Desmond_2022-4[illegible] HPC[illegible] MD[illegible]

[illegible] Amber[illegible] Gromacs[illegible] Desmond[illegible]

[illegible]

[illegible] Amber[illegible] Gromacs[illegible] GUI [illegible] Desmond[illegible] [illegible] **Wang-Lin**[illegible] **Schrödinger** [illegible] [illegible] [illegible] **Schrödinger**[illegible]

[illegible] **Amber**[illegible] AmberTools[illegible] Amber [illegible] CPU[illegible] GPU[illegible] Amber [illegible]

[illegible] **Gromacs** [illegible] http://jerkwin.github.io/[illegible] B[illegible] http://bbs.keinsci.com/forum.php[illegible] [illegible] http://www.mdtutorials.com/[illegible]

[illegible] **Desmond**[illegible] “[illegible] ”Maestro[illegible] D.E.Shaw Research [illegible] MD[illegible] Schrödinger[illegible] GUI[illegible] D.E.Shaw[illegible]

I□□□□□□□□□□□□□□□□□□□ Anton2□□□□□□□□□□□ Gromacs

□□□□□□□□□□□□□□□□□□□□□□□□□□□ D.E.Shaw Research□□□□ □□□□□

□□□□□□ **Maestro□□□** **Desmond□□** □

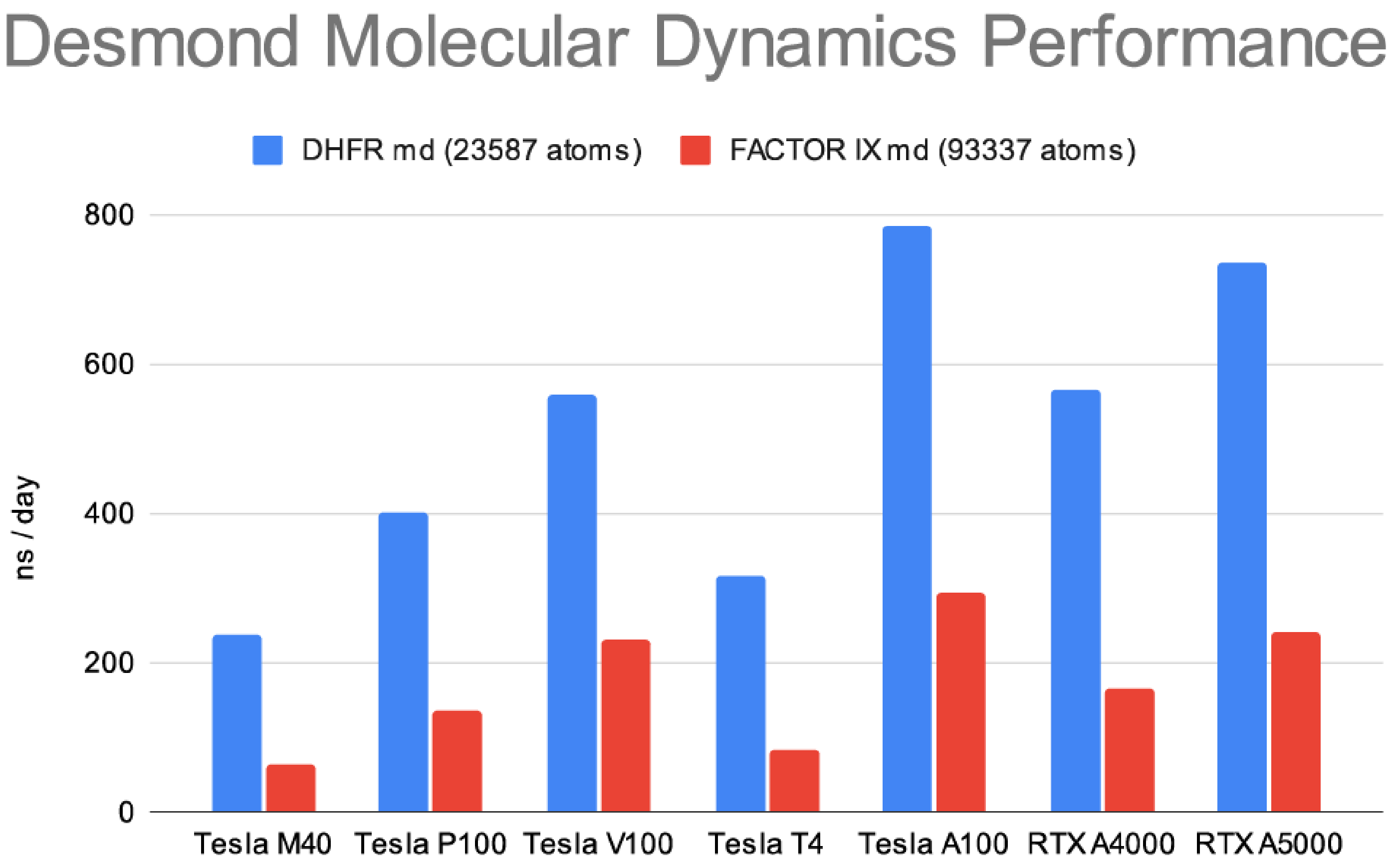


Figure 1: *Molecular Dynamics run using default parameters in Schrödinger Suite 2023-1.*

□□ Desmonds□□□□

□ D.E.Shaw Research□□□□□□□□□□□□□□ HPC□□□□ home

□□□□□□□□□□□□□□□□□□□□□□□□□□□ pwd□□□□□□□□□□□

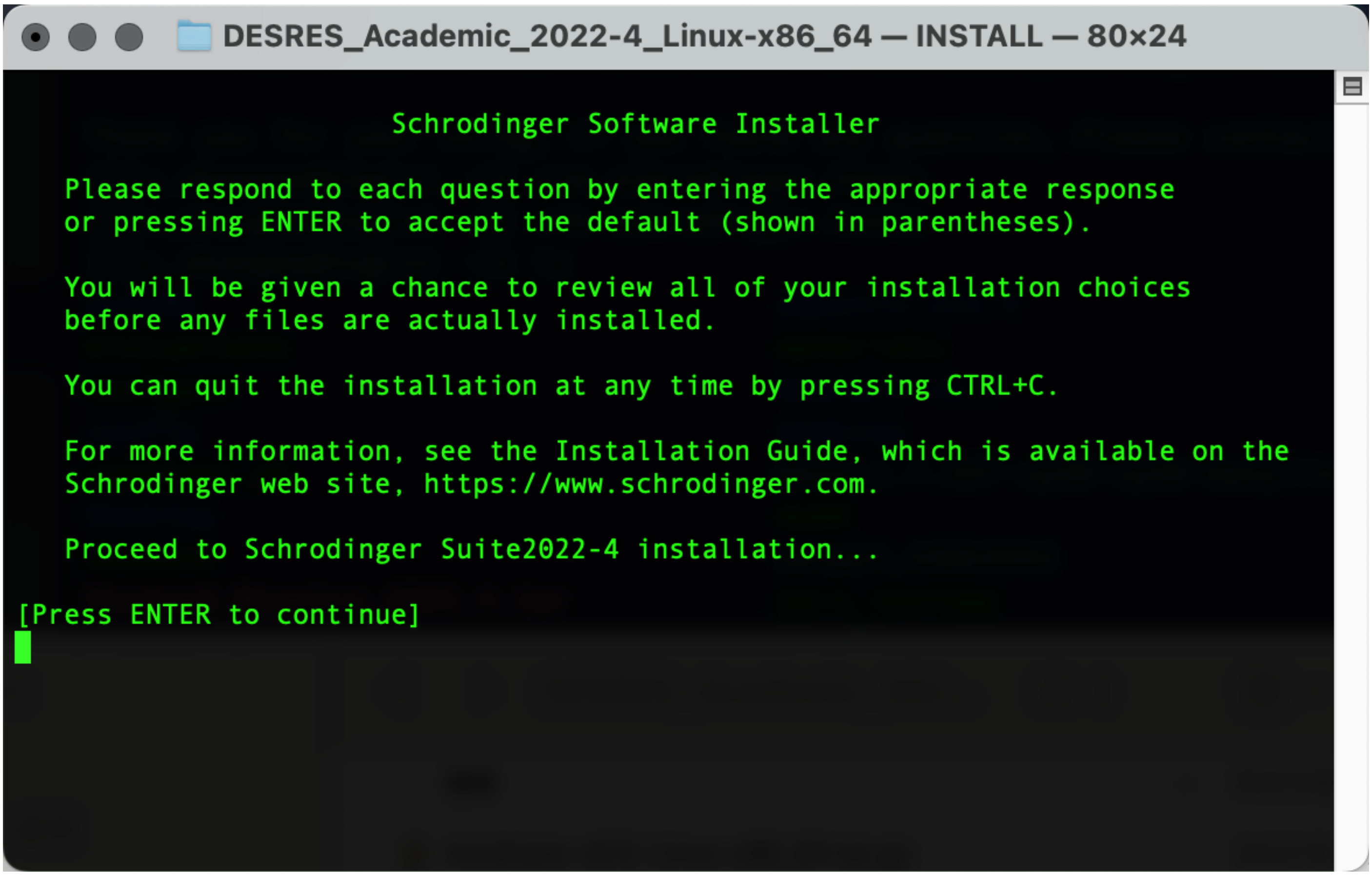
/fsa/home/ljx_zhangzw/software□□

```
mkdir software
pwd
```

□□□□□□□□□□□□□□□ Enter□□□□□□□

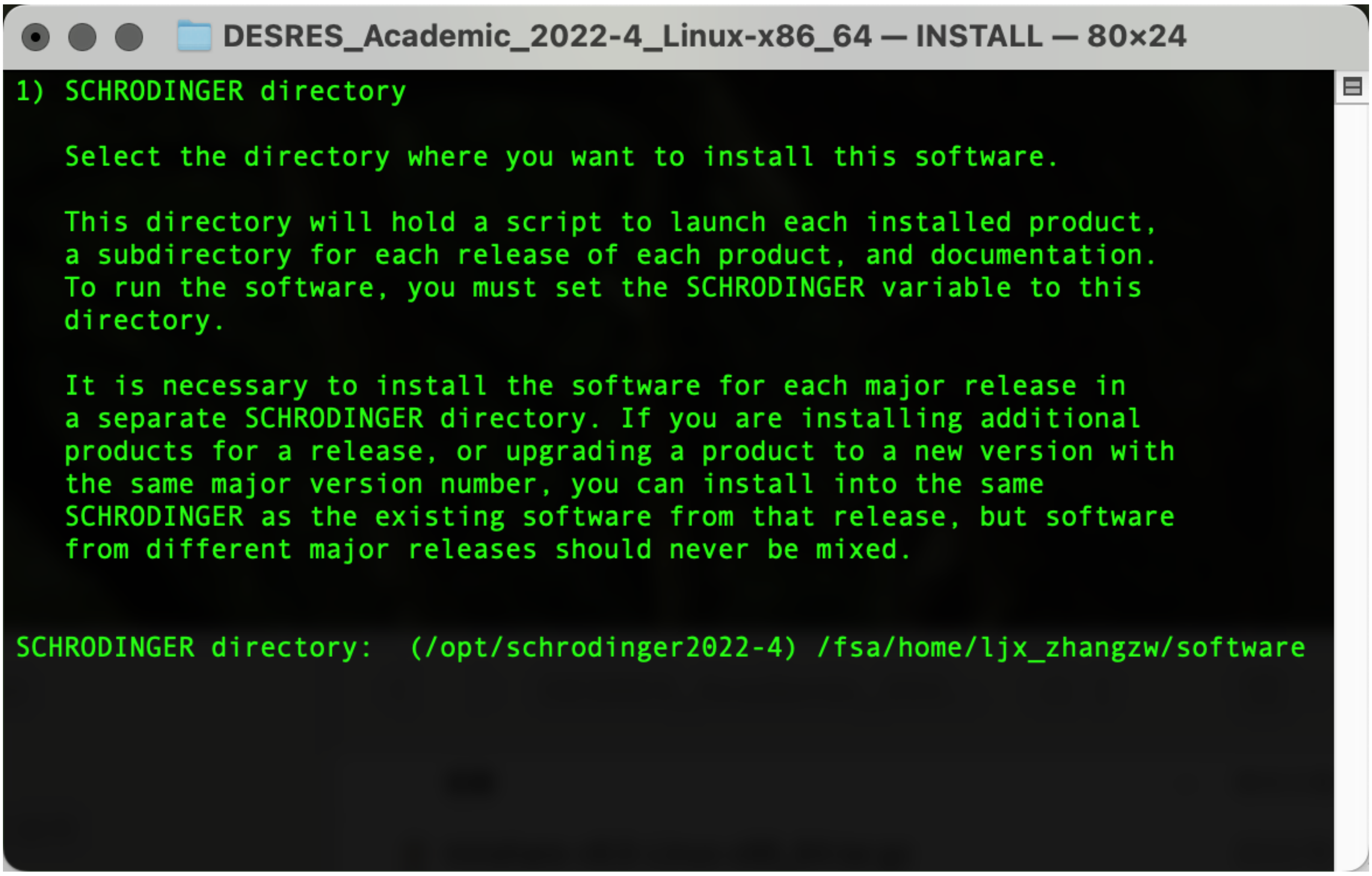
```
tar -xvf Desmond_Maestro_2022.4.tar
cd DESRES_Academic_2022-4_Linux-x86_64
```

```
./INSTALL
```



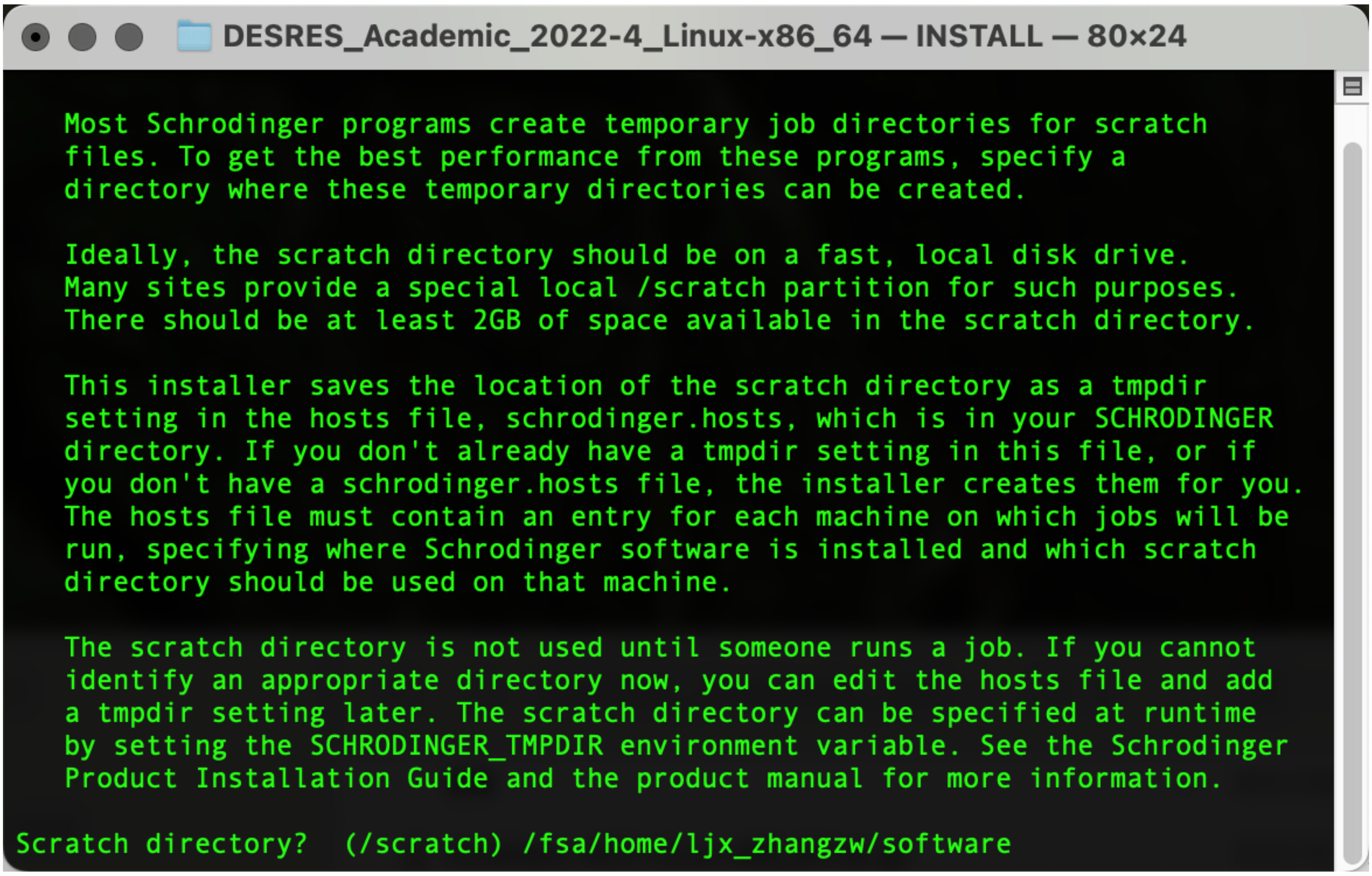
```
                      Schrodinger Software Installer

   Please respond to each question by entering the appropriate response
   or pressing ENTER to accept the default (shown in parentheses).

   You will be given a chance to review all of your installation choices
   before any files are actually installed.

   You can quit the installation at any time by pressing CTRL+C.

   For more information, see the Installation Guide, which is available on the
   Schrodinger web site, https://www.schrodinger.com.

   Proceed to Schrodinger Suite2022-4 installation...

[Press ENTER to continue]
```

[illegible] pwd[illegible] Enter[illegible]



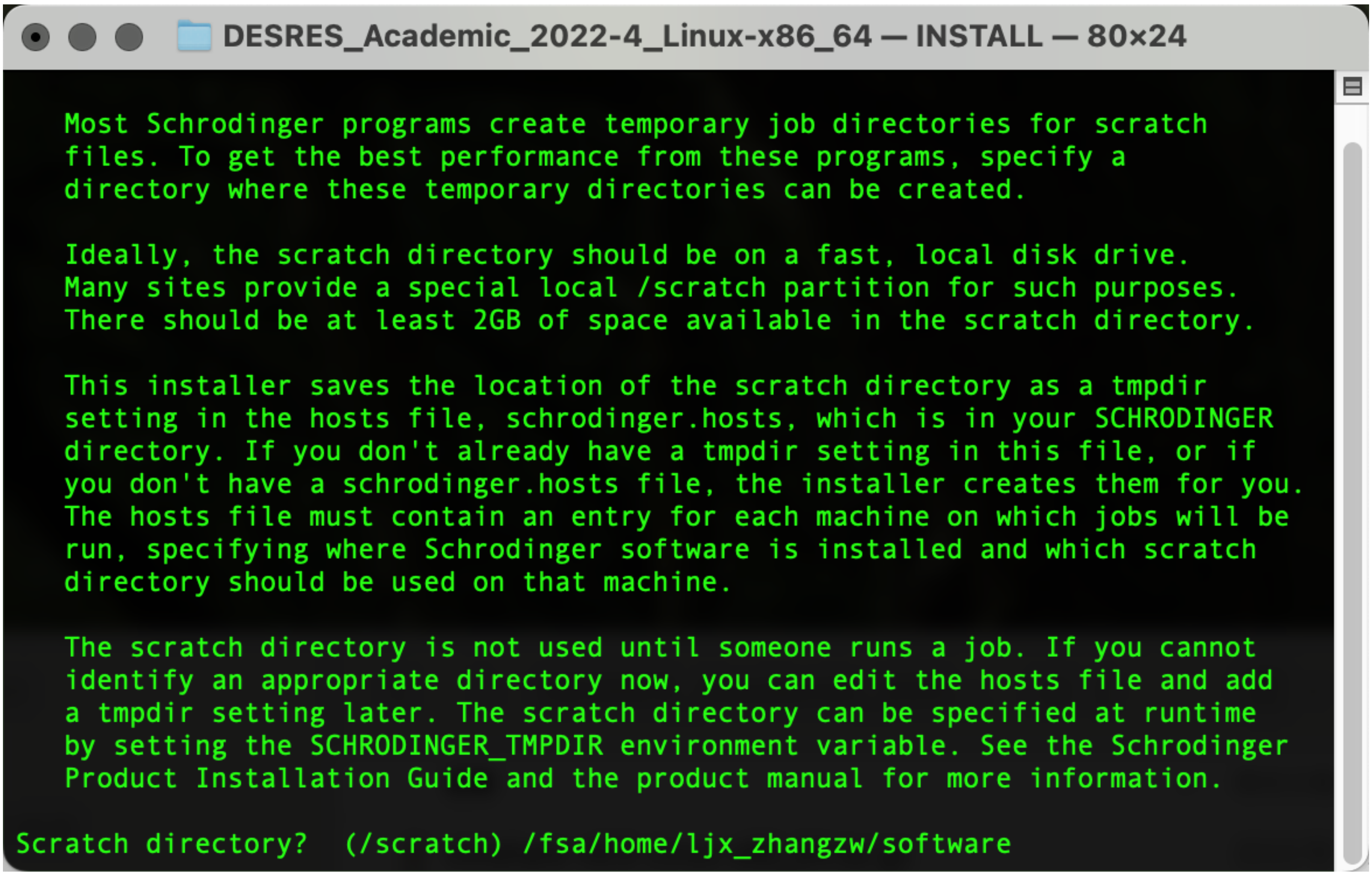
```
1) SCHRODINGER directory

   Select the directory where you want to install this software.

   This directory will hold a script to launch each installed product,
   a subdirectory for each release of each product, and documentation.
   To run the software, you must set the SCHRODINGER variable to this
   directory.

   It is necessary to install the software for each major release in
   a separate SCHRODINGER directory. If you are installing additional
   products for a release, or upgrading a product to a new version with
   the same major version number, you can install into the same
   SCHRODINGER as the existing software from that release, but software
   from different major releases should never be mixed.


SCHRODINGER directory:  (/opt/schrodinger2022-4) /fsa/home/ljx_zhangzw/software
```

□□□□□□□□□　　　　pwd□□□□□□□　　　　Enter□□□□□□□□



```
   Most Schrodinger programs create temporary job directories for scratch
   files. To get the best performance from these programs, specify a
   directory where these temporary directories can be created.

   Ideally, the scratch directory should be on a fast, local disk drive.
   Many sites provide a special local /scratch partition for such purposes.
   There should be at least 2GB of space available in the scratch directory.

   This installer saves the location of the scratch directory as a tmpdir
   setting in the hosts file, schrodinger.hosts, which is in your SCHRODINGER
   directory. If you don't already have a tmpdir setting in this file, or if
   you don't have a schrodinger.hosts file, the installer creates them for you.
   The hosts file must contain an entry for each machine on which jobs will be
   run, specifying where Schrodinger software is installed and which scratch
   directory should be used on that machine.

   The scratch directory is not used until someone runs a job. If you cannot
   identify an appropriate directory now, you can edit the hosts file and add
   a tmpdir setting later. The scratch directory can be specified at runtime
   by setting the SCHRODINGER_TMPDIR environment variable. See the Schrodinger
   Product Installation Guide and the product manual for more information.

Scratch directory?  (/scratch) /fsa/home/ljx_zhangzw/software
```

[illegible] y[illegible] Enter[illegible]



DESRES_Academic_2022-4_Linux-x86_64 — INSTALL — 80×24

```
   Most Schrodinger programs create temporary job directories for scratch
   files. To get the best performance from these programs, specify a
   directory where these temporary directories can be created.

   Ideally, the scratch directory should be on a fast, local disk drive.
   Many sites provide a special local /scratch partition for such purposes.
   There should be at least 2GB of space available in the scratch directory.

   This installer saves the location of the scratch directory as a tmpdir
   setting in the hosts file, schrodinger.hosts, which is in your SCHRODINGER
   directory. If you don't already have a tmpdir setting in this file, or if
   you don't have a schrodinger.hosts file, the installer creates them for you.
   The hosts file must contain an entry for each machine on which jobs will be
   run, specifying where Schrodinger software is installed and which scratch
   directory should be used on that machine.

   The scratch directory is not used until someone runs a job. If you cannot
   identify an appropriate directory now, you can edit the hosts file and add
   a tmpdir setting later. The scratch directory can be specified at runtime
   by setting the SCHRODINGER_TMPDIR environment variable. See the Schrodinger
   Product Installation Guide and the product manual for more information.

Scratch directory?  (/scratch) /fsa/home/ljx_zhangzw/software
```

[illegible]

```
cd /fsa/home/ljx_zhangzw/software
echo "export Desmond=${PWD}/" >> ~/.bashrc
```

[illegible] Desmond[illegible] schrodinger.host[illegible] GPU

[illegible] 83a100ib[illegible] m001[illegible]

```
#localhost
name:localhost
temdir:/fsa/home/ljx_zhangzw/software


#HPC
name: HPC
host: m001
queue: LSF
qargs: -q 83a100ib -gpu num=1
tmpdir: /fsa/home/ljx_zhangzw/software
```

```
schrodinger: /fsa/home/ljx_zhangzw/software
gpgpu: 0, NVIDIA A100
gpgpu: 1, NVIDIA A100
gpgpu: 2, NVIDIA A100
gpgpu: 3, NVIDIA A100
gpgpu: 4, NVIDIA A100
gpgpu: 5, NVIDIA A100
gpgpu: 6, NVIDIA A100
gpgpu: 7, NVIDIA A100
```

[illegible]

1[illegible] tmpdir[illegible] schrodinger[illegible]

2[illegible] host[illegible] 83a100ib[illegible] gpgpu: 0-7, NVIDIA A100[illegible] HPC[illegible] [illegible] [illegible]

3[illegible] Schrödinger[illegible] KNOWLEDGE BASE[illegible] [illegible] Q2[illegible]

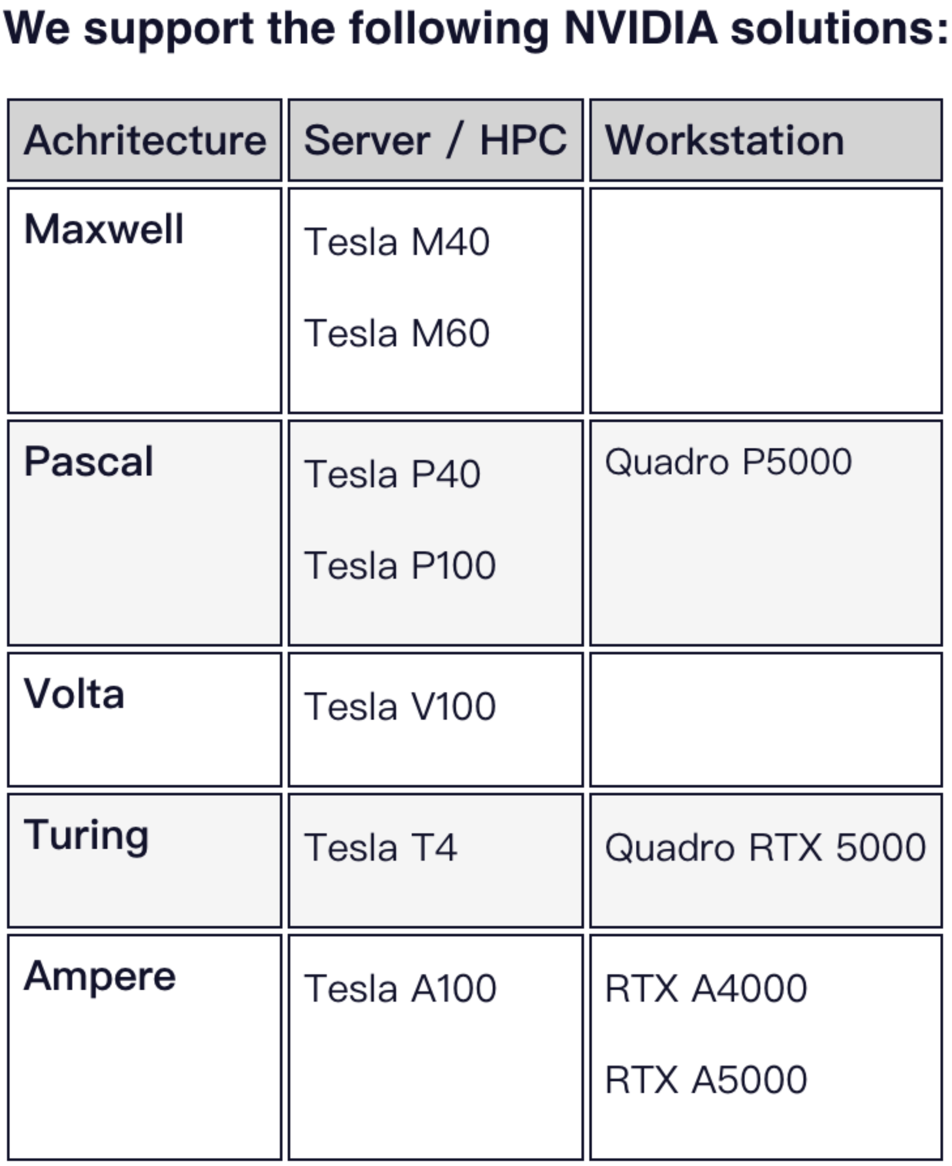
4[illegible] 2023[illegible] 6[illegible] 16[illegible] HPC[illegible] Desmond[illegible] [illegible] **3090**[illegible] **4090**[illegible] [illegible]

[illegible]	**GPU**	**Hostname**	[illegible]		
e5v3k40ib	2*Intel Xeon E5-2680v3 2*NVIDIA Tesla K40 12GB 128GB RAM 56Gb FDR InfiniBand	x001 gpgpu: 0, NVIDIA Tesla K40 gpgpu: 1, NVIDIA Tesla K40	[illegible] 1.2 [illegible] /[illegible] /[illegible] =0.1[illegible] /[illegible] /[illegible]	1.2	=0.1
e5v4p100ib	2*Intel Xeon E5-2660v4 2*NVIDIA Tesla P100 PCIe 16GB 128GB RAM 56Gb FDR InfiniBand	x002 gpgpu: 0, NVIDIA Tesla P100 gpgpu: 1, NVIDIA Tesla P100	[illegible] 1.68 [illegible] /[illegible] /[illegible] =0.12[illegible] /[illegible] /[illegible]		

62v100ib	2*Intel Xeon Gold 6248 8*NVIDIA Tesla V100 SXM2 32GB 768GB RAM 100Gb EDR InfiniBand	n002 gpgpu: 0, NVIDIA Tesla V100 gpgpu: 1, NVIDIA Tesla V100 gpgpu: 2, NVIDIA Tesla V100 gpgpu: 3, NVIDIA Tesla V100 gpgpu: 4, NVIDIA Tesla V100 gpgpu: 5, NVIDIA Tesla V100 gpgpu: 6, NVIDIA Tesla V100 gpgpu: 7, NVIDIA Tesla V100	[illegible] 3 [illegible] /[illegible] /[illegible] =0.6 [illegible] /[illegible] /[illegible]		
72rtxib	AMD EPYC 7302 4*NVIDIA TITAN RTX 24GB 128GB RAM 100Gb HDR100 InfiniBand	g005 or g006 or g007 gpgpu: 0, NVIDIA TITAN RTX gpgpu: 1, NVIDIA TITAN RTX gpgpu: 2, NVIDIA TITAN RTX gpgpu: 3, NVIDIA TITAN RTX	[illegible] 1.8 [illegible] /[illegible] /[illegible] =0.45[illegible] /[illegible] /[illegible]		

83a100ib	2*Intel Xeon Platinum 8358 8*NVIDIA Tesla A100 SXM4 40GB 512GB RAM 200Gb HDR InfiniBand	m001 gpgpu: 0, NVIDIA A100 gpgpu: 1, NVIDIA A100 gpgpu: 2, NVIDIA A100 gpgpu: 3, NVIDIA A100 gpgpu: 4, NVIDIA A100 gpgpu: 5, NVIDIA A100 gpgpu: 6, NVIDIA A100 gpgpu: 7, NVIDIA A100	[illegible] 4.8 [illegible] /[illegible] /[illegible] =0.6[illegible] /[illegible] /[illegible]		

[illegible] Desmond[illegible] GPGPU (General-purpose computing on graphics processing units) [illegible]



We support the following NVIDIA solutions:

Achritecture	Server / HPC	Workstation
Maxwell	Tesla M40 Tesla M60	
Pascal	Tesla P40 Tesla P100	Quadro P5000
Volta	Tesla V100	
Turing	Tesla T4	Quadro RTX 5000
Ampere	Tesla A100	RTX A4000 RTX A5000

1[illegible] Tesla K20[illegible] Tesla K40[illegible] Tesla K80[illegible] GPGPU[illegible] NVIDIA [illegible] CUDA 11.2[illegible]

2[illegible] CUDA[illegible] 11.2[illegible] NVIDIA“[illegible] /[illegible] ”Linux[illegible]

3[illegible] GPU[illegible] GeForce GTX[illegible]

[illegible] MD[illegible]

[illegible] MD[illegible]

[illegible] Linux[illegible] Windows[illegible] [illegible] Web[illegible] HPC[illegible] [illegible]

[illegible] 2022[illegible] [illegible] [illegible] [illegible] Desmond[illegible] Windows[illegible] Mac[illegible] Linux[illegible]

[illegible] MD[illegible]

1[illegible] plmd/DSMDrun: Desmond[illegible]

[illegible] Wang-Lin-boop/Schrodinger-Script: Some scripts to run Schrödinger jobs on HPC or localhost. (github.com)

[illegible] Wang-Lin-boop/Schrodinger-Script: Some scripts to run Schrödinger jobs on HPC or localhost. (github.com)

[illegible] mae[illegible] PDB[illegible] Maestro[illegible]

[illegible] plmd -h[illegible]

```
Usage: plmd [OPTION] <parameter>

An automatic Desmond MD pipline for protein-ligand complex MD simulation.

Example:
1) plmd -i "*.mae" -S INC -P "chain.name A" -L "res.ptype UNK" -H HPC_CPU -G HPC_GPU
2) plmd -i "*.mae" -S OUC -P "chain.name A" -L "chain.name B" -t 200 -H HPC_CPU -G HPC_gpu01
3) plmd -i "*.mae" -S "TIP4P:Cl:0.15-Na-Cl+0.02-Fe2-Cl+0.02-Mg2-Cl" -L "res.num 999" -G
HPC_gpu03
4) plmd -i "*.cms" -P "chain.name A" -L "res.ptype ADP" -H HPC_CPU -G HPC_gpu04

Input parameter:
  -i[illegible]Use a file name (Multiple files are wrapped in "", and split by ' ') *.mae or *.cms ;
            or regular expression to represent your input file, default is *.mae.

System Builder parameter:
  -S    System Build Mode: <INC>
            INC: System in cell, salt buffer is 0.15M KCl, water is TIP3P. Add K to neutralize
system.
            OUC: System out of cell, salt buffer is 0.15M NaCl, water is TIP3P. Add Na to
neutralize system.
```

```
            Custom Instruct: Such as: "TIP4P:Cl:0.15-Na-Cl+0.02-Fe2-Cl+0.02-Mg2-Cl"
                Interactive addition of salt. Add Cl to neutralize system.
                    for positive_ion: Na, Li, K, Rb, Cs, Fe2, Fe3, Mg2, Ca2, Zn2 are
predefined.
                    for nagative_ion: F, Cl, Br, I are predefined.
                    for water: SPC, TIP3P, TIP4P, TIP5P, DMSO, METHANOL are predefined.

  -b□Define a boxshape for your systems. <cubic>
            box types: dodecahedron_hexagon, cubic, orthorhombic, triclinic
  -s□Define a boxsize for your systems.  <15.0>
□□for dodecahedron_hexagon and cubic, defulat is 15.0;
□□for orthorhombic or triclinic box, defulat is [15.0 15.0 15.0];
□□If you want use Orthorhombic or Triclinic box, your parameter should be like "15.0 15.0 15.0"
  -R    Redistribute the mass of heavy atoms to bonded hydrogen atoms to slow-down high
frequency motions.
  -F□Define a force field to build your systems. <OPLS_2005>
□□OPLS_2005, S-OPLS, OPLS3e, OPLS3, OPLS2 are recommended to protein-ligand systems.

Simulation control parameter:
  -m□Enter the maximum simulation time for the Brownian motion simulation, in ps. <100>
  -t    Enter the Molecular dynamics simulation time for the product simulation, in ns. <100>
  -T    Specify the temperature to be used, in kelvin. <310>
  -N    Number of Repeat simulation with different random numbers. <1>
  -P    Define a ASL to protein, such as "protein".
  -L    Define a ASL to ligand, such as "res.ptype UNK".
  -q    Turn off protein-ligand analysis.
  -u    Turn off md simulation, only system build.
  -C    Set constraint to an ASL, such as "chain.name A AND backbone"
  -f    Set constraint force, default is 10.
  -o    Specify the approximate number of frames in the trajectory.  <1000>
        This value is coupled with the recording interval for the trajectory and the
simulation time: the number of frames times the trajectory recording interval is the total
simulation time.
        If you adjust the number of frames, the recording interval will be modified.

Job control:
  -G□HOST of GPU queue, default is HPC_GPU.
  -H    HOST of CPU queue, default is HPC_CPU.
  -D□Your Desmond path. <$Desmond>
```

2[illegible] AutoMD: Desmond[illegible]

[illegible] Wang-Lin-boop/AutoMD: Easy to get started with molecular dynamics simulation. (github.com)

[illegible] AutoMD[illegible] MD[illegible] Linux[illegible] - [illegible] (zhihu.com)

[illegible] Desmond[illegible] plmd[illegible] **plmd[illegible]** **AutoMD[illegible]** D.E.Shaw Research[illegible] Viparr[illegible] Msys[illegible] Desmond[illegible] Amber[illegible] Charmm[illegible]

#[illegible] HPC[illegible] Viparr[illegible] Msys

[illegible] Linux[illegible] home[illegible] software[illegible]

```
wget https://github.com/DEShawResearch/viparr/releases/download/4.7.35/viparr-4.7.35-cp38-
cp38-manylinux2014_x86_64.whl
wget https://github.com/DEShawResearch/msys/releases/download/1.7.337/msys-1.7.337-cp38-cp38-
manylinux2014_x86_64.whl
git clone git://github.com/DEShawResearch/viparr-ffpublic.git
git clone https://github.com/Wang-Lin-boop/AutoMD
```

[illegible] desmond[illegible] Viparr[illegible] Msys[illegible]

```
cd /fsa/home/ljx_zhangzw/software
./run schrodinger_virtualenv.py schrodinger.ve
source schrodinger.ve/bin/activate
pip install --upgrade pip
pip install msys-1.7.337-cp38-cp38-manylinux2014_x86_64.whl
pip install viparr-4.7.35-cp38-cp38-manylinux2014_x86_64.whl
echo "export viparr=${PWD}/schrodinger.ve/bin" >> ~/.bashrc
```

[illegible] viparr-ffpublic.git[illegible]

```
echo "export VIPARR_FFPATH=${PWD}/viparr-ffpublic/ff" >> ~/.bashrc
```

[illegible] AutoMD.git[illegible]

```
cd AutoMD
echo "alias AutoMD=${PWD}/AutoMD" >> ~/.bashrc
chmod +x AutoMD
source ~/.bashrc
cp -r ff/* ${VIPARR_FFPATH}/
```

#AutoMD -h

[illegible] -[illegible] -[illegible] DNA/RNA-[illegible] Meastro[illegible]

[illegible] -i[illegible] .mae[illegible] Maestro[illegible] .cms[illegible]

[illegible] -S[illegible] INC[illegible] 0.15M KCl[illegible] SPC[illegible] K+[illegible] OUC[illegible] 0.15M NaCl[illegible] SPC [illegible] Na+[illegible] "SPC:Cl:0.15-K-Cl+0.02-Mg2-Cl"[illegible]

[illegible] -b[illegible] -s[illegible] Box[illegible]

[illegible] -F[illegible] Desmond[illegible] S-OPLS[illegible] OPLS_2005 [illegible] OPLS[illegible] -[illegible] Amber[illegible] -[illegible] Charmm[illegible] DES-Amber[illegible] PPI[illegible] Amber [illegible] Charmm[illegible]

-F Charmm ff	*aa.charmm.c36m, misc.charmm.all36, carb.charmm.c36, ethers.charmm.c35, ions.charmm36, lipid.charmm.c36 and na.charmm.c36*
-F Amber ff	*aa.amber.19SBmisc, aa.amber.ffncaa, lipid.amber.lipid17, ions.amber1lm_iod.all, ions.amber2ff99.tip3p, na.amber.bsc1 and na.amber.tan2018*
-F DES-Amber	*aa.DES-Amber_pe3.2, dna.DES-Amber_pe3.2, rna.DES-Amber_pe3.2 and other force fields in -F Amber*

```
Usage: AutoMD [OPTION] <parameter>


Example:
```

```
1) MD for cytoplasmic protein-ligand complex:
AutoMD -i "*.mae" -S INC -P "chain.name A" -L "res.ptype UNK" -F "S-OPLS"
2) MD for plasma protein-protein complex:
AutoMD -i "*.mae" -S OUC -F "DES-Amber"
3) MD for DNA/RNA-protein complex:
AutoMD -i "*.mae" -S "SPC:Cl:0.15-K-Cl+0.02-Mg2-Cl" -F Amber
4) MD for membrane protein, need to prior place membrane in Meastro.
AutoMD -i "*.mae" -S OUC -l "POPC" -r "Membrane" -F "Charmm"

Input parameter:
  -i    Use a file name (Multiple files are wrapped in "", and split by ' ') *.mae or *.cms ;
            or regular expression to represent your input file, default is *.mae.

System Builder parameter:
  -S    System Build Mode: <INC>
        INC: System in cell, salt buffer is 0.15M KCl, water is SPC. Add K to neutralize
system.
        OUC: System out of cell, salt buffer is 0.15M NaCl, water is SPC. Add Na to neutralize
system.
        Custom Instruct: Such as: "TIP4P:Cl:0.15-Na-Cl+0.02-Fe2-Cl+0.02-Mg2-Cl"
            Interactive addition of salt. Add Cl to neutralize system.
                for positive_ion: Na, Li, K, Rb, Cs, Fe2, Fe3, Mg2, Ca2, Zn2 are predefined.
                for nagative_ion: F, Cl, Br, I are predefined.
                for water: SPC, TIP3P, TIP4P, TIP5P, DMSO, METHANOL are predefined.
  -l    Lipid type for membrane box. Use this option will build membrane box. <None>
            Lipid types: POPC, POPE, DPPC, DMPC.
  -b    Define a boxshape for your systems. <cubic>
            box types: dodecahedron_hexagon, cubic, orthorhombic, triclinic
  -s    Define a boxsize for your systems.  <15.0>
            for dodecahedron_hexagon and cubic, defulat is 15.0;
            for orthorhombic or triclinic box, defulat is [15.0 15.0 15.0];
            If you want use Orthorhombic or Triclinic box, your parameter should be like "15.0
15.0 15.0"
  -R    Redistribute the mass of heavy atoms to bonded hydrogen atoms to slow-down high
frequency motions.
  -F    Define a force field to build your systems. <OPLS_2005>
            OPLS_2005, S-OPLS are recommended to receptor-ligand systems.
            Amber, Charmm, DES-Amber are recommended to other systems. Use -O to show more
details.
            Use the "Custom" to load parameters from input .cms file.
```

```

Simulation control parameter:
  -m    Enter the maximum simulation time for the Brownian motion simulation, in ps. <100>
  -r    The relaxation protocol before MD, "Membrane" or "Solute". <Solute>
  -e    The ensemble class in MD stage, "NPT", "NVT", "NPgT". <NPT>
  -t    Enter the Molecular dynamics simulation time for the product simulation, in ns. <100>
  -T    Specify the temperature to be used, in kelvin. <310>
  -N    Number of Repeat simulation with different random numbers. <1>
  -P    Define a ASL to receptor, such as "protein".
  -L    Define a ASL to ligand and run interaction analysis, such as "res.ptype UNK".
  -u    Turn off md simulation, only system build.
  -C    Set constraint to an ASL, such as "chain.name A AND backbone"
  -f    Set constraint force, default is 10.
  -o    Specify the approximate number of frames in the trajectory.  <1000>
        This value is coupled with the recording interval for the trajectory and the
simulation time:
        the number of frames times the trajectory recording interval is the total simulation
time.
        If you adjust the number of frames, the recording interval will be modified.

Job control:
  -G    HOST of GPU queue, default is GPU.
  -H    HOST of CPU queue, default is CPU.
  -D    Your Desmond path. </fsa/home/ljx_zhangzw/>
  -V    Your viparr path. </fsa/home/ljx_zhangzw/schrodinger.ve/bin>
  -v    Your viparr force fields path. </fsa/home/ljx_zhangzw/viparr-ffpublic/ff>
```

□□□□□□□□□□□   □□□□□□□□□□□□□□□□□□

```
#BSUB -L /bin/bash
#BSUB -q 83a100ib
#BSUB -m m001

module load cuda/12.0.0
export Desmond=/fsa/home/ljx_zhangzw/
export viparr=/fsa/home/ljx_zhangzw/schrodinger.ve/bin
export VIPARR_FFPATH=/fsa/home/ljx_zhangzw/viparr-ffpublic/ff
alias AutoMD=/fsa/home/ljx_zhangzw/AutoMD/AutoMD

```

```
echo $PATH
echo $LD_LIBRARY_PATH
nvidia-smi
env|grep CUDA

AutoMD -i "*.mae" -S OUC -P "chain.name A" -L "res.ptype UNK" -F "OPLS_2005" -T 298 -s 10 -H m001 -G m001
```

[illegible] -P[illegible] -L[illegible] mae[illegible] -m[illegible]

-T[illegible] 298 K[illegible] 100 ns[illegible] 1[illegible] -H[illegible] -G[illegible] host [illegible]

[illegible]

[illegible] [illegible] [illegible] Desmond[illegible] AutoMD[illegible] 12[illegible]

```
Summary of user stages:
  stage 1 - task
  stage 2 - build_geometry
  stage 3 - assign_forcefield
  stage 4 - simulate, Brownian Dynamics NVT, T = 10 K, small timesteps, and restraints on solute_heavy_atom, 100 ps
  stage 5 - simulate, Brownian Dynamics NVT, T = 10 K, small timesteps, and restraints on user defined sets, 100 ps
  stage 6 - simulate, Langevin small steps NVT, T = 10 K, and restraints on solute heavy atoms, 12 ps
  stage 7 - simulate, Langevin NPT, T = 10 K, and restraints on solute heavy atoms, 12 ps
  stage 8 - simulate, Langevin NPT, T = 10 K, and restraints on solute heavy atoms, 12 ps
  stage 9 - simulate, Langevin NPT and restraints on solute heavy atoms, 12ps
  stage 10 - simulate, Langevin NPT and no restraints, 24ps
  stage 11 - simulate, Final MD and analysis, 100000.0 ps
  stage 12 - pl_analysis
(12 stages in total)
```

[illegible] 83a100ib[illegible] 72rtxib[illegible]

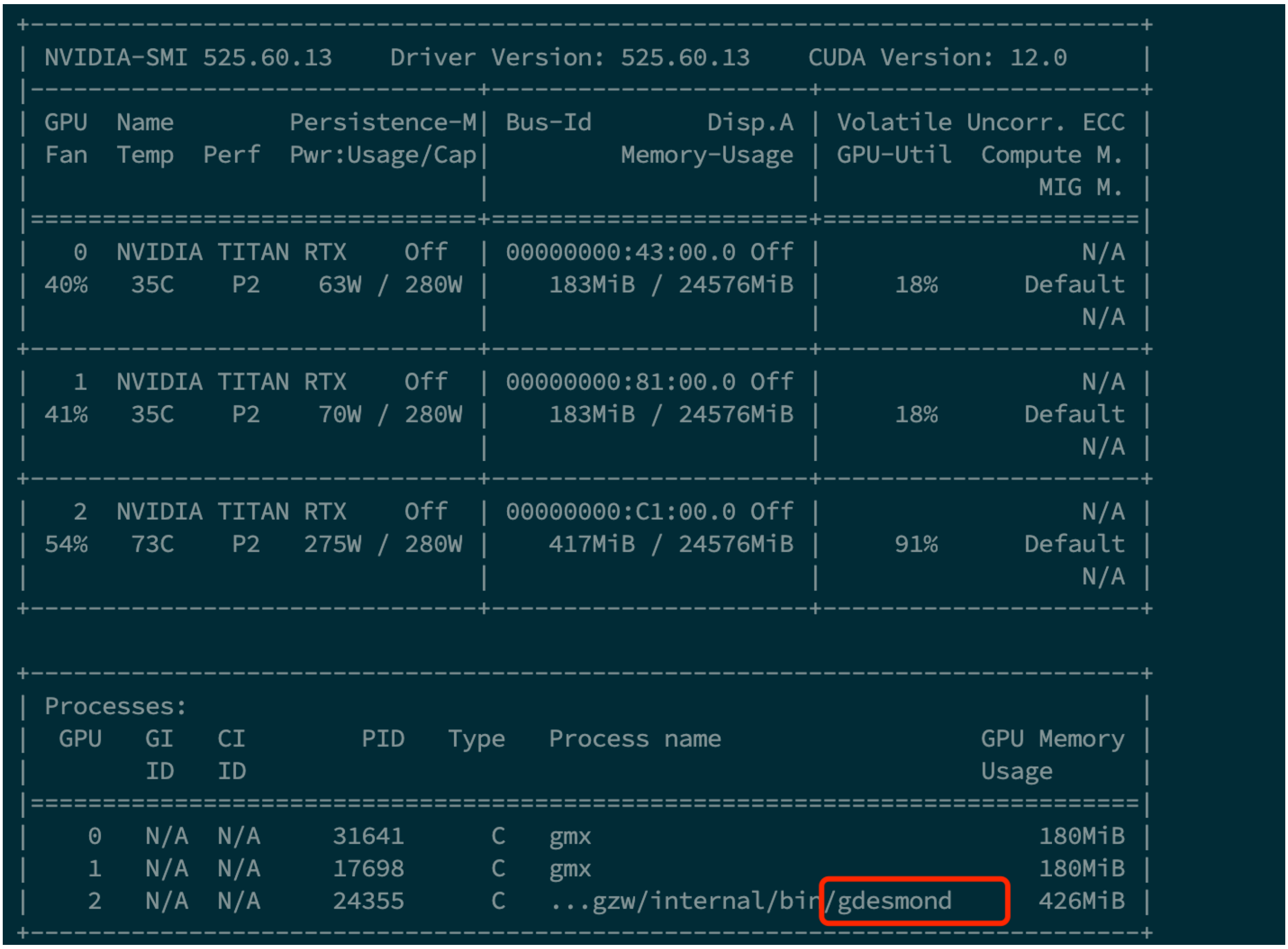
```
System Atoms：21173
System Residues：6363
System MOls：6202
Protein Atoms：2584
Protein Residues：162
Ligand Atoms：73
Ligand Residues：1



MIN time: 100 ps
MD time: 100000.0 ps
temperature: 298 K
Repeat: 1
Rondom numbers list: 2007

#Desmond
Multisim summary :
Total duration: 3h 38' 7"
Total GPU time: 3h 24' 42" (used by 8 subjob(s))

#HPC-83a100ib
Resource usage summary:

    CPU time :                                   12745.00 sec.
    Max Memory :                                 965 MB
    Average Memory :                             893.78 MB
    Total Requested Memory :                     -
    Delta Memory :                               -
    Max Swap :                                   -
    Max Processes :                              15
    Max Threads :                                45
    Run time :                                   13262 sec.
    Turnaround time :                            13260 sec.
```

```
System Atoms：63284
System Residues：20178
System MOls：19955
Protein Atoms：2585
Protein Residues：224
Ligand Atoms：72
```

```
Ligand Residues□1

MIN time: 100 ps
MD time: 100000.0 ps
temperature: 298 K
Repeat: 1
Rondom numbers list: 2007

#Desmond
Multisim summary :
  Total duration: 7h 51' 5"
  Total GPU time: 7h 27' 48" (used by 8 subjob(s))

#HPC-72rtxib
Resource usage summary:

    CPU time :                                   31316.00 sec.
    Max Memory :                                 1634 MB
    Average Memory :                             400.16 MB
    Total Requested Memory :                     -
    Delta Memory :                               -
    Max Swap :                                   -
    Max Processes :                              15
    Max Threads :                                40
    Run time :                                   85601 sec.
    Turnaround time :                            85602 sec.
```

□□□□□□□□ 72rtxib g007□□□□□□□ gmx□ desmond□ GPU □□□□□□□□□□□□□□□□□□□□□□□□ desmond□□□□□□□□□□□□□ GPU □□□□

*****□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ QQ□□□ *****



```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.60.13    Driver Version: 525.60.13    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA TITAN RTX    Off  | 00000000:43:00.0 Off |                  N/A |
| 40%   35C    P2    63W / 280W |    183MiB / 24576MiB |     18%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA TITAN RTX    Off  | 00000000:81:00.0 Off |                  N/A |
| 41%   35C    P2    70W / 280W |    183MiB / 24576MiB |     18%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA TITAN RTX    Off  | 00000000:C1:00.0 Off |                  N/A |
| 54%   73C    P2   275W / 280W |    417MiB / 24576MiB |     91%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A     31641      C   gmx                               180MiB |
|    1   N/A  N/A     17698      C   gmx                               180MiB |
|    2   N/A  N/A     24355      C   ...gzw/internal/bin/gdesmond      426MiB |
+-----------------------------------------------------------------------------+
```

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ _□□□□□ -md□□□□□□□□□□□

□□□□□□□□□□	AutoMD.msj
□□□□ launching□□□□	md_multisim.log
□□□□ launching□□□□	md_1-out.tgz … md_12-out.tgz
□□□□□□□□	-in.cms and -out.cms
□□□□□□□□	md.cpt.cfg and md-out.cfg
□□□□□□□	md_trj
□□□□□□□□	md.log
□□□□□	md.cpt

[illegible]	md.ene
[illegible] RMSF[illegible] RMSF[illegible] SSE[illegible]	md.eaf

[illegible] MD[illegible] [Scripts | Schrödinger (schrodinger.com)](https://www.schrodinger.com)[illegible]

[illegible]	
Resume previous MD	bash resume.sh
Extend MD	bash extend.sh # The total time will be extend to 2-fold of initial time
Cluster trajectory	bash cluster.sh "<rmsd selection>" "<fit selection>" "<number>"
Analysis Occupancy	bash occupancy.sh "<selection to analysis>" "<fit selection>"
Analysis ppi contact	bash ppi.sh "<Component A>" "<Component B>"
[illegible] MD[illegible]	
Calculate the radius of gyration	calc_radgyr.py
Thermal MMGBSA	thermal_mmgbsa.py
Calculate Trajectory B-factors	trajectory_bfactors.py
Analyze a Desmond Simulation	analyze_simulation.py
Merge Trajectories	trj_merge.py

[illegible] terminal[illegible] $SCHRODINGER/run *.py -h

[illegible] (LINUX)[illegible] 80-100ns[illegible] mmgbsa[illegible]

```
vim [illegible]/.bashrc
#[illegible]schrodinger[illegible]
export SCHRODINGER=/opt/schrodinger2018-1
source [illegible]/.bashrc
#[illegible]80-100 ns[illegible]mmgbsa[illegible]4[illegible]
$SCHRODINGER/run thermal_mmgbsa.py *-2007-md-out.cms -j casein80_100 -start_frame 800  -
```

```
end_frame 1000  -HOST localhost:8 -NJOBS 4 -step_size 4
```

[illegible] thermal_mmgbsa.py[illegible] breakdown_MMGBSA_by_residue.py[illegible] mmgbsa [illegible]

```
$SCHRODINGER/run breakdown_MMGBSA_by_residue.py *-prime-out.maegz *-breakdown.csv -average
```

[illegible] thermal_mmgbsa.py[illegible] MD_0_100ns-complexes.maegz[illegible] calc_radgyr.py [illegible]

```
$SCHRODINGER/run calc_radgyr.py -h
#usage: calc_radgyr.py [-v] [-h] [-asl ASL] infile

Calculates the radius of gyration of structures in the input file.

Calculate the radius of gyration of each structure in the input file,
which can be Maestro, SD, PDB, or Mol2 format.

Copyright Schrodinger, LLC. All rights reserved.

positional arguments:
  infile        Input file.

optional arguments:
  -v, -version  Show the program's version and exit.
  -h, -help     Show this help message and exit.
  -asl ASL      ASL expression for atoms to be included in the calculation. By
                default, the radius of gyration includes all atoms.
#[illegible]
$SCHRODINGER/run thermal_mmgbsa.py *-2007-md-out.cms -j * -start_frame 0  -end_frame 1001  -
HOST localhost:8 -NJOBS 4 -step_size 1
$SCHRODINGER/run calc_radgyr.py *-2007-md-out.cms
```

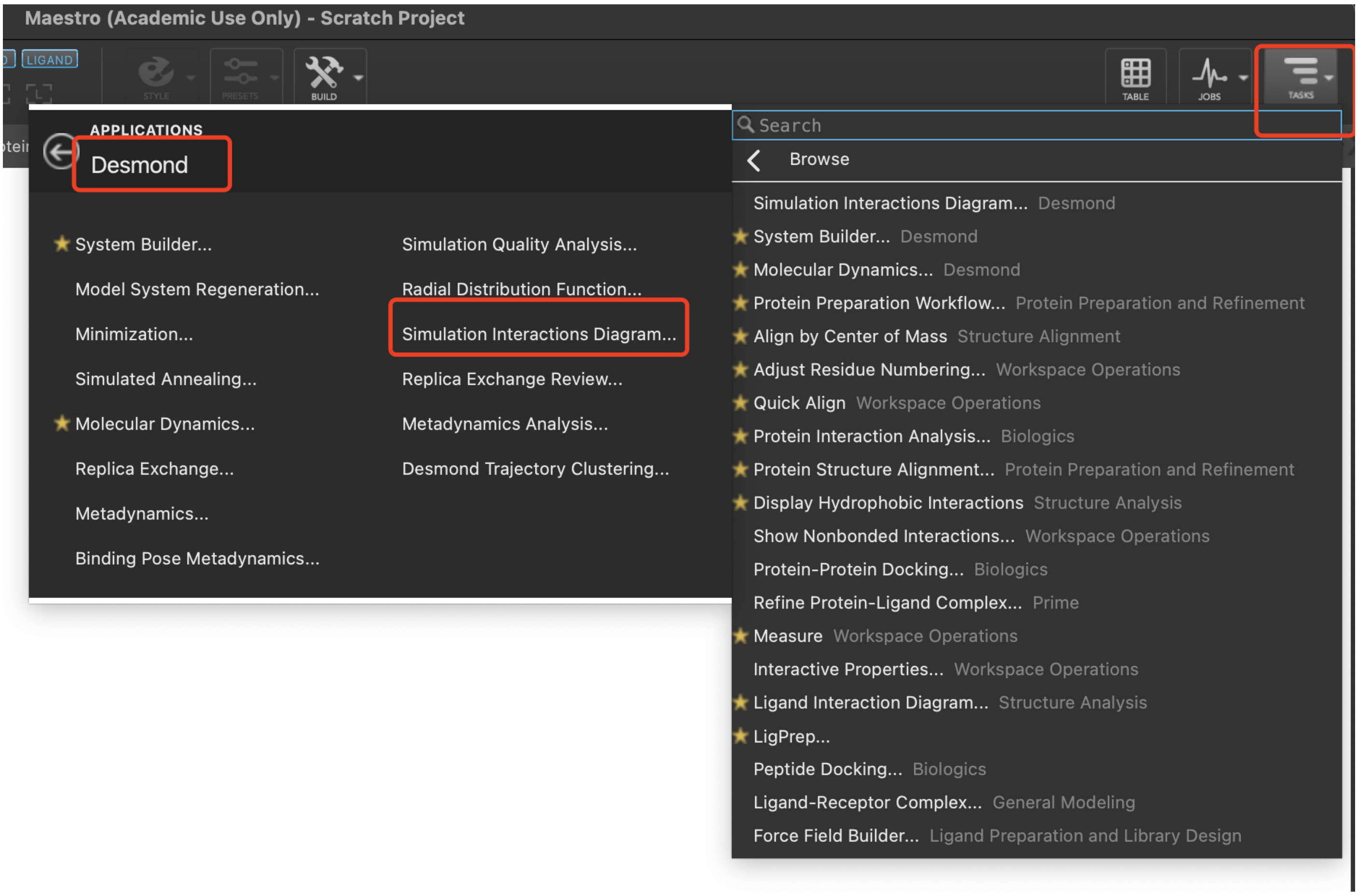
[illegible] binding_sasa.py[illegible] SASA[illegible]

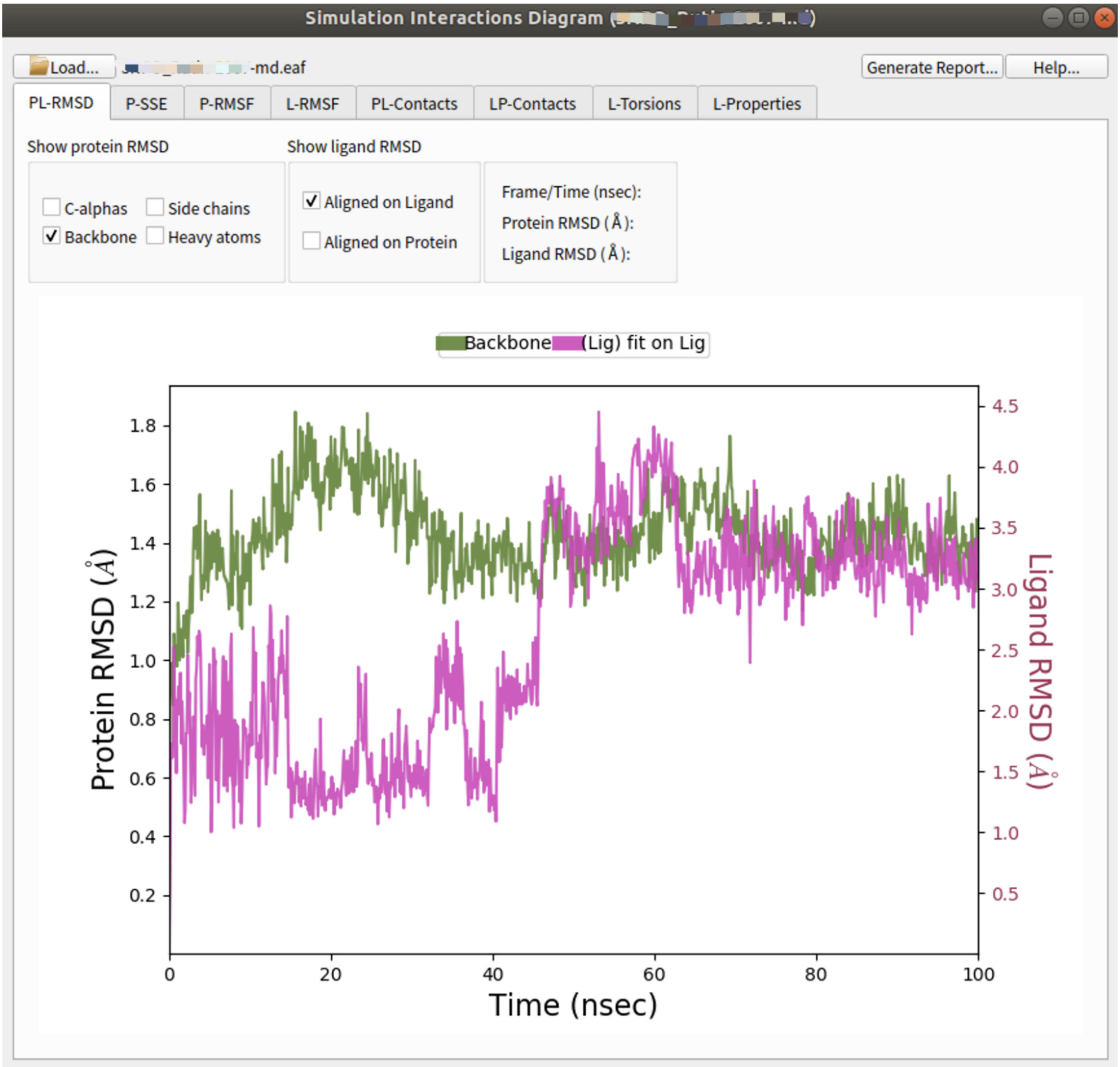
```
$SCHRODINGER/run binding_sasa.py -h
#usage: binding_sasa.py [-v] [-h] [-r RESIDUES] [-d <angstrom distance> | -f] [-s] [-n]
                       [-o OUTPUT_FILE] [-l ASL] [-j JOBNAME]
```

```
                        [--trajectory <path to trj>] [--ligand_asl <ligand asl>]
                        [--framestep <frames in each step>]
                        [--water_asl <asl for waters to keep>]
                        input_file

Calculates SASA of ligand and receptor before and after receptor-ligand binding.

Computes the change in solvent accessible surface area (SASA) upon binding for
a ligand and receptor. The total SASA for the unbound system and the
difference upon binding is computed and decomposed into functional subsets,
such as per-residue terms, charged, polar, and hydrophobic

The script will operate on PV files, Desmond CMS files or a receptor-ligand
complex Maestro file.

Copyright Schrodinger, LLC. All rights reserved.

positional arguments:
  input_file            Input PV file, Desmond CMS file, or a receptor-ligand complex
                        Maestro file.

optional arguments:
  -v, -version          Show the program's version and exit.
  -h, -help             Show this help message and exit.
  -r RESIDUES, --residue RESIDUES
                        Add specific residue to SASA report. Can be done multiple times
                        to add multiple residues. (format: ChainID,ResNum)
  -d <angstrom distance>, --distance <angstrom distance>
                        Distance cutoff in angstroms. Receptor atoms further than
                        cutoff distance from ligand will not be included in protein-
                        ligand complex SASA calculation. Default is 5 angstroms.
  -f, --full            Calculates SASA for entire ligand and protein.
  -s, --structure       For each ligand and receptor, append additional output
                        reporting the surface area of backbone atoms, side chain atoms,
                        and atoms of each residue type.
  -n, --secondary       For each ligand and receptor, append additional output
                        reporting surface area for helix , strand and loop receptor
                        atoms.
  -o OUTPUT_FILE, --output OUTPUT_FILE
                        Where to write the output file. If not specified, for PV input,
```

```
                        output file will be called <jobname>_out_pv.mae; for complex
                        files, it will be named <jobname>-out.mae. For --trajectory
                        jobs, only CSV output file is written
  -l ASL, --selection ASL
                        ASL pattern to define specific residues to calculate SASA for.
  -j JOBNAME, --jobname JOBNAME
                        Sets the jobname - base name for output files. If not
                        specified, it will be derived from the input file name.
  --trajectory <path to trj>
                        Compute binding SASA over a trajectory. Use of this option
                        means the input file should be a Desmond .cms file. The path to
                        the Desmond trj directory must be supplied as the argument to
                        this flag. Output will be to a .csv file only. The ligand
                        structure will be determined automatically unless the
                        --ligand_asl flag is used.
  --ligand_asl <ligand asl>
                        The ASL used to determine the ligand structure. By default,
                        this is determined automatically. Only valid for --trajectory
                        jobs.
  --framestep <frames in each step>
                        --trajectory jobs calculate SASA every X frames. By default,
                        X=10. Use this flag to change the value of X.
  --water_asl <asl for waters to keep>
                        By default, --trajectory jobs remove all waters before
                        calculating SASA. Use this to specify waters to keep
                        (--water_asl=all keeps all waters, --water_asl="within 5
                        backbone" keeps all waters with 5 Angstroms of the protein
                        backbone).
#[illegible]
$SCHRODINGER/run binding_sasa.py *-2007-md-out.cms -f -j sasa_complex --trajectory *-2007-
md_trj  --framestep 1
```

[illegible] _[illegible] -md[illegible] TASKS[illegible] Desmond[illegible] Simulation Interaction Digram[illegible] md.eaf[illegible]


Maestro (Academic Use Only) - Scratch Project
LIGAND
STYLE
PRESETS
BUILD
TABLE
JOBS
TASKS
APPLICATIONS
Desmond
System Builder...
Model System Regeneration...
Minimization...
Simulated Annealing...
Molecular Dynamics...
Replica Exchange...
Metadynamics...
Binding Pose Metadynamics...
Simulation Quality Analysis...
Radial Distribution Function...
Simulation Interactions Diagram...
Replica Exchange Review...
Metadynamics Analysis...
Desmond Trajectory Clustering...
Search
Browse
Simulation Interactions Diagram... Desmond
System Builder... Desmond
Molecular Dynamics... Desmond
Protein Preparation Workflow... Protein Preparation and Refinement
Align by Center of Mass Structure Alignment
Adjust Residue Numbering... Workspace Operations
Quick Align Workspace Operations
Protein Interaction Analysis... Biologics
Protein Structure Alignment... Protein Preparation and Refinement
Display Hydrophobic Interactions Structure Analysis
Show Nonbonded Interactions... Workspace Operations
Protein-Protein Docking... Biologics
Refine Protein-Ligand Complex... Prime
Measure Workspace Operations
Interactive Properties... Workspace Operations
Ligand Interaction Diagram... Structure Analysis
LigPrep...
Peptide Docking... Biologics
Ligand-Receptor Complex... General Modeling
Force Field Builder... Ligand Preparation and Library Design



Simulation Interactions Diagram ([redacted])

Load... [redacted]-md.eaf

Show protein RMSD

☐ C-alphas ☐ Side chains
☑ Backbone ☐ Heavy atoms

Show ligand RMSD

☑ Aligned on Ligand
☐ Aligned on Protein

Frame/Time (nsec):
Protein RMSD (Å):
Ligand RMSD (Å):

□□ AutoMD□□ AutoTRJ□□□□□□□□□□□□□□□□□□□ AutoMD□□□□□□

```
(base) parallels@parallels:$ AutoTRJ -h
The trajectory analysis script for AutoMD. Refer to: https://github.com/Wang-Lin-boop/AutoMD
Usage: AutoTRJ [OPTION] <parameter>

Required Options:
  -i <string>□The path to trajectory or regular expression for trajectories(will be merged).
such as "*-md".
  -J <string>   Specify a basic jobname for this task, the basename of -i is recommended.
  -M <string>   The running mode for this analysis task. the <..> means some options.
```

```
                APCluster_<n>: affinity propagation clustering, the <n> is the number of most populated clusters.
                CHCluster_<n>_<cutoff>: centroid hierarchical clustering, the <cutoff> is the RMSD thethreshold.
                LigandAPCluster_<n>: APCluster for ligand, require "-L" option.
                LigandCHCluster_<n>_<cutoff>: CHCluster for ligand, require "-L" option.
                Occupancy: calculates the occupancy of component 2 in trajectory, require "-L" option.
                PPIContact: identifies interactions occurring between the components, require "-L" option.
                FEL: analyze the free energy landscape (FEL) for CA atoms, and cluster the trajectories by FEL.
                CCM: plot the cross-correlation matrix of the trajectory.
                SiteMap_<n>: running SiteMap on all frames of the trajectory, the <n> is the number of sites.
                PocketMonitor: monitor the ligand binding pocket on the trajectory, require "-L" option.
                MMGBSA: running MM-GBSA on the trajectory, require "-L" option.
                BFactor: calculate atom B-factors from trajectory (receptor and ligand).
                RMSF: calculate RMSF from trajectory (receptor).
                ConvertXTC: convert the trajectory to XTC format.
            You can parallel the analysis task by "+", such as, "APCluster_5+PPIContact+MMGBSA".

Analysis Range Options:
  -R <ASL>      The ASL of component 1 to be centered. <protein>
  -L <ASL>      The ASL of component 2 to be analyzed. <ligand>

Trajectories Processing Options:
  -T <string>   Slice trajectories when analyzed, if more than one trajectories given by -i, this slice will applied to all of them. The default is None. Such as "100:1000:1". <START:END:STEP>
  -t <string>   Slice trajectories when calculating MM/GBSA, such as "100:1000:1". <START:END:STEP>
  -c            During the clustering process, the frames of a trajectory are preserved for each cluster, while representative members are stored in the CMS files.
  -A <file>     Align the trajectory to a reference structure, given in ".mae" format.
  -a            Align the trajectory to the frist frame.
  -C <ASL>      Set a ASL to clean the subsystem from trajectory, such as -C "not solvent".
  -P            Parch waters far away from the component 2.
```

```
  -w <int>      Number of retained water molecules for parch (-P) stage. <200>
  -m            Treat the explicit membrane to implicit membrane during MM-GBSA calculation.

Job Control Options:
  -H <string>   HOST of CPU queue, default is CPU.[illegible]Desmond[illegible]schrondinger.host[illegible]
  -N <int>      CPU number for per analysis task. <30>
  -G <path>     Path to the bin of Gromacs. <[illegible]Gromacs[illegible]>
  -D <path>[illegible]    Path to the installation directory of Desmond. <[illegible]Desmond[illegible]Schrodinger[illegible]>
#[illegible]
AutoTRJ -i *-md_trj -J test_xpm -M FEL -G /usr/local/gromacs/bin/
#[illegible]md[illegible]test_xpm[illegible]FEL[illegible]gmx[illegible]
```

[illegible]

[illegible] AutoMD[illegible] Desmond[illegible] Desmond [illegible] [illegible] **D. E. Shaw Research** [illegible] [illegible]

AutoMD[illegible] MD[illegible] MD[illegible] **AutoMD**[illegible] **MD**[illegible] **Desmond**[illegible] [illegible]

[illegible] [illegible] **Wang-Lin**[illegible]

[illegible] **AutoMD**[illegible] A new variant of the colistin resistance gene MCR-1 with co-resistance to β-lactam antibiotics reveals a potential novel antimicrobial peptide. Liang L, Zhong LL, Wang L, Zhou D, Li Y, et al. (2023) A new variant of the colistin resistance gene MCR-1 with co-resistance to β-lactam antibiotics reveals a potential novel antimicrobial peptide. *PLOS Biology* 21(12): e3002433. https://doi.org/10.1371/journal.pbio.3002433

Revision #17
Created 2023-06-16 16:43:47 CST by Yao Ge
Updated 2024-03-27 15:46:24 CST by [illegible]