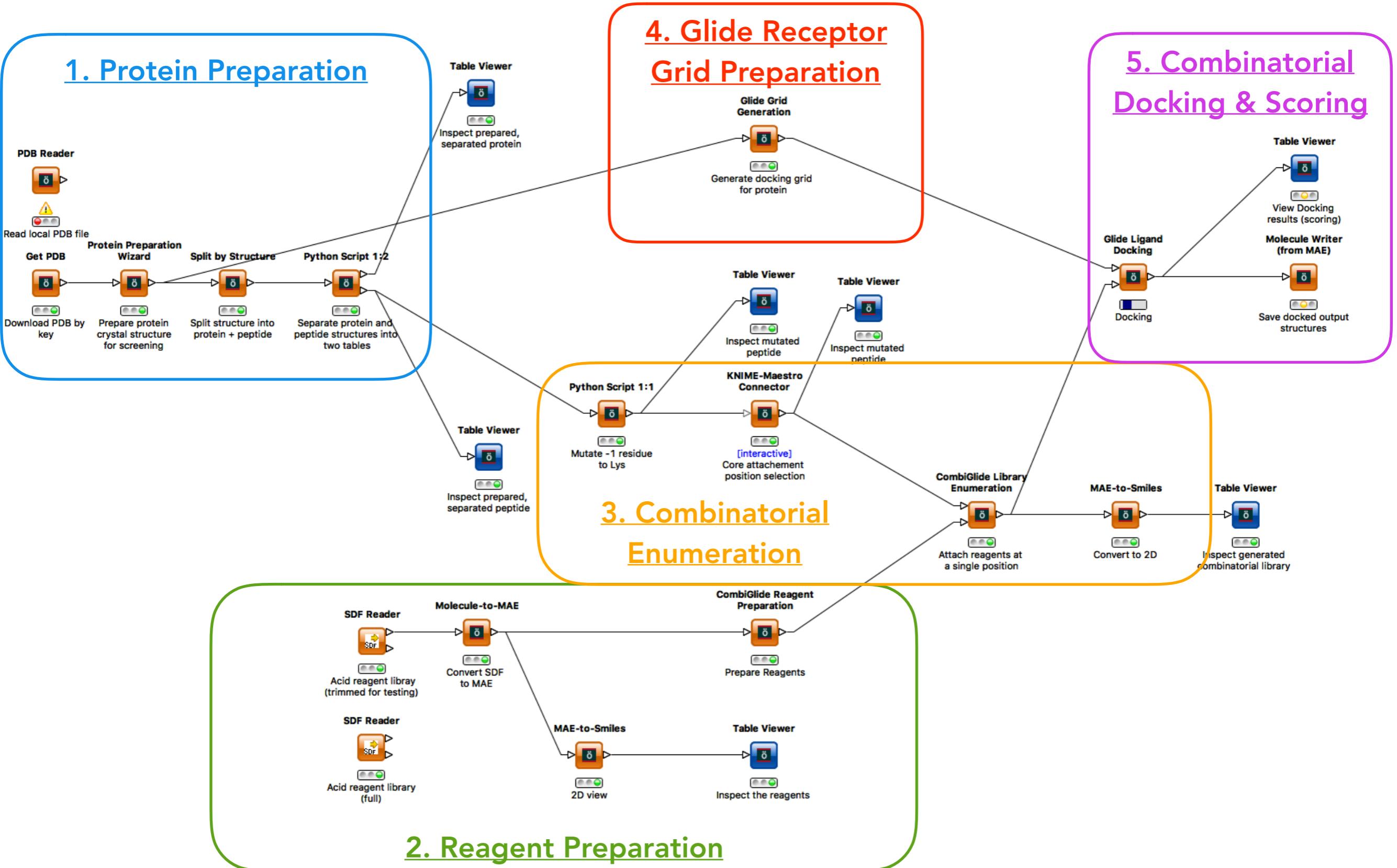
**Right Window (Excluded Volumes Tab):**

- Title Bar:** Receptor Grid Generation
- Tab Bar:** Receptor, Site, Constraints, Rotatable Groups, Excluded Volumes (selected)
- Text:** Define excluded volumes that should not be occupied by any ligand atom during docking. The center of the sphere is the centroid of the picked atoms. Strength of the repulsive potential can be chosen at docking.
- Text:** Excluded volumes:
- Table:** A table for defining excluded volumes.

Name	X	Y	Z	Radius
- Buttons:** New..., Import..., Delete, Delete All
- Checkboxes:** Show markers, Label excluded volumes
- Job Name:** glide-grid\_2
- Host:** localhost
- Run Button:** Run

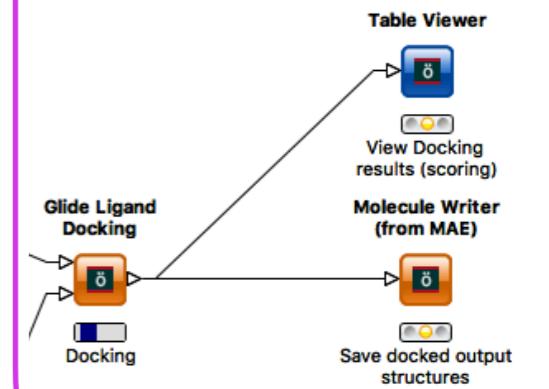
Step 1C: Glide Receptor Grid Generation—Rotatable groups & excluded volumes

# KEY COMPONENTS

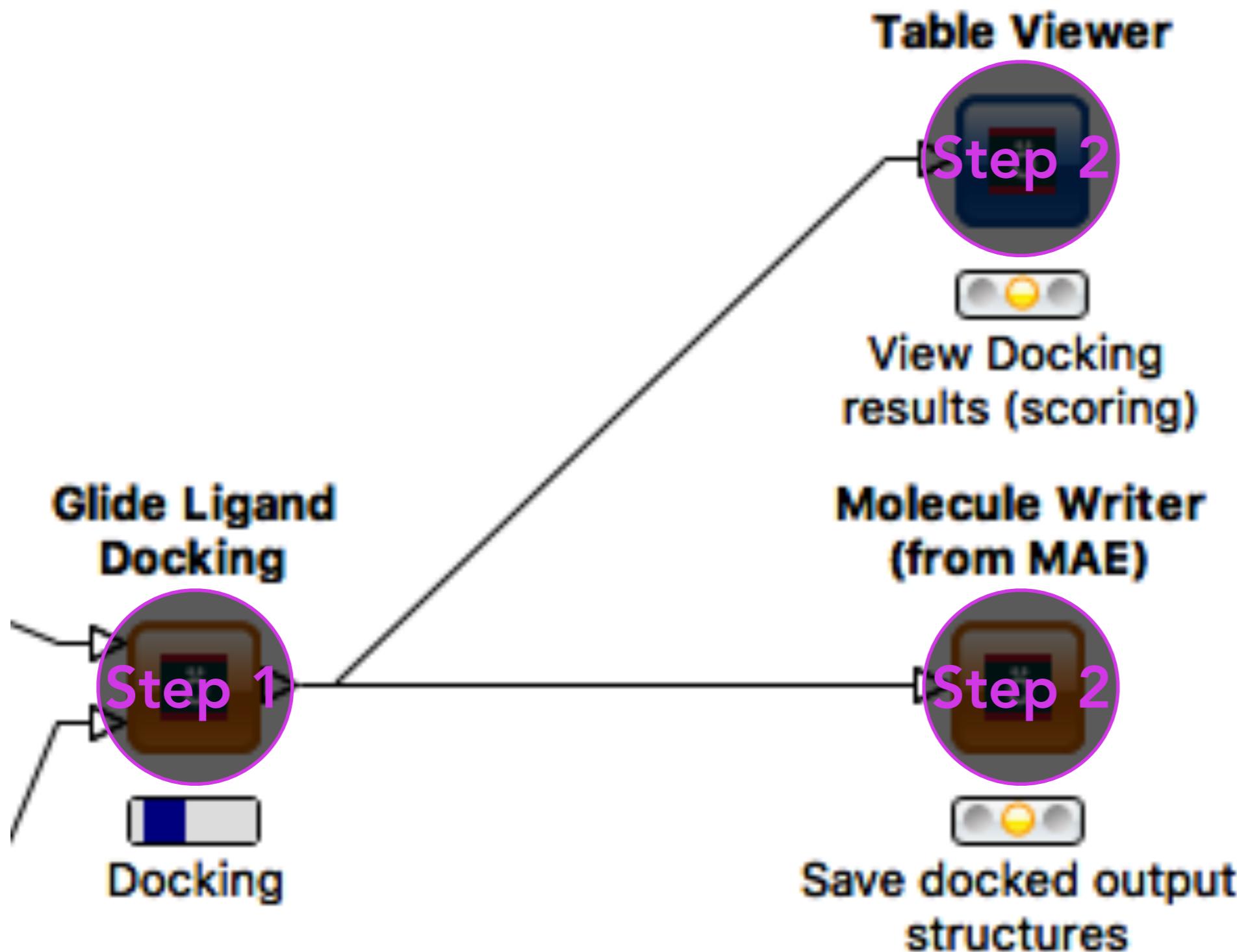


# COMPONENT 5: COMBINATORIAL DOCKING & SCORING

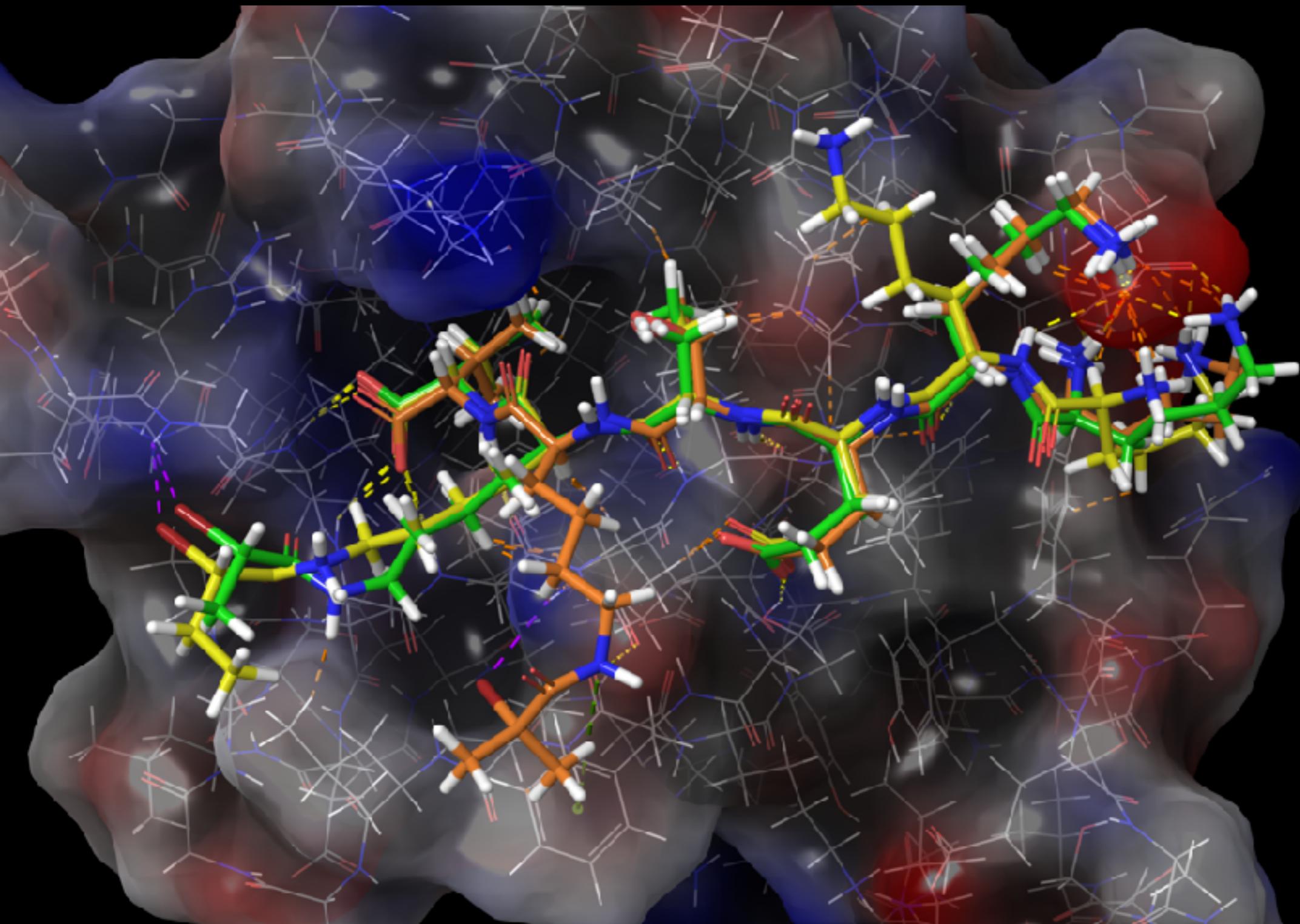
## 5. Combinatorial Docking & Scoring



# COMPONENT 5: COMBINATORIAL DOCKING & SCORING

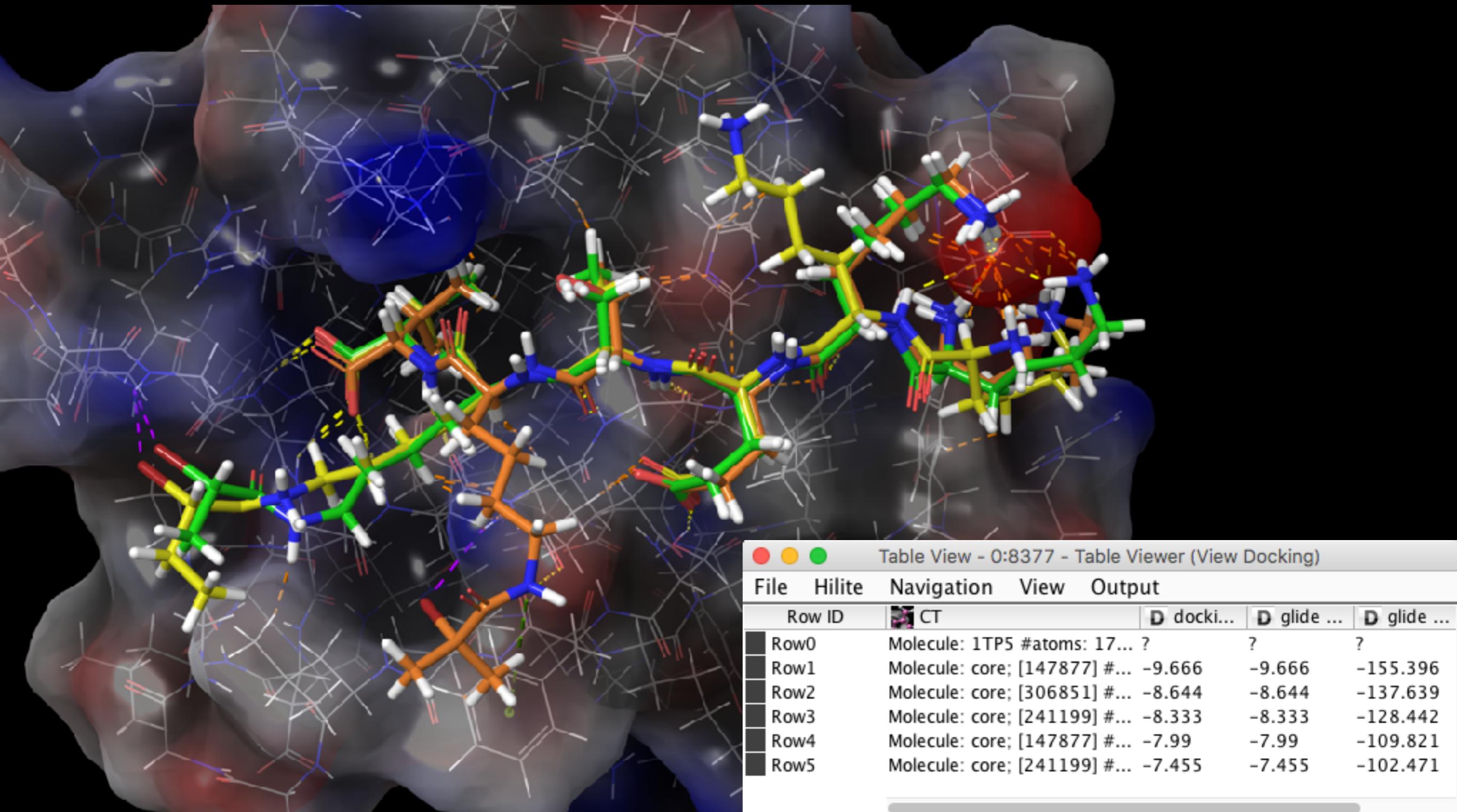


# COMPONENT 5: COMBINATORIAL DOCKING & SCORING



Step 1: Docking & Scoring—Use Glide's HTVS, SP, or XP method

# COMPONENT 5: COMBINATORIAL DOCKING & SCORING



Step 2: Writing and Viewing Results—Save docked ligands + protein and examine docking scores for each compound

# NEXT STEPS

- Integrate with CombiGlide screening directly
  - Manual python integration
  - No intermediary enumeration
  - Principle core conformations
- Verify predictions qualitatively for PSD-95
- Apply to GIPC at Cal
- Tun on Dartmouth Research Computing Cluster

# FURTHER STEPS

- **Refinement of Results:**
  - *Induced Fit Docking*
  - *QM-Polarized Docking / QSite / Jaguar*
  - *WaterMap?*
- **Generation of novel cores:**
  - *Core Hopping*: Ligand/receptor-based scaffold exploration
  - *SiteMap / WaterMap*
  - Structure-based ligand design
    - *Phase*: ligand-based drug design, pharmacophore modeling
  - High throughput small molecule screening

# QUESTION FOR YOU

- How should we deal with peptides different than given in the crystal structure, mutate the crystallized peptide and re-dock with constraints or construct the peptide from scratch and dock?
- Other modifications other than organic acids?
- General limitations on synthesizable modifications?

# QUESTIONS