Implementing Workflow for Biophysical Simulations in openDIEL

You Han(City University of Hong Kong)
Mentors: Kwai Wong(UTK), John Ossyra(ORNL), Xiaolin Cheng(ORNL)

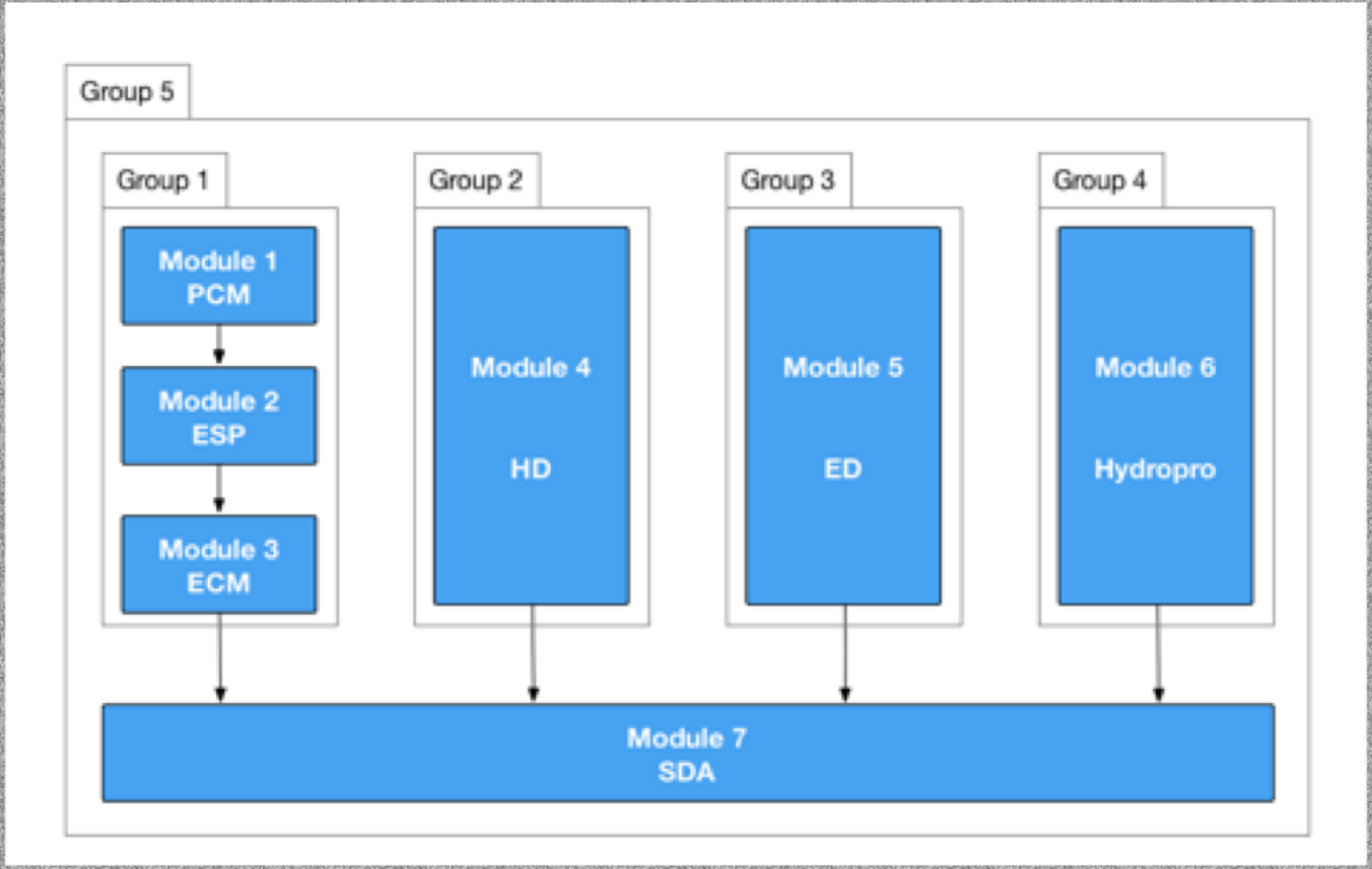
Background

About BD simulation

Brownian Dynamics simulation (BDS) is a method in computational biophysics used to study protein diffusion processes such as association mechanisms, measuring binding rates.

The biomolecules are regarded as rigid bodies; for each molecule, the motion is computed based on pre-calculated intermolecular physical potentials

About BD simulation(cont'd)



Module 1
PCM

Module 2
ESP

Module 3
ECM

Module 4
HD

Module 5
ED

Module 6
Hydropro

Module 7
SDA

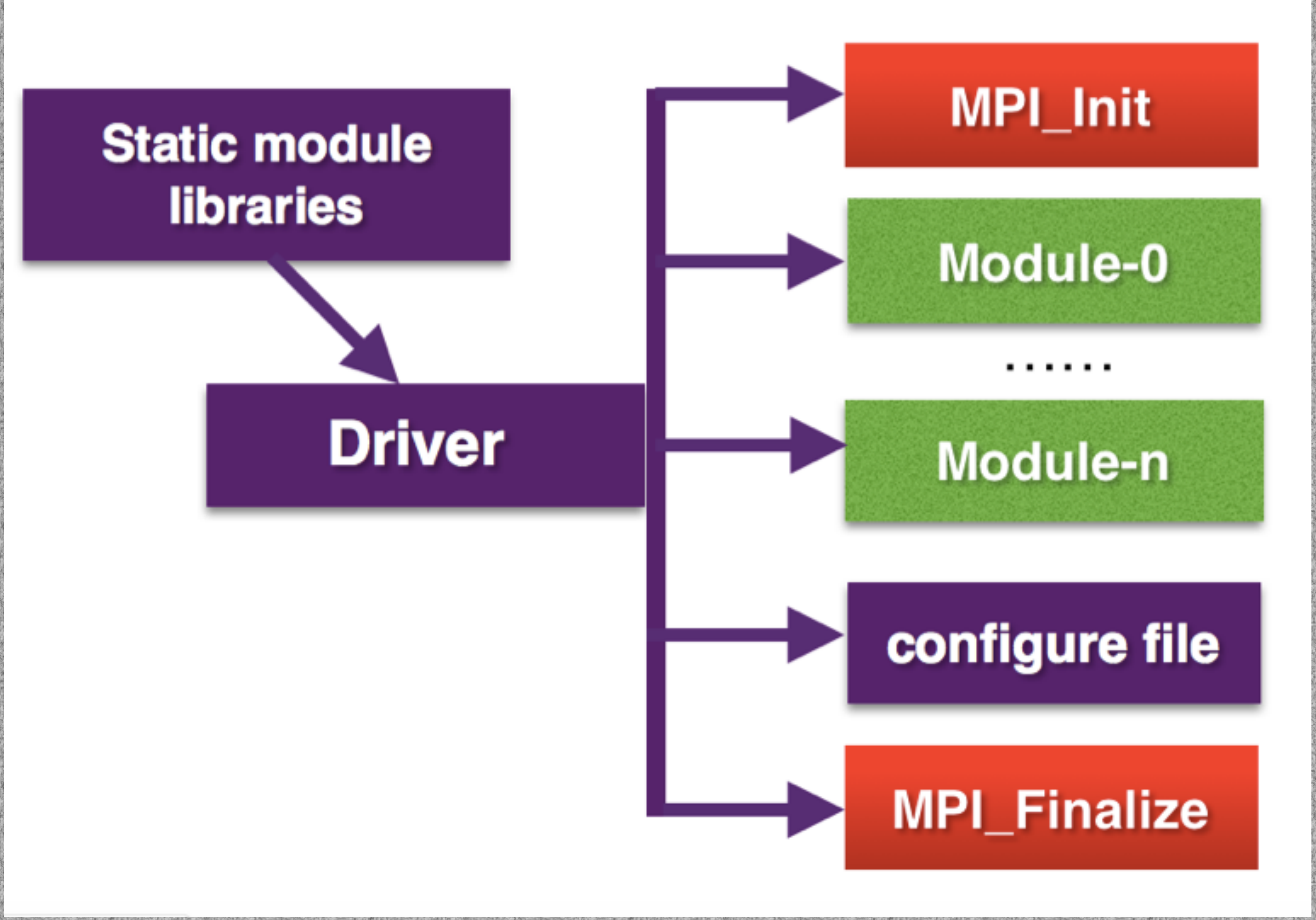
Fig. 1

Fig. 1 shows the organisational structure for preliminary molecular structure computa

About openDIEL

The **openDIEL**(open Distributive Interoperable Executive Library) is workflow engine that aims to facilitate communication between loosely coupled simulations in large-scale parallel computing. (openDIEL is developed by UT CFD LAB.)

About openDIEL(cont'd)



Static module libraries

Driver

MPI_Init

Module-0

……

Module-n

configure file

MPI_Finalize

Fig.2, simple mechanism of openDIEL

Abstract/Objectives

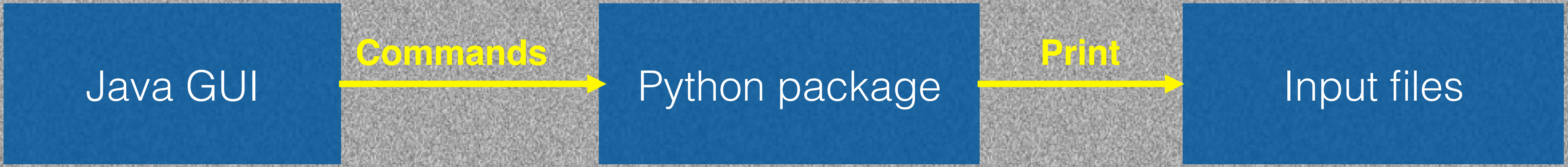
- First stage: Integrate with BDS toolkit from Mr. John Ossyra
- Second stage: build GUI with which users can configure and run BDS in openDIEL
- Third stage: Expand the GUI for parallel computing, insert modules of commonly used biophysical packages such as NAMD, Gromacs into the workflow and build physical coupling between simulations.

Objectives

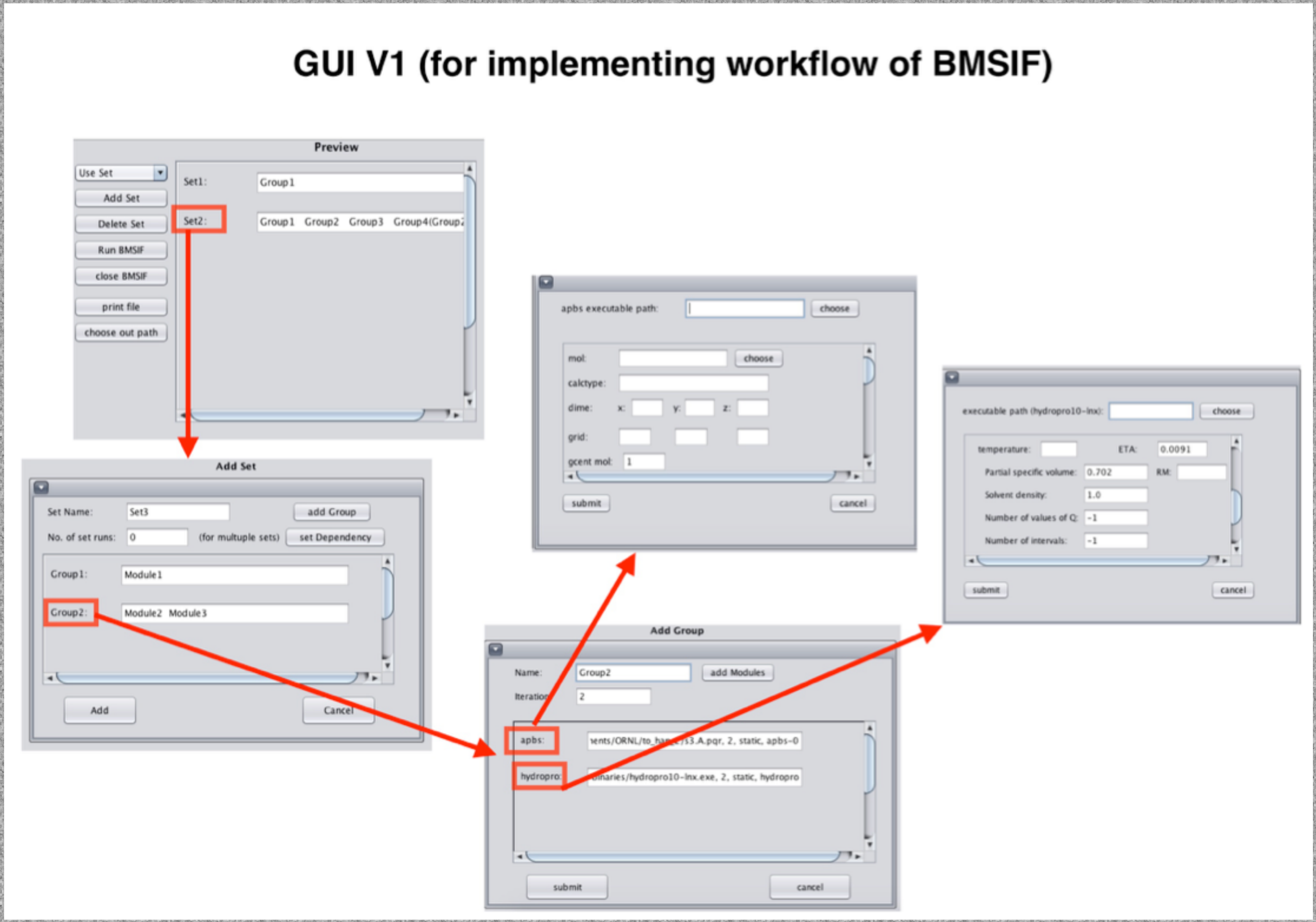
GUI V1—configure file & input files

GUI V1 provides a graphical user interface to interact with users, then generate the required input files for BD simulation and a configure file for openDIEL, according to user's definitions.

- Configure file for openDIEL
- Input files for Running BD simulation



GUI V1—configure file & input files(cont’d



GUI V1 (for implementing workflow of BMSIF)

Set1: Group1

Set2: Group1 Group2 Group3 Group4(Group

Set Name: Set3

No. of set runs: 0 (for multuple sets)

Group1: Module1

Group2: Module2 Module3

Name: Group2

Iteration: 2

apbs: ents/ORNL/to_ha[illegible]/s3.A.pqr, 2, static, apbs-0

hydropro: inaries/hydropro10-lnx.exe, 2, static, hydropro

apbs executable path:

mol:

calctype:

dime: x: y: z:

grid:

gcent mol: 1

executable path (hydropro10-lnx):

temperature:

ETA: 0.0091

Partial specific volume: 0.702

RM:

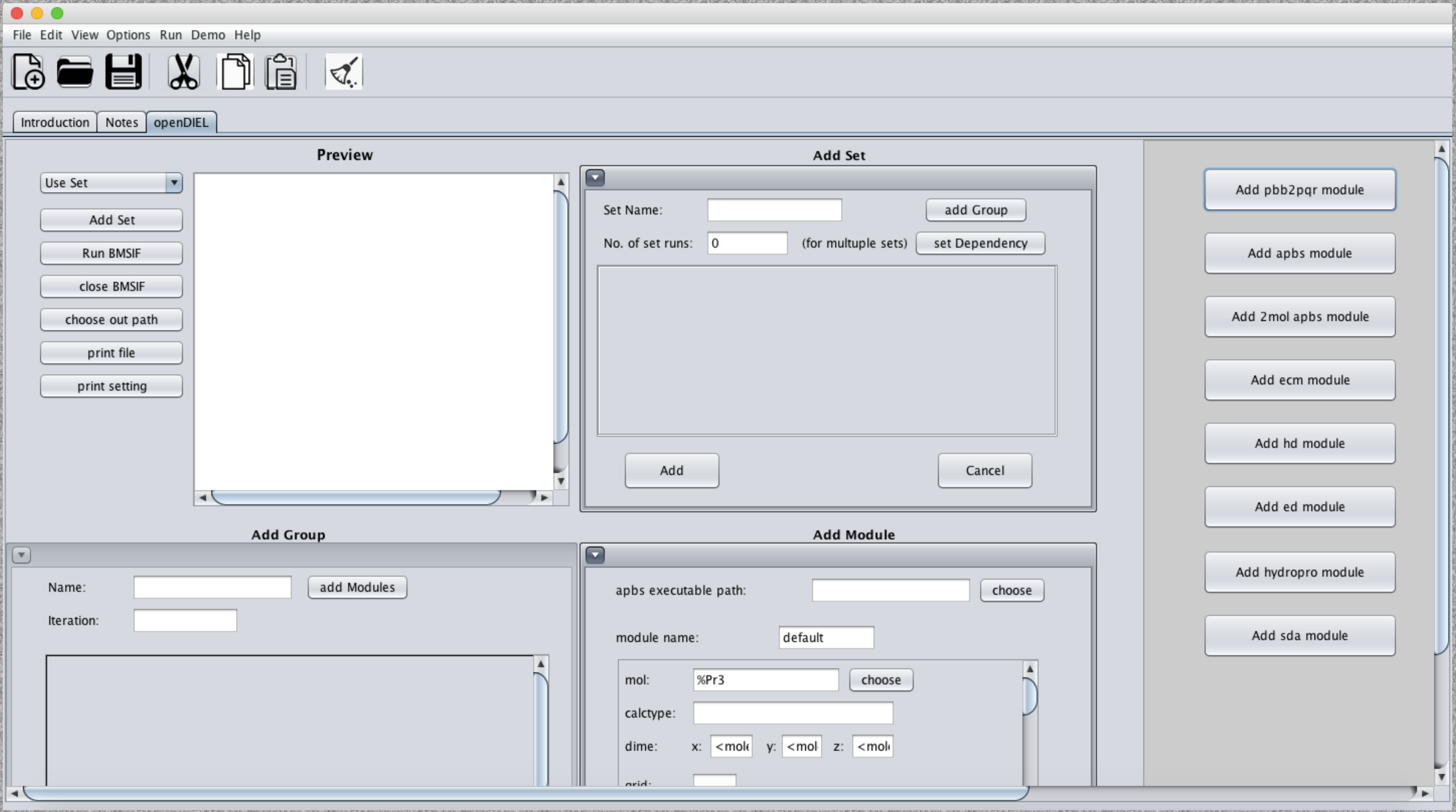
Solvent density: 1.0

Number of values of Q: -1

Number of intervals: -1

Fig.3 , the user interface of GUI V1

GUI V1—configure file & input files(cont'd



No. of set runs: 0

module name: default

mol: %Pr3

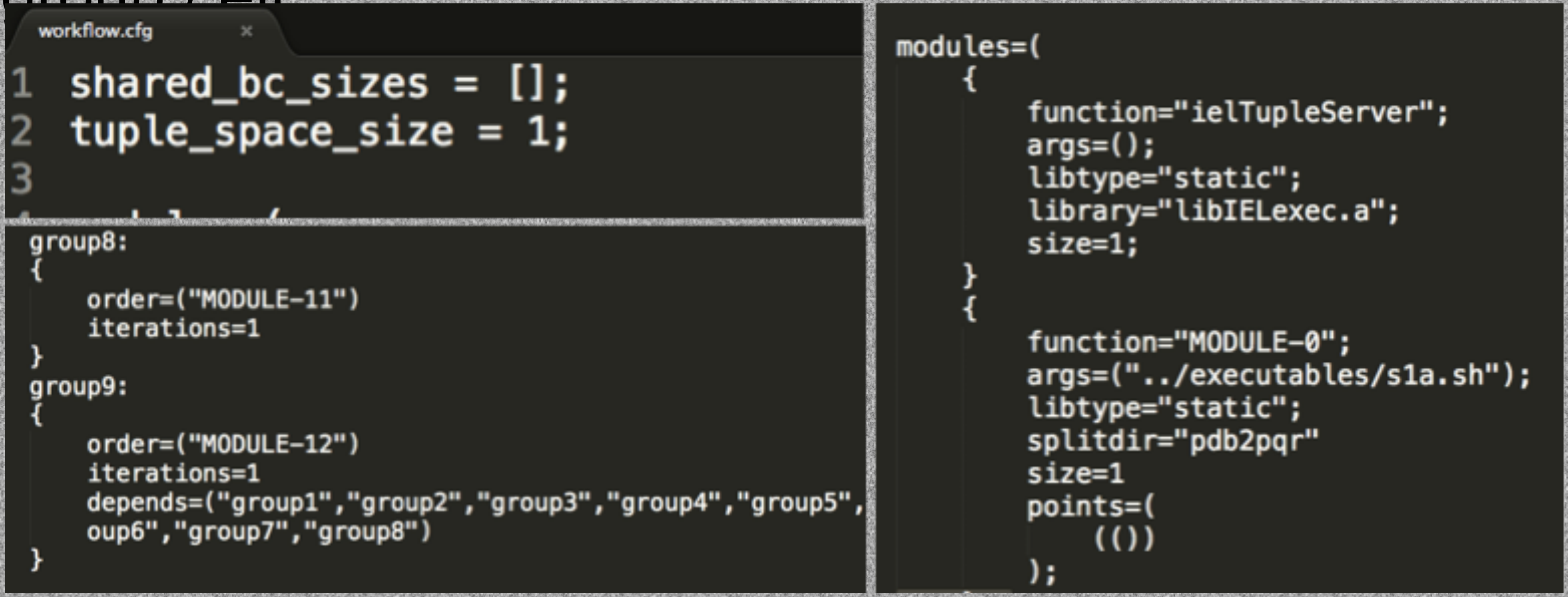
dime: x: <mol y: <mol z: <mol

Fig.4 , the user interface of GUI V1

GUI V1—configure file & input files(cont'd

About configure file:

size of tuple space / executable path / running size for each module, execution order / dependencies of each group / etc



```
shared_bc_sizes = [];
tuple_space_size = 1;

```

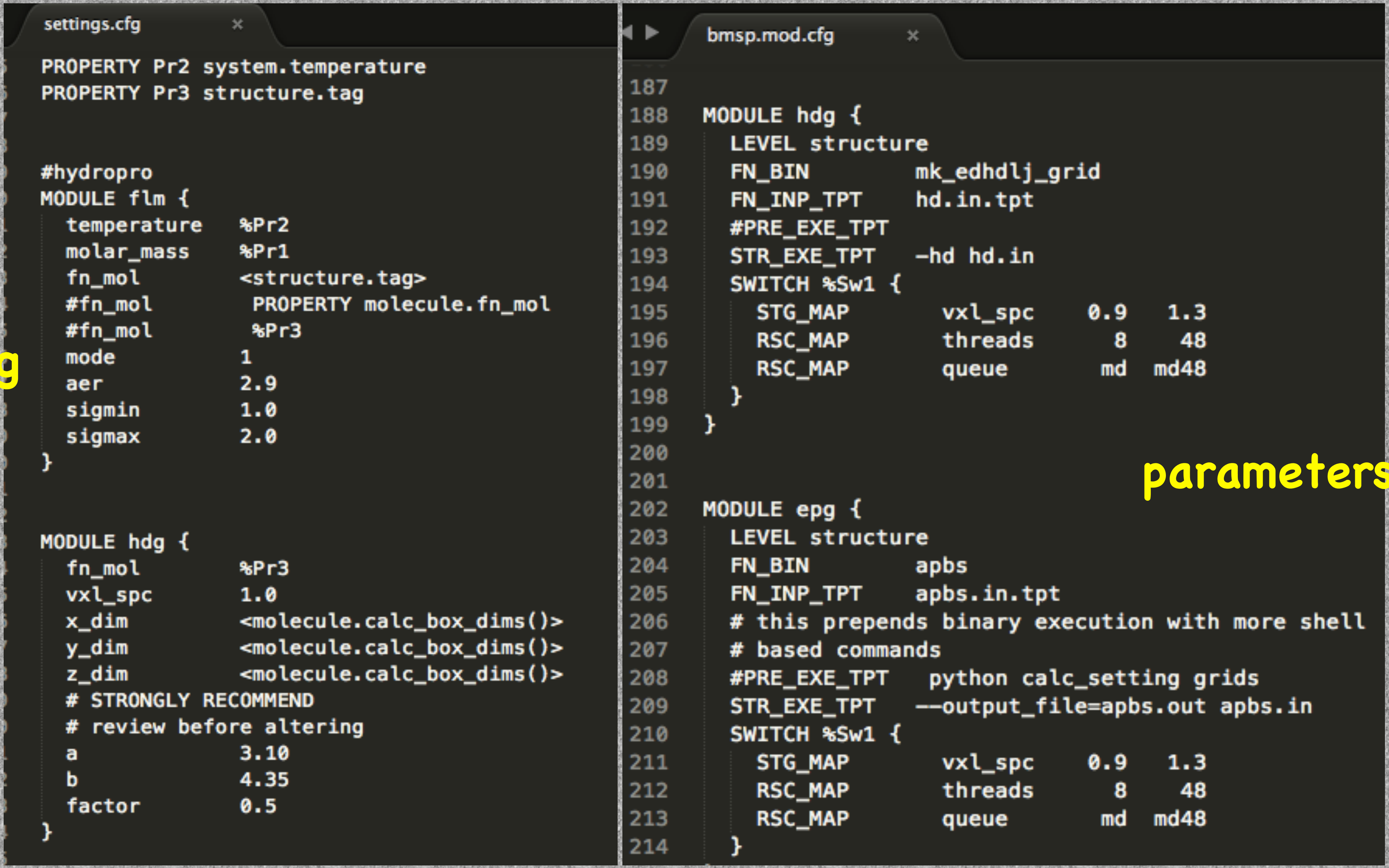
```
group8:
{
    order=("MODULE-11")
    iterations=1
}
group9:
{
    order=("MODULE-12")
    iterations=1
    depends=("group1","group2","group3","group4","group5",
    oup6","group7","group8")
}
```

```
modules=(
    {
        function="ielTupleServer";
        args=();
        libtype="static";
        library="libIELexec.a";
        size=1;
    }
    {

        function="MODULE-0";
        args=("../executables/s1a.sh");
        libtype="static";
        splitdir="pdb2pqr"
        size=1
        points=(
            (())
        );
```

Fig.4, details of the configure file

GUI V1—configure file & input files(cont’d

About Setting files:



```
PROPERTY Pr2 system.temperature
PROPERTY Pr3 structure.tag


#hydropro
MODULE flm {
  temperature   %Pr2
  molar_mass    %Pr1
  fn_mol        <structure.tag>
  #fn_mol        PROPERTY molecule.fn_mol
  #fn_mol        %Pr3
  mode          1
  aer           2.9
  sigmin        1.0
  sigmax        2.0
}


MODULE hdg {
  fn_mol        %Pr3
  vxl_spc       1.0
  x_dim         <molecule.calc_box_dims()>
  y_dim         <molecule.calc_box_dims()>
  z_dim         <molecule.calc_box_dims()>
  # STRONGLY RECOMMEND
  # review before altering
  a             3.10
  b             4.35
  factor        0.5
}
```

```

MODULE hdg {
  LEVEL structure
  FN_BIN        mk_edhdlj_grid
  FN_INP_TPT    hd.in.tpt
  #PRE_EXE_TPT
  STR_EXE_TPT   -hd hd.in
  SWITCH %Sw1 {
    STG_MAP       vxl_spc    0.9   1.3
    RSC_MAP       threads      8    48
    RSC_MAP       queue       md  md48
  }
}


MODULE epg {
  LEVEL structure
  FN_BIN        apbs
  FN_INP_TPT    apbs.in.tpt
  # this prepends binary execution with more shell
  # based commands
  #PRE_EXE_TPT    python calc_setting grids
  STR_EXE_TPT   --output_file=apbs.out apbs.in
  SWITCH %Sw1 {
    STG_MAP       vxl_spc    0.9   1.3
    RSC_MAP       threads      8    48
    RSC_MAP       queue       md  md48
  }
```

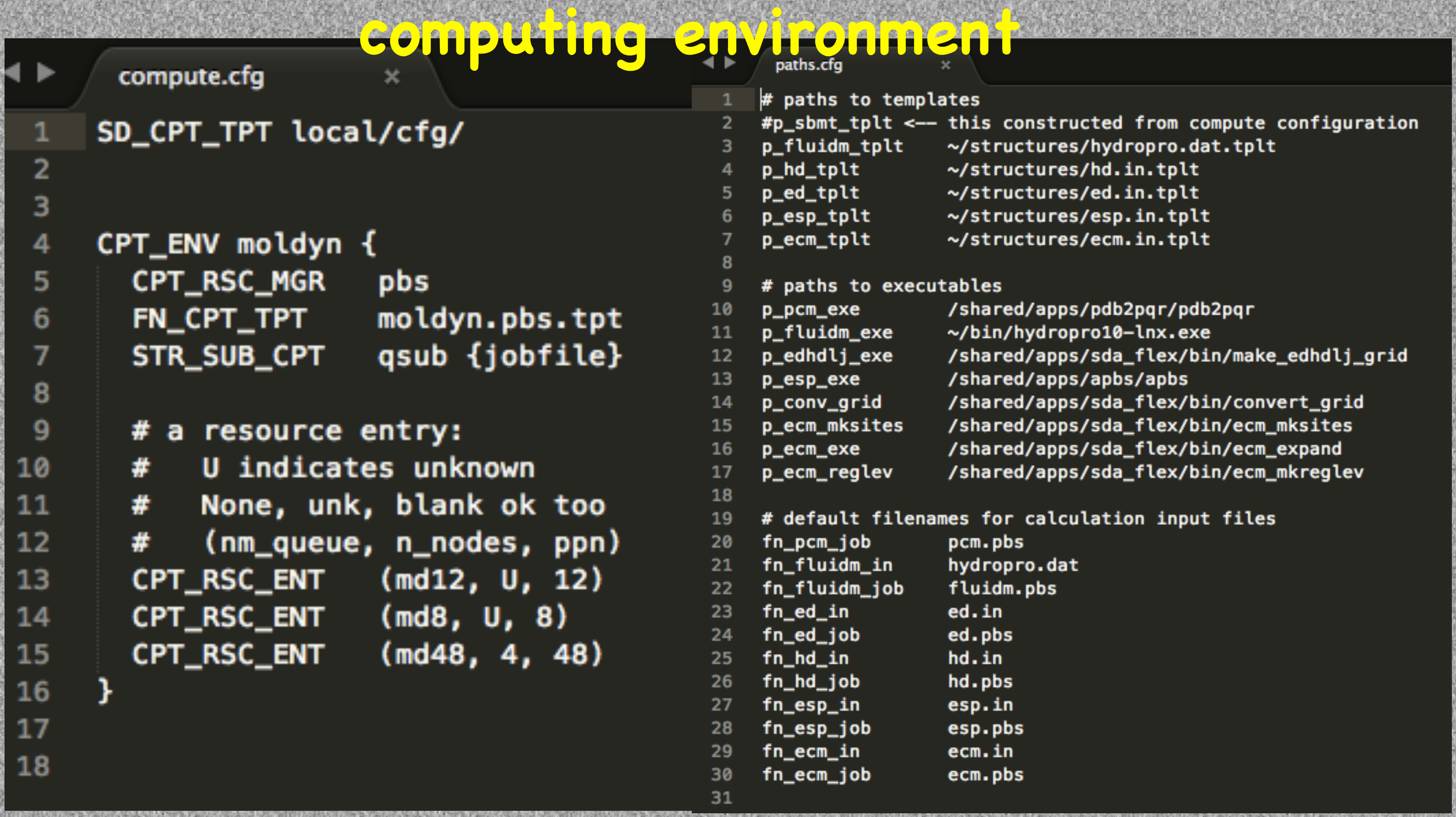
Values setting

parameters setting

Fig.5, the setting files to generate input files

GUI V1—configure file & input files(cont'd

About Setting files:



computing environment

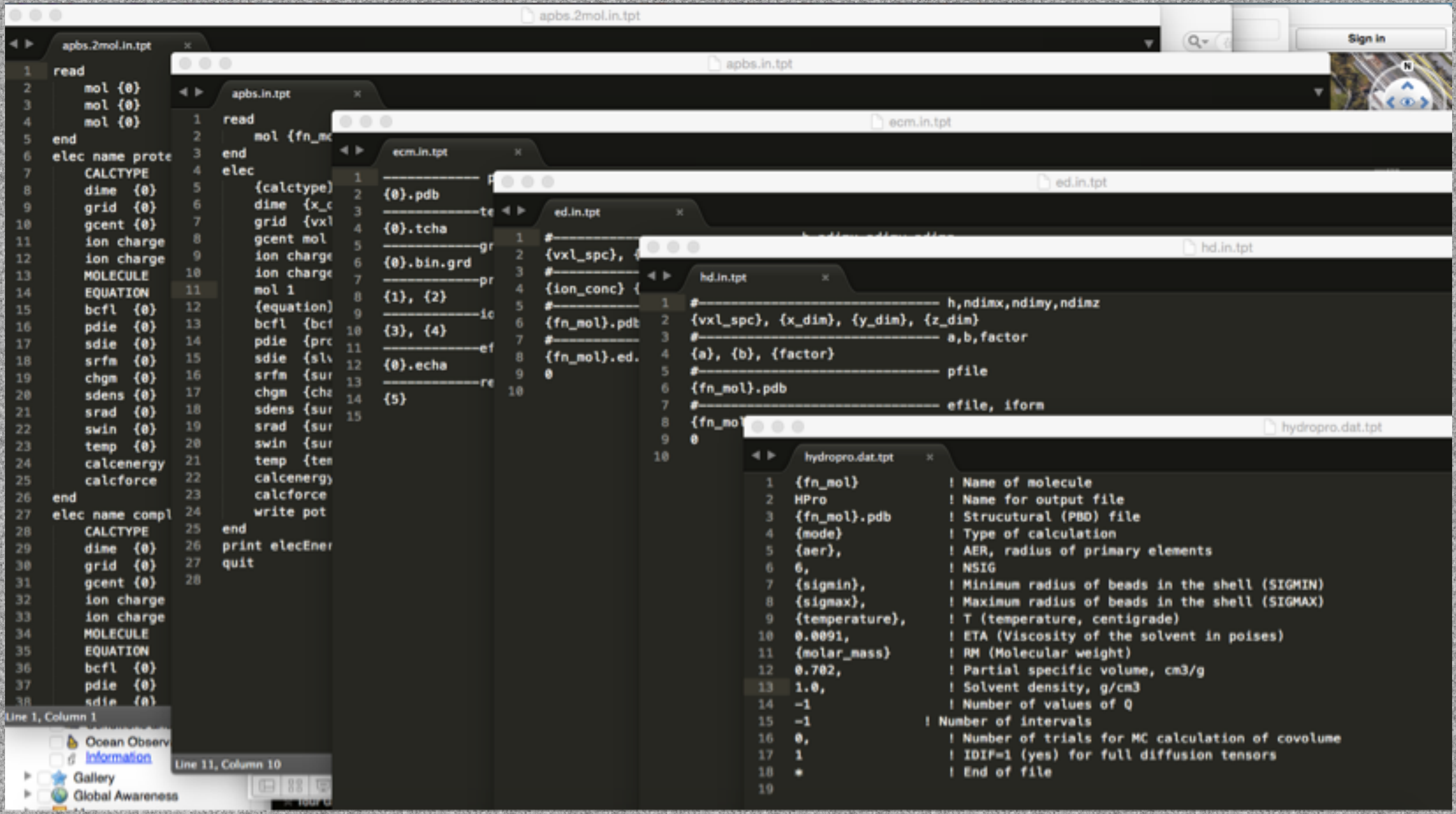
```
SD_CPT_TPT local/cfg/


CPT_ENV moldyn {
  CPT_RSC_MGR    pbs
  FN_CPT_TPT     moldyn.pbs.tpt
  STR_SUB_CPT    qsub {jobfile}

  # a resource entry:
  #   U indicates unknown
  #   None, unk, blank ok too
  #   (nm_queue, n_nodes, ppn)
  CPT_RSC_ENT    (md12, U, 12)
  CPT_RSC_ENT    (md8, U, 8)
  CPT_RSC_ENT    (md48, 4, 48)
}
```

```
# paths to templates
#p_sbmt_tplt <-- this constructed from compute configuration
p_fluidm_tplt    ~/structures/hydropro.dat.tplt
p_hd_tplt        ~/structures/hd.in.tplt
p_ed_tplt        ~/structures/ed.in.tplt
p_esp_tplt       ~/structures/esp.in.tplt
p_ecm_tplt       ~/structures/ecm.in.tplt

# paths to executables
p_pcm_exe        /shared/apps/pdb2pqr/pdb2pqr
p_fluidm_exe     ~/bin/hydropro10-lnx.exe
p_edhdlj_exe     /shared/apps/sda_flex/bin/make_edhdlj_grid
p_esp_exe        /shared/apps/apbs/apbs
p_conv_grid      /shared/apps/sda_flex/bin/convert_grid
p_ecm_mksites    /shared/apps/sda_flex/bin/ecm_mksites
p_ecm_exe        /shared/apps/sda_flex/bin/ecm_expand
p_ecm_reglev     /shared/apps/sda_flex/bin/ecm_mkreglev

# default filenames for calculation input files
fn_pcm_job       pcm.pbs
fn_fluidm_in     hydropro.dat
fn_fluidm_job    fluidm.pbs
fn_ed_in         ed.in
fn_ed_job        ed.pbs
fn_hd_in         hd.in
fn_hd_job        hd.pbs
fn_esp_in        esp.in
fn_esp_job       esp.pbs
fn_ecm_in        ecm.in
fn_ecm_job       ecm.pbs
```

Fig.6, the setting file to store machine information

GUI V1—configure file & input files(cont’d

About input files for BD simulation:



```
read
    mol {0}
    mol {0}
    mol {0}
end
elec name prote
    CALCTYPE
    dime  {0}
    grid  {0}
    gcent {0}
    ion charge
    ion charge
    MOLECULE
    EQUATION
    bcfl  {0}
    pdie  {0}
    sdie  {0}
    srfm  {0}
    chgm  {0}
    sdens {0}
    srad  {0}
    swin  {0}
    temp  {0}
    calcenergy
    calcforce
end
elec name compl
    CALCTYPE
    dime  {0}
    grid  {0}
    gcent {0}
    ion charge
    ion charge
    MOLECULE
    EQUATION
    bcfl  {0}
    pdie  {0}
    sdie  {0}
```

```
read
    mol {fn_mo
end
elec
    {calctype}
    dime  {x_c
    grid  {vxl
    gcent mol
    ion charge
    ion charge
    mol 1
    {equation}
    bcfl  {bc
    pdie  {pro
    sdie  {sl
    srfm  {sur
    chgm  {cha
    sdens {sur
    srad  {sur
    swin  {sur
    temp  {tem
    calcenergy
    calcforce
    write pot
end
print elecEner
quit
```

```
--------------- p
{0}.pdb
-----------------te
{0}.tcha
-----------------gr
{0}.bin.grd
-----------------pr
{1}, {2}
-----------------io
{3}, {4}
-----------------ef
{0}.echa
-----------------re
{5}
```

```
#-----------
{vxl_spc}, {
#-----------
{ion_conc} {
#-----------
{fn_mol}.pdb
#-----------
{fn_mol}.ed.
0
```

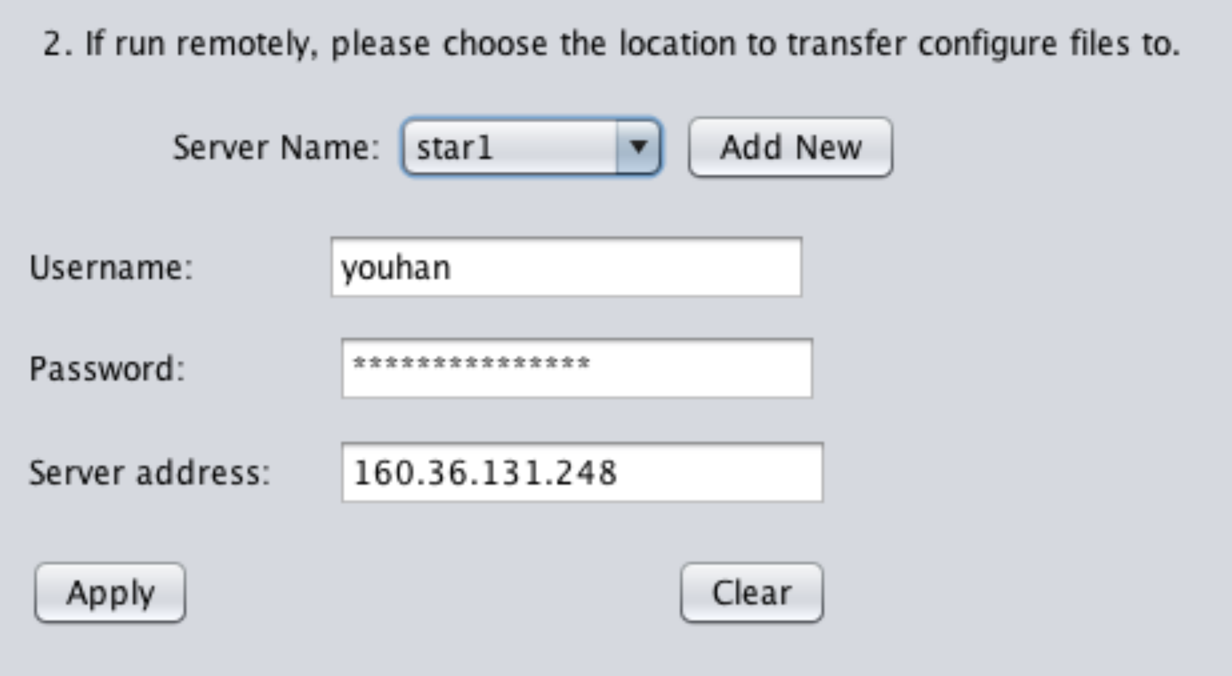
```
#------------------------------ h,ndimx,ndimy,ndimz
{vxl_spc}, {x_dim}, {y_dim}, {z_dim}
#------------------------------ a,b,factor
{a}, {b}, {factor}
#------------------------------ pfile
{fn_mol}.pdb
#------------------------------ efile, iform
{fn_mo
0
```

```
{fn_mol}            ! Name of molecule
HPro                ! Name for output file
{fn_mol}.pdb        ! Strucutural (PBD) file
{mode}              ! Type of calculation
{aer},              ! AER, radius of primary elements
6,                  ! NSIG
{sigmin},           ! Minimum radius of beads in the shell (SIGMIN)
{sigmax},           ! Maximum radius of beads in the shell (SIGMAX)
{temperature},      ! T (temperature, centigrade)
0.0091,             ! ETA (Viscosity of the solvent in poises)
{molar_mass}        ! RM (Molecular weight)
0.702,              ! Partial specific volume, cm3/g
1.0,                ! Solvent density, g/cm3
-1                  ! Number of values of Q
-1                ! Number of intervals
0,                  ! Number of trials for MC calculation of covolume
1                   ! IDIF=1 (yes) for full diffusion tensors
*                   ! End of file
```

Fig.5, the input files to print for running BD simulation

GUI V2— supplements of GUI V1

Transfer files to server:


2. If run remotely, please choose the location to transfer configure files to.

Server Name: star1

Username: youhan

Password: ****************

Server address: 160.36.131.248

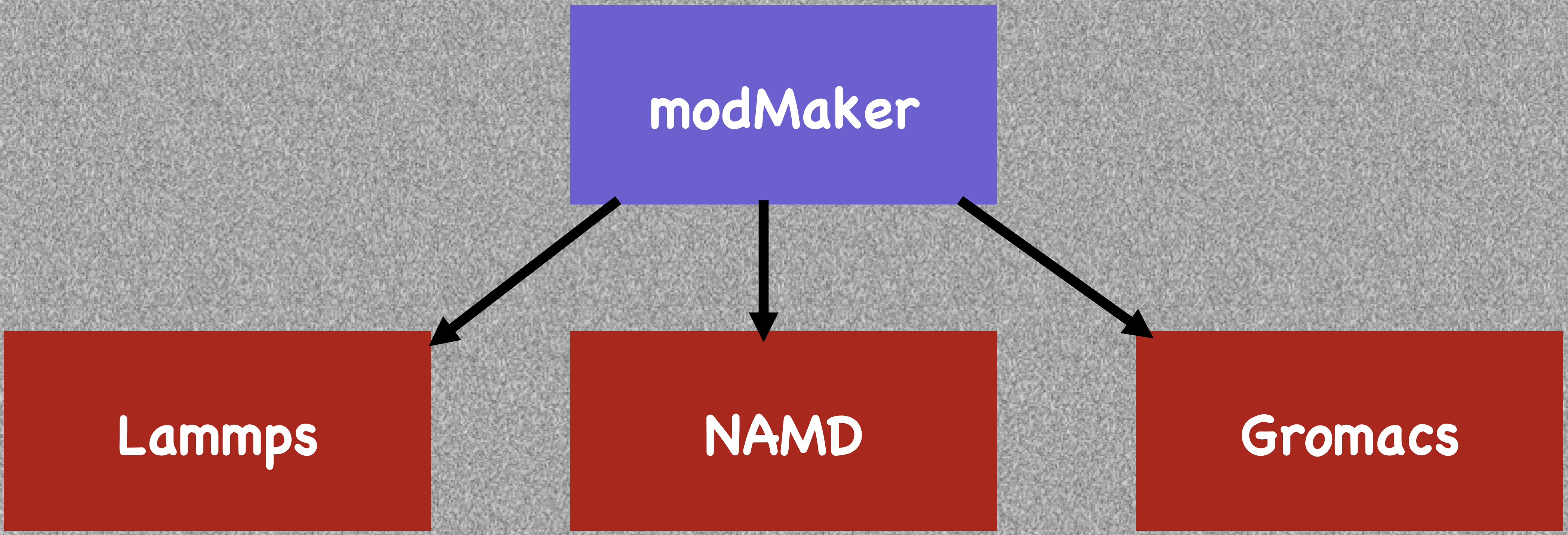
Fig.5 , transfer file to server

Targets:

- Run simulation from GUI
- Get outputs back to GUI
- Display it for post-process

GUI V2—Parallel Version(in process)

GUI V1 focuses mainly on running BDS and configuring the pre-calculation modules to optimize models in the simulation, requiring iteration and other small changes within the workflowso GUI V2 for running biophysical simulations in parallel is on the way.



Demonstration

Acknowledgements

This project is supported by University of Tennessee, NSF, and Oak Ridge National Laboratory.

JICS
Joint Institute for Computational Sciences
UT ORNL Computational Sciences

CityU
香港城市大學
City University of Hong Kong
專業 創新 胸懷全球
Professional · Creative
For The World

OAK RIDGE
National Laboratory

NSF

THE UNIVERSITY of TENNESSEE UT
KNOXVILLE