

Aug 11, 2025

Version 3

How to Assign AMBER Parameters to Desmond-generated System with viparr4 V.3

 Version 1 is forked from [How to Assign CHARMM Parameters to Desmond-generated System with viparr4](#)

DOI

<https://dx.doi.org/10.17504/protocols.io.bp2l6bqwkqge/v3>

Thilo Mast¹, Magd Badaoui¹, Dmitry Lupyan¹

¹Schrodinger Inc

Schroedinger Inc.



Thilo Mast

Schrödinger Inc.

AMBER
to
DESMOND
via
VIPARR

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.bp2l6bqwkqge/v3>

Protocol Citation: Thilo Mast, Magd Badaoui, Dmitry Lupyan 2025. How to Assign AMBER Parameters to Desmond-generated System with viparr4. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.bp2l6bqwkqge/v3> Version created by **Magd Badaoui**

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: August 11, 2025

Last Modified: August 11, 2025

Protocol Integer ID: 224464

Keywords: Molecular Dynamics, MD, Desmond, CHARMM, Amber, Force Field, Simulations, custom amber parameters for the ligand, generated amber parameter, amber parameters to desmond, amber parameter, custom amber parameter, charmm parameters to desmond, systems for molecular dynamics simulation, charmm parameter, molecular dynamics simulation, opIs4 force field, system with viparr4, parametrization step, party force fields like amber, viparr utility, amber, ligand, schrödinger software suite, differs in the parametrization step, viparr4, ligand complex,

Disclaimer

The converter tool described in this document is not supported by either Schrödinger Inc. or D.E. Shaw Research. For help please consult other Desmond users via **Google Desmond Group**.

Abstract

This tutorial partially copies steps from **How to Assign CHARMM Parameters to Desmond-generated System with viparr4** but differs in the parametrization steps.

You can use the Schrödinger software suite to prepare systems for molecular dynamics simulations, however, only the OPLS _2005 and OPLS4 force fields can be automatically assigned. This tutorial will show you how to use Desmond with third-party force fields like AMBER, using the Viparr utility from D.E. Shaw Research.

You will prepare a protein-ligand complex, generate custom AMBER parameters for the ligand, and use the Viparr utility to convert the generated AMBER parameters into a viparr-formatted template, that can be used for simulations.

You can find the input files for this tutorial here:



amber_to_desmond.tgz



Materials

- Schrödinger Software version 2020-4 or newer installed on a [supported platform](#) with Linux
- Bash shell
- Optional: git

Note

In order to run MDs with Desmond, you'll also need a supported Nvidia GPU.

Before start

You should have a working installation of the Schrödinger Software suite version 2025-1 or newer on a supported version of a supported Linux distribution. See [Supported Platforms](#) for more information.

Set the environment variable SCHRODINGER to the installation directory of the Schrödinger software. For example, for a bash shell, enter the following:

```
export SCHRODINGER=/path/to/schrodinger2025-1
```



Setting up a Schrödinger Python Environment

1 Install AmberTools + conda

AmberTools is needed for deriving force field parameters for the ligand. For installation of AmberTools + conda please follow the instructions on <https://ambermd.org/GetAmber.php#ambertools> . This tutorial used the installation procedure via conda.

2 Set up a Python Virtual Environment

In order to avoid compatibility issues with the Python modules, and ensure interoperability with Schrödinger's Python modules, use the Schrödinger software to create a new virtual environment.

2.1 Initialize a new Python virtual environment using the Schrödinger software:

```
%> $SCHRODINGER/run schrodinger_virtualenv.py schrodinger.ve
```

2.2 Activate the virtual environment:

```
%> source schrodinger.ve/bin/activate
```

2.3 Upgrade pip:

```
[schrodinger.ve] %> pip install --upgrade pip
```

3 Install the Viparr and Msys Utilities



Viparr is a utility developed by D.E. Shaw Research to assign force field parameters to systems, which can then be used to perform MD simulations with Desmond.

3.1 Download these Python packages:

-  msys-1.7.359-cp311-cp311-manylin... 45.1MB
-  viparr-4.7.35-cp311-cp311-manylin... 36.4MB



These packages are distributed in the Python 'wheel' format, compatible with 'manylinux2014' Linux distributions.

3.2 Install the packages into the virtual environment

```
[schrodinger.ve] %> pip install msys-1.7.359-cp311-cp311-  
manylinux_2_17_x86_64.manylinux2014_x86_64.whl  
[schrodinger.ve] %> pip install viparr-4.7.35-cp311-cp311-  
manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

3.3 Clone the public Viparr parameters from D.E. Shaw Research's GitHub repository:

```
[schrodinger.ve] %> git clone  
git://github.com/DEShawResearch/viparr-ffpublic.git
```

Alternatively, you can download the repository as ZIP archive: [**viparr-ffpublic**](#)

3.4 Set the VIPARR_FFPATH environment variable

```
[schrodinger.ve] %> export VIPARR_FFPATH=`pwd`/viparr-ffpublic/ff
```

Re-activate your Virtual Environment

- 4 Steps 1 to 3 cover the one-time installation and creation of your virtual environment with **Viparr4** installed. In the future, to use Viparr4 again, you will only need to **activate the environment** and **export the library** containing the Viparr public force field.

4.1 First, navigate to the directory where you created the virtual environment using the

```
cd {path_to_schrodinger.ve}
```

4.2 Once you are in the correct directory, activate the environment:

```
source schrodinger.ve/bin/activate
```

- 4.3 After activating your virtual environment, you need to tell **Viparr** where to find the public force field files. To do this, export the **VIPARR_FFPATH** environment variable. In the command below, replace {path_to_viparr-ffpublic} with the absolute path to the directory where you cloned the **viparr-ffpublic** repository.

```
export VIPARR_FFPATH={path_to_viparr-ffpublic}/viparr-ffpublic/ff
```

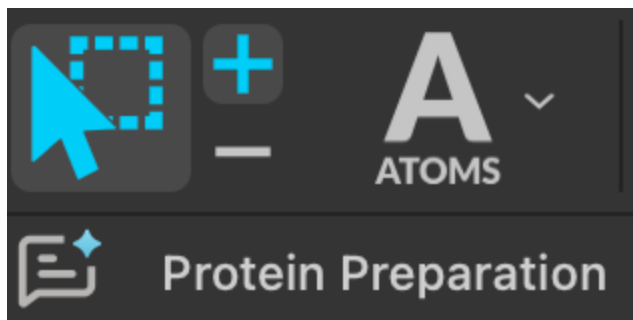
Preparing the Protein-Ligand Complex for Simulation

- 5 First, we download a structure from the protein data base (PDB) and prepare the protein-ligand complex. This tutorial uses a PDB structure for the human adipocyte lipid-binding protein FABP4 in complex with (S)-ibuprofen (PDB ID: [3p6h](#)). We build the system with Schrödinger's Maestro and Desmond tools. The default force field used by the Schrödinger software is OPLS4. After building the system for MD simulations, we can reassign the forcefield using Viparr.

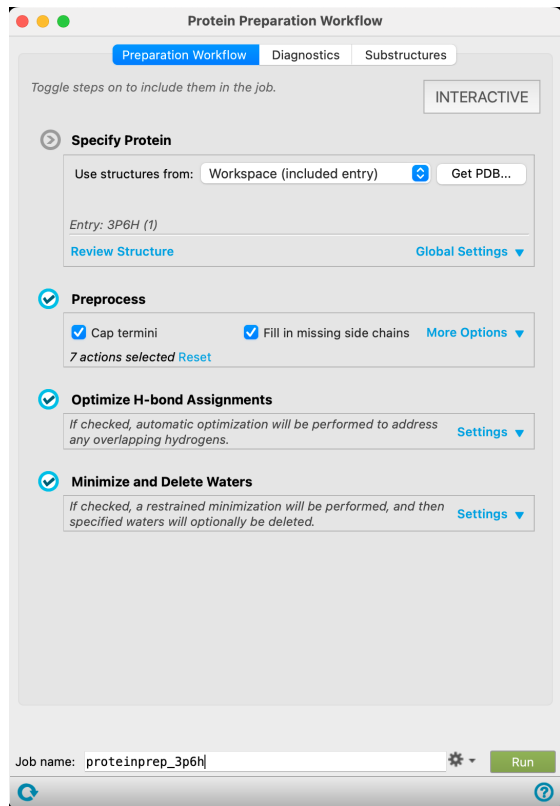
To prepare the system for MD, open Maestro and follow these steps:

5.1 Prepare the Protein-Ligand Complex

In Maestro, go to **Tasks** → **Protein Preparation and Refinement** → **Protein Preparation Wizard** or click **Protein Preparation** in the toolbar.



In the Protein Preparation Wizard, enter '3p6h' in the **PDB** text box, and click **Import** to download the structure from the PDB database and import it into Maestro.



The screenshot shows the 'Protein Preparation Workflow' window with the following settings:

- Preparation Workflow** (selected tab)
- Specify Protein**:
 - Use structures from: Workspace (included entry) [dropdown]
 - Get PDB... [button]
 - Entry: 3P6H (1)
 - Review Structure [link]
 - Global Settings [dropdown]
- Preprocess** (checked):
 - Cap termini [checked]
 - Fill in missing side chains [checked]
 - More Options [dropdown]
 - 7 actions selected [link]
 - Reset [link]
- Optimize H-bond Assignments** (checked):
 - If checked, automatic optimization will be performed to address any overlapping hydrogens. [text]
 - Settings [dropdown]
- Minimize and Delete Waters** (checked):
 - If checked, a restrained minimization will be performed, and then specified waters will optionally be deleted. [text]
 - Settings [dropdown]
- Job name**: proteinprep_3p6h [text input]
- Run** [button]

After the PDB has been imported into the 3D workspace, preprocess the protein and the ligand. In the Protein Preparation Wizard, make sure the following options are selected:

☐ Align to: ☒ First selected entry ☐ PDB:

☒ Assign bond orders:

☒ Replace hydrogens

Create: ☒ Zero-order bonds to metals
☒ Disulfide bonds

☒ Identify protein features:
Antibody scheme:
☐ Renumber residues to match scheme

☐ Add terminal oxygens to protein chains

☐ Convert selenomethionines to methionines

☐ Delete waters beyond hets: Å

☐ Include peptides when capping termini

☐ Fill in missing loops (using Prime) ⓘ

☒ Generate het states (with Epik): pH: +/-

Max states to process automatically

- Assign bond orders
- Add hydrogens
- Create disulfide bonds
- Cap termini
- Fill in missing side chains using Prime (requires a Prime license)

Click **Run**.

The Protein Preparation Wizard runs structural sanity checks and displays potential issues in a warning. In this tutorial, you can safely ignore these warnings and click OK to close the warning. In real applications, you should pay close attention to these warnings and resolve issues that may lead to problems during the MD simulations.

Note

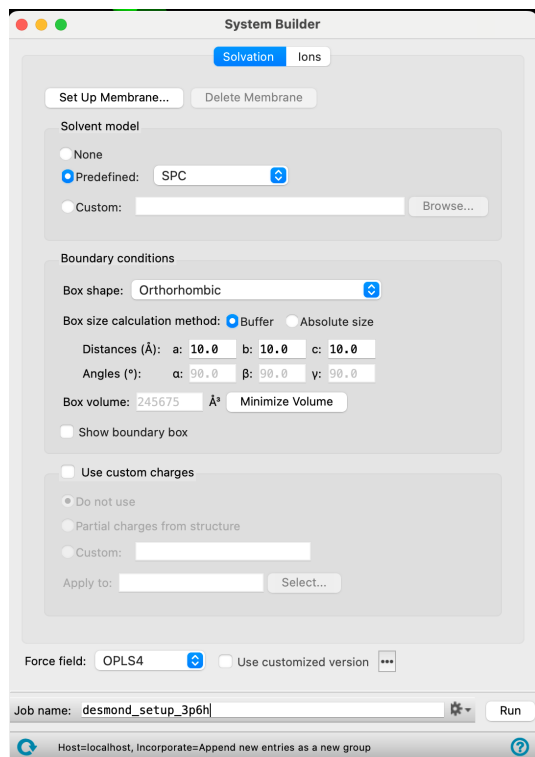
For the 3p6h structure, the Protein Preparation Wizard shows missing atoms on the terminal residues. Since we cap the terminals, this is not an issue. Further, overlapping atoms and alternative residue positions can be ignored for this tutorial.

After the Protein Preparation Wizard finishes, a protein-ligand complex is added to Maestro's Workspace where formal charges, protonation states and bond orders are assigned and hydrogen atoms are added to the protein, ligand and crystallographic water molecules.

5.2 Solvate and Neutralize the Protein-Ligand Complex

Now that we have a protein-ligand complex loaded in the Workspace, we can solvate and neutralize the complex to prepare it for MD simulations.

In Maestro, go to **Tasks** → **Desmond** → **System Builder** to open the System Builder.



Click **Run** to build the system with the default settings. By default, a water buffer of 10 Å and neutralizing counterions are added to the protein-ligand complex. The System Builder also assigns OPLS4 force field parameters to the protein-ligand atoms using SPC as the model for water.

After the System Builder finishes, a solvated system is added to Maestro's Workspace. The solvated system is also saved to a file **desmond_setup_1-out.cms**.

Generating Custom AMBER parameters

- 6 To simulate a molecule that is not covered by the standard AMBER parameter set, you first need to obtain parameter files for that molecule. You can obtain parameters from one of these sources:

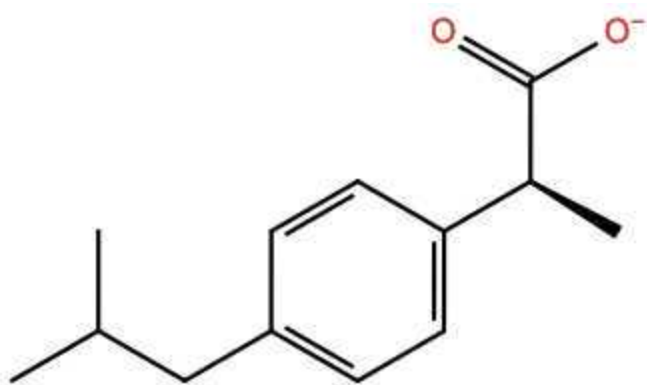
- Deriving parameters via AmberTools antechamber as shown in these tutorials: [AM1-bcc](#) or [RESP fitting](#)
- Use [q4md-forcefieldtools / PyRED server](#)
- Search in parameter databases [like this one](#)
- Search the AMBER mailing list to see if anyone else is willing to share their parameters: [AMBER mailing list](#)
- Adopt parameters from the literature
- ...

You can use any of the above methods to generate AMBER-compatible parameters.

In this tutorial, we use AmberTools' antechamber to generate parameter files.

6.1 Export the Ligand as SDF File

The 3p6h complex contains a ligand: *Ibuprofen* (IBP). This ligand is not part of the standard AMBER parameter set so you have to derive its parameters via antechamber (or from one of the above mentioned options).



A 2D representation of *Ibuprofen*

We provide the ligand structure to antechamber as **.sdf** file.

To prepare a **.sdf** file with the prepared ligand structure, follow these steps:

1. Select the ligand atoms in Maestro's Workspace, right-click the selected atoms and select **Create New Entry by** → **Copying Selected Atoms**. This creates a new entry in Maestro only containing the ligand atoms.
2. Select the ligand entry in the **Entry List** on the left side of Maestro, right-click and select **Export** → **Structures**
3. Enter *ibp.sdf* as file name.

- 6.2 (optional) If you have installed AmberTools via the conda option (see Step 1.) it is now the right time to activate the AmberTools conda environment via:w

```
[schrodinger.ve] %> conda activate AmberTools21
```

6.3 Generate Custom AMBER Parameters with antechamber

We now derive partial charges for the IBP ligand by running:

```
[AmberTools21] [schrodinger.ve] %> antechamber -fi mdl -i ibp.sdf  
-fo prepi -o IBP.prepi -c bcc -at amber -nc -1 -rn IBP
```

with

- the *-fi* flag we define the format of the input file (here: MDL SDF format).
- the *-i* flag we provide the name of the ligands .sdf file.
- the *-fo* flag we define the format of the output file (here: antechambers .prepi parameter format).
- the *-c* flag we define the parametrization method which is being used for deriving point charges (here: semi-empirical AM1-bcc).
- the *-nc* flag we define the total charge of the molecule.
- the *-rn* flag we provide a residue name for our ligand.

Running the above command will roughly take one minute.

After the job has completed successfully the parameter definition file **IBP.prepi** should have been created and the **sqm.out** output file should list at the end of the file:

```
----- Calculation Completed -----
```

- 6.4 Run antechambers parmchk2 program to check which parameters are still missing. The parmchk2 program tries to make educated guesses for them and writes the information to the .frcmmod file:



```
[AmberTools21] [schrodinger.ve] %> parmchk2 -i IBP.prepi -f prepi  
-o IBP.frcmod
```

6.5 Convert the AMBER Parameters to a Viparr Template

The parameters need to be converted into a format that you can use with Desmond, enter the following command to generate a viparr-formatted template:

```
[AmberTools21] [schrodinger.ve] %> viparr-convert-amber -m ligand  
-p IBP.frcmod -t IBP.prepi IBP/
```

After the program finishes, a new directory **IBP** is created, which contains the viparr-formatted AMBER template for Ibuprofen.

Assigning AMBER Force Fields to the Molecular System

7 Now, we can reassign the AMBER force field to the CMS file.

The viparr utility comes with standard sets of protein parameters which we can use for the protein.

For the Ibuprofen ligand, we can use the viparr template in the directory IBP.

7.1 Run Viparr

To reassign the force field to AMBER, enter the following command:

```
[AmberTools21] [schrodinger.ve] %> viparr desmond_setup_1-out.cms  
amber-start.cms -f aa.amber.ff19SB -f water.tip3p -f  
ions.amber1jc.tip3p -d IBP
```

```
Importing forcefield from ../viparr-ffpublic/ff/aa.amber.ff19SB  
Importing forcefield from ../viparr-ffpublic/ff/water.tip3p  
Importing forcefield from ../viparr-  
ffpublic/ff/ions.amber1jc.tip3p  
Importing forcefield from IBP  
Importing structure from desmond_setup_1-out.cms  
WARNING: will NOT fix prochiral atom names because rename_atoms or  
rename_residues was not enabled  
Exporting parametrized system to amber-start.cms  
VIPARR exited successfully
```



With

-f *aa.amber.ff19SB* the AMBER ff19SB parameters are assigned to the protein part

-f *water.tip3p* the TIP3P 3-site water model will be used as water model

-f *ions.amber1jc.tip3p* Ambers monovalent ion parameters will be used

-d IBP we provide the viparr template for the IBP ligand

Note

For a list of further available AMBER parameters in viparr, have a look at viparrs help text:

```
%> viparr -h
```

After viparr finishes, a new file: **amber-start.cms** is created which contains the structure and AMBER force field information.

7.2 Clean Properties and Prepare for MD

To prepare the converted system for MD simulations, certain properties need to be removed from the CMS file and the full-system structure needs to be prepared. Run the

Python program:  `gen_fsys.py`

```
%> $SCHRODINGER/run gen_fsys.py amber-start.cms
```

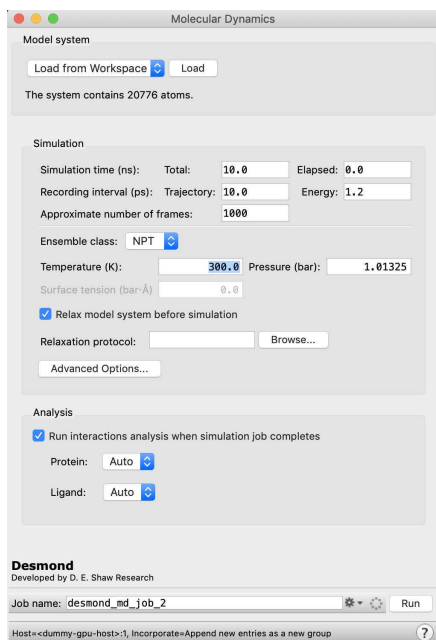
This script will overwrite the file **amber-start.cms**. The system is now ready for simulation.

Running an MD Simulation with Desmond

- 8 Now that the system is prepared with AMBER parameters, Desmond can minimize, equilibrate, and run an MD simulation of the system. For further information check the *Desmond User Manual*.

8.1 Set up an MD Simulation

In Maestro, go to **Tasks** → **Desmond** → **Molecular Dynamics** to open the **Molecular Dynamics** panel.



Molecular Dynamics Panel

In the **Model system** section, choose Import from a file, and click Browse to locate and load **amber-start.cms**

1. In the **Model system** section, select **Import from file**, browse to the file **amber-start.cms**, and click **Load**.
2. *Optional:* Adjust simulation parameters, such as simulation time.
3. Ensure the **Relax model system before simulation** checkbox is selected.
4. *Optional:* Enter a job name, click the gear icon to change job settings, or click the ? icon to open the Help for more information.
5. Click **Run** to start the MD simulation.

8.2 Analyze the Results

In Maestro, go to **Tasks** → **Desmond** → **Simulation Interactions Diagram (SID)** and 'Load ...' in the .eaf file.

