Introduction

Contents

Welcome to the comprehensive ***Solvate Suite User Guide***! The *Solvate Suite* is a modular and integrated toolkit designed to streamline the preparation and processing of molecular systems with explicit solvents, bridging molecular dynamics (MD) simulations with quantum chemistry calculations. This package was conceived to meet the needs of researchers working at the interface between classical mechanics and electronic structure methods, especially in contexts that require a more realistic modeling of solvated environments, such as reaction mechanisms in solution, spectroscopic studies, and thermochemical properties.

With a flexible and highly customizable approach, *Solvate Suite* allows you to:

- Create initial configurations for AIMD/BOMD simulations with *ORCA* or classical MD simulations with *GROMACS*;
- Generate solvation boxes with various solvents and force fields (OPLS-AA, GROMOS, CHARMM, and AMBER) from diverse sources (*GROMACS* database, LigParGen, ATB, or *Q-Force*);
- Integrate data from molecular dynamics simulations with input files for electronic structure calculations (semiempirical, DFT, MP2, etc.);
- Automate repetitive steps such as snapshot extraction, and input preparation for software like *Gaussian*, *ORCA*, *xTB*, and *GROMACS*;
- Perform statistical and structural analyses of solvent environments around target species.

This manual was written to guide users — from beginners to advanced — through the efficient use of *Solvate Suite*'s features. Throughout the chapters, you will find step-by-step instructions, practical examples, useful tips, and conceptual explanations that help you understand not only the "how", but also the "why" of each step.

Get ready to accelerate your computational workflow and explore complex systems with greater control, reproducibility, and speed.

Citation

- ***Solvate Suite*: A Command-Line Interface for Molecular Simulations and Multiscale Microsolvation Modeling** Otávio L. Santana; Daniel G. Silva; Sidney R. Santana *J. Chem. Inf. Model. 2024, 64, 9, 3767–3778* (https://pubs.acs.org/doi/10.1021/acs.jcim.3c01899)

1. **Configuration**

SETTING UP MD SIMULATIONS

In the **first stage**, *Solvate Suite* provides optimized protocols for setting up the initial configuration of simulation boxes, enabling AIMD/BOMD simulations with *ORCA* or classical MD simulations with *GROMACS*. The construction of the initial simulation box is handled by the `PACKS` module, through the iterative use of the *PackMol* program. In the case of classical MD simulations, this process includes creating or adjusting force field parameters and building the topology input file. The generation of force field parameters is managed by the `GMDFF` module, which produces parameter files (itp) and applies specific adjustments to the topology file (top) based on various input formats, such as files generated by *Gaussian*, *ORCA*, *xTB*, or *GROMACS*.

> **Note**
>
> Although the PACKS module manages the use of GMDFF, enabling the generation of force field parameters during the packing process, it is recommended to run the GMDFF module beforehand. Regardless, the PACKS module will verify the consistency of the parameters at the end of the packing procedure.

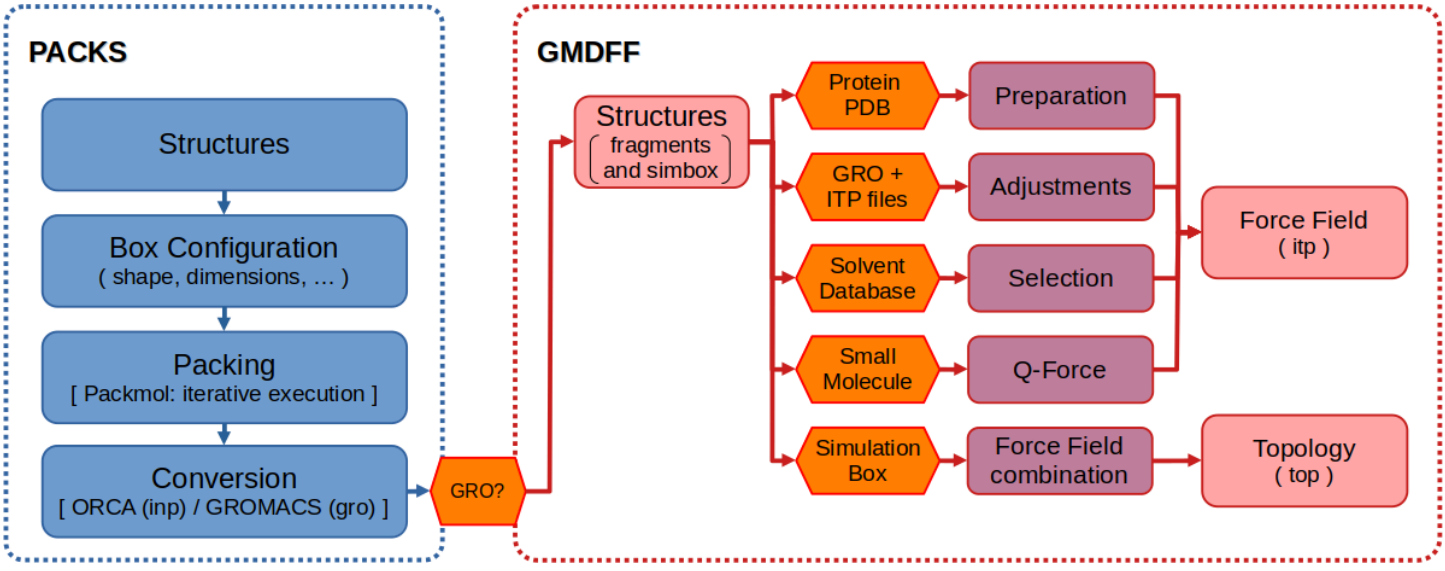
The main program can control the use of the core modules, allowing uniform execution throughout all stages of the suite's use. For example, the suite's first stage involves creating and configuring the simulation box. This is done using the following basic command structure:

> solvate ⟨solute.ext⟩ ⟨solvent.ext⟩ [⟨-options⟩]

where *ext* is one of the extensions recognized by *Solvate Suite* (com/log [*Gaussian*], inp/out [*ORCA*], xyz/xtb [*xTB*], gro/pdb [*GROMACS*]), *inputN* the files with the structures of the solute and solvent molecules, and "-options" a set of options for the configuration of the box to be created.

> **Note**
>
> The list of options can be seen from the applications menu, which is accessible from the command line by typing the module name, without any other parameters.

STAGE STRUCTURE



1.1. PACKS

1.1.1. Simulation Box Creation

The `PACKS` module mediates the creation of the simulation box, through the iterative use of the *PackMol* program. The module can be run from the main program or directly from the command:

> packs ⟨input1.ext⟩ ⟨input2.ext⟩ {…} [⟨-options⟩]

where *ext* is one of the extensions recognized by *Solvate Suite*, *inputN* the files with the structures of the solute and solvent molecules, and “-options” a set of options for the configuration of the box to be created. The program accepts input and output file formats of *Gaussian* (com, gjf and log), *ORCA* (inp and out), xTB (xyz and xtb), *GROMACS* (gro) and *PDB* (pdb), common formats in Computational Chemistry.

> **Note**
>
> Simply type `packs` in the terminal and press *enter* to see the full list of execution options for the module.

The most common use of this module is to create an initial configuration for a system with a single solute molecule surrounded by a number of solvent molecules of a single type:

> packs ⟨solute.ext⟩ ⟨solvent.ext⟩ -mol *N*

where *N* corresponds to the number of solvent molecules. In the case of ionic solutes, it is possible to include counter-ions with the command:

> packs ⟨solute.ext⟩ ⟨solvent.ext⟩ -mol *N* -cnt couterion.ext

The program determines the number of counterions needed to neutralize the charge in the simulation box. In these cases, the number *N* of molecules includes the incorporated counterions. The *Solvate* program also makes it possible to create simulation boxes with solvent mixtures, in which case it is necessary to inform the proportion between the molecules:

> packs ⟨solute.ext⟩ ⟨solvent1.ext⟩ ⟨solvent2.ext⟩ … -pro [*N1*:*N2*:*N3*…]

The initial box can also be created based on the molar concentration *M* of the solute (a situation in which there is more than one solute molecule in the box):

> packs ⟨solute.ext⟩ ⟨solvent.ext⟩ -cco *M*

or based on molal concentration *M*:

> packs ⟨solute.ext⟩ ⟨solvent.ext⟩ -bco *M*

The box density *D* is automatically adjusted based on the density of the solvent used (if it is part of the program's database), or defined manually by the user:

> packs ⟨solute.ext⟩ ⟨solvent.ext⟩ -mol *N* -den *D*

The list of parameterized solvents is obtained from the command:

> packs -lst

Any of the parameterized solvents can be provided as input to the program in the simulation box creation step.

Several additional options for adjusting packaging parameters are available. The complete list can be accessed from the help menu of the PACKS module.

1.1.2. Execution Options

> packs ⟨solute⟩ {⟨solvent⟩} [⟨-options⟩]

```
  ------------------------------------------------------------------------------
 ⟨solute⟩ = File with solute  geometry (.com/log/inp/out/xyz/xtb/gro/pdb)
⟨solvent⟩ = File with solvent geometry (.com/log/inp/out/xyz/xtb/gro/pdb)
  ------------------------------------------------------------------------------
     -mol = Number of solvent molecules in the packing step.           [Default: 0][
     -npk = Number of packing layers.                                  [Default: 2]
     -cel = Dimensions of the simulation cell (in nanometers).         [Default: Not
     -gap = Gap between solute and cell edges (in nanometers).         [Default: Not
     -bco = Molal concentration of the simulation cell (in mol·kg⁻¹).  [Default: Not
     -cco = Molar concentration of the simulation cell (in mol·L⁻¹).   [Default: Not
     -pro = Proportion between solvent molecules. [N1:N2:...]          [Default: Not
     -den = Density of the simulation cell (in g·mL⁻¹).                [Default: 1.0
     -cnt = File with counterion to be added for charge neutrality.    [Default: Nor
     -shp = Packaging format (sphere|cube|box).                        [Default: Sph
     -fmt = Type of generated file (inp|gro).                          [Default: Aut
     -tol = Minimum distance tolerated in packing (in Å).              [Default: Not
     -rep : Resubmit the packing procedure with new parameters.        [Default: No]
     -for : Run in forced mode (no warning messages).                  [Default: No]
     -nor : Run in normal mode (conventional packing configuration).   [Default: No]
     -ite : Run in iterative mode (to adjust tolerance).               [Default: No]
     -inc : Insert external force field into generated topologie.      [Default: No]
     -fft = Force field parameters.      [OPL|GRO|CHM|AMB|LPG|QFC|ATB] [Default: OPL
     -bak : Save original input files.                                 [Default: No]
     -svt = Solvent to be used in C-PCM/ALPB solvation model.          [Default: Aut
     -lst : Lists the set of solvents in the database.                 [Default: No]
     -chk : Checks the consistency of simulation box settings.         [Default: No]
  ------------------------------------------------------------------------------
  (*) Configurable options in automated mode. (†) GROMACS only
```

1.1.3. Quickstart Examples

- Example #1: Creating a box for *ORCA* with a solute and water molecules
- Example #2: Creating a box for *ORCA* with a solute and solvent mixture
- Example #3: Creating a box for *GROMACS* with a solute and water molecules
- Example #4: Creating a box for *GROMACS* with a solute and a solvent mixture

1.2. GMDFF

1.2.1. MD Force Fields

The `GMDFF` module mediates the creation of force field parameters (itp files), as well as making special adjustments for the construction of the topology (top file). The simplest usage consists of the

following command:

> gmdff ⟨input.ext⟩

where *ext* is one of the extensions recognized by the *Solvate Suite* (com/log [*Gaussian*], inp/out [*ORCA*], xyz/xtb [*xTB*], gro/pdb [*GROMACS*]).

> **Note**
>
> Simply type `gmdff` in the terminal and press *enter* to see the full list of execution options for the module.

If the simulation box is prepared for the *GROMACS* program (which is determined by the extension of the input files) the program automatically runs the GMDFF module, which prepares the force field files as well as the system topology file. This means that, in the general case, it is possible to create the simulation box and its topology file from a single command line. In these cases, however, it is recommended to run the GMDFF module first, in order to prepare all the necessary force field files required for executing the PACKS module. This is done using the following command structure:

> gmdff ⟨input.ext⟩ [⟨-options⟩]

where *ext* is one of the extensions recognized by Solvate Suite (com/log [*Gaussian*], inp/out [*ORCA*], xyz/xtb [*xTB*], gro/pdb [*GROMACS*]), *input* is the file with the structure of the solute or solvent molecule, and “-options” a set of options for the configuration of the force field to be created.

There are 4 scenarios in which this module can be used:

- In the case of biomolecule PDB files, the program performs structure preparation (removal of heteroatoms and addition of hydrogens, without adjusting protonation based on pH, which may require additional preparations to be carried out by the user), followed by conversion to the *GROMACS* format (gro file) and creation of the force field parameters in a single input (itp file).
- In the case of force fields obtained by external servers (such as LigParGen and ATB) the module removes any residual charges, a necessary step for building simulation boxes that are, in fact, electrically neutral. This is done by homogeneously compensating for any residual charge between all atoms in the structure.

- In the case of internally parameterized solvent molecules, the program selects the appropriate parameters and writes them to a force field input (itp file), as well as converting the input file to the appropriate format (gro file).
- In the case of small molecules, the module sequentially executes the *Q-Force* program to generate parameters from a quantum chemical calculation.

> **Note**
>
> In the case of using the *Q-Force* program, the user can generate a default settings file with the `-chk` option. The edited file can then be read by the program with the `-cfg` option. Alternatively, the user can provide his own settings file with the `-cfg ⟨file.ini⟩` command.

For *Q-Force*, it is possible to check whether the options are appropriate to start the force field creation:

```
gmdff ⟨input.ext⟩ -chk
```

In this case, a configuration file (ini) is generated with a default set of execution options, which can be edited by the user to apply custom configurations. This file can then be read using the following command:

```
gmdff ⟨input.ext⟩ -cfg
```

Alternatively, the user can provide a custom settings file via the following command:

```
gmdff ⟨input.ext⟩ -cfg config.ini
```

Several force field parameters (OPLS/AA, GROMOS, CHARMM and AMBER) for a wide range of solvents are part of the program database, coming from several sources (GROMACS repository, LigParGen, ATB and Q-Force). The complete list of parameters can be displayed from the command:

```
gmdff -lst
```

It is possible to display only the parameters of a specific force field, such as OPLS/AA:

> gmdff -lst opls

Once you have identified the desired parameters, you can obtain them from the command:

> gmdff ⟨solvent_forcefield⟩

For example, the TIP3P parameters in the OPLS/AA force field are obtained from:

> gmdff tip3p_oplsaa.wtr

Finally, if the input file matches that of the simulation box, the program combines the force field files into a single and independent topology file. This results in only two files for starting the simulation: the simulation box (gro file) and the topology (top file, with all the force field parameters).

1.2.2. Execution Options

> gmdff ⟨molecule.ext⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   .ext = Input file extension (.com/log/inp/out/xyz/xtb/pdb/gro)
        + ----------------------------------------------------------
        • Q-Force Options
   -chr = Molecular charge.                                        [Default: 0]
   -mul = Molecular multiplicity.                                  [Default: 1]
   -pcm = Solvent for C-PCM model.                                 [Default: Nor
   -chk : Verify generated configuration file before submit.       [Default: No]
   -cfg = Execute with user defined input configurations.          [Default: No]
   -prc = Number of processors to be used.                         [Default: 4]
   -mem = Amount of memory in GB to be used.                       [Default: 8]

        • Additional Options
   -fld = Force field parameters.      [OPL|GRO|CHM|AMB|LPG|QFC|ATB] [Default: OPL
   -wtr = Water model (conditional).   [SPC|SPCE||TIP3P|TIP4P|TIP5P] [Default: SPC
   -lst : Provides the complete list of parameterized solvents.    [Default: No]
   -for : Execute in forced mode.                                  [Default: No]
-------------------------------------------------------------------------------
Total CPUs: 12 (max. MPI-procs.) :: System Memory: 5 GB
-------------------------------------------------------------------------------
(†) Specifies the model to use if the input file corresponds to a water molecule.
(*) Repositories: https://traken.chem.yale.edu/ligpargen
                  https://atb.uq.edu.au
                  https://cgenff.com
```

1.2.3. Quickstart Examples

- Example #1: Preparing a protein PDB file
- Example #2: Creating the Q-Force force field parameters
- Example #3: Handling force field parameters from external servers
- Example #4: Handling force field parameters from internal database

2. **Simulation**

EXECUTING MD SIMULATIONS

In the **second stage**, the *Solvate Suite* provides a group of modules to manage the MD simulations. From the `MDRUN` module, the simulations with the *ORCA* and *GROMACS* programs are executed using the same set of command lines. In the case of *ORCA*, the simulation execution options are set directly in the input (inp file) at run time on the command line. In the case of *GROMACS*, the options are adjusted and saved in its own input settings (mdp file). In the latter case, there are more

simulation execution options, such as the type of barostat and thermostat, time constant for pressure and temperature coupling, isothermal compressibility, and saving frequency, among others. In any case, the simulation properties can be extracted and analyzed from the `DATAS` module, which has resources for statistical and graphical analysis.

> **Note**
>
> The modular design allows users to run each step of the workflow separately, enabling detailed control and progress evaluation at any point in the simulation.

The second stage of using the suite consists of executing and analyzing the simulation. The execution can be done using the following command structure:

```
solvate ⟨simbox.ext⟩ -mdrun [⟨-options⟩]
```

where *ext* is one of the extensions recognized by *Solvate Suite* (inp [*ORCA*], or gro [*GROMACS*]), *simbox* is the simulation box file in it's initial configuration, and "-options" a set of options for the simulation control. After each step performed (minimization, equilibration, and production), the data can be analyzed using the following command structure:

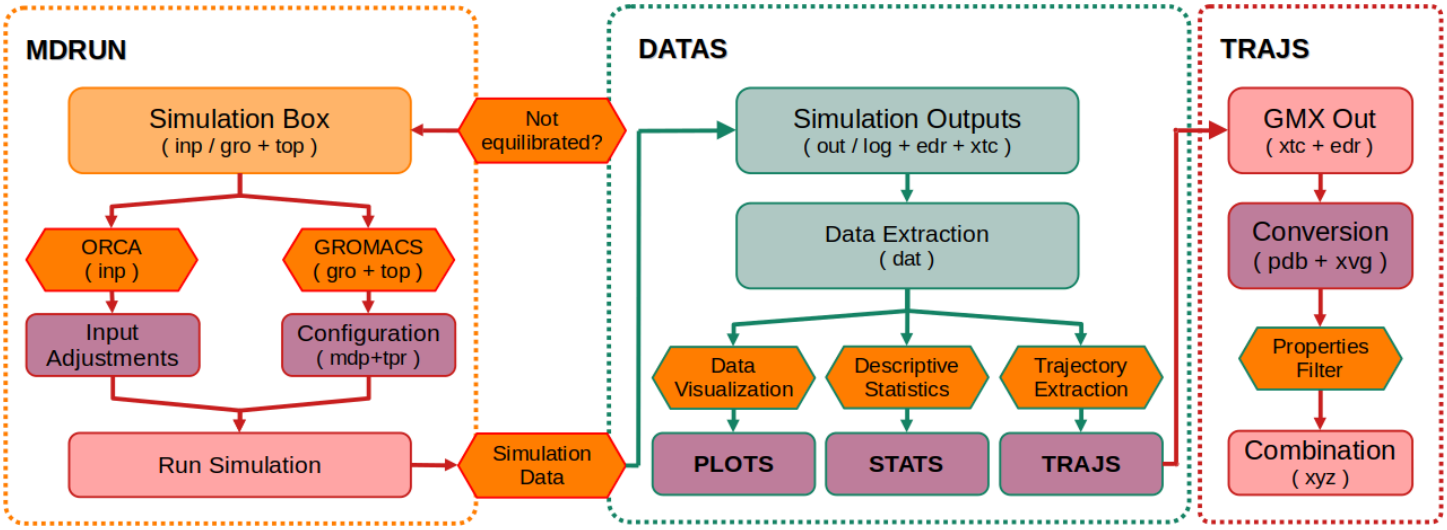
```
solvate ⟨simbox.ext⟩ -datas [⟨-options⟩]
```

where "-options" is a set of options for the simulation data analysis.

Several options for adjusting simulation parameters, as well as analyzing properties and extracting data, are available for the *ORCA* and *GROMACS* programs. The complete list can be accessed in the help menu of the MDRUN and DATAS modules.

> **Note**
>
> The list of options can be seen from the applications menu, which is accessible from the command line by typing the module name, without any other parameters.

STAGE STRUCTURE



2.1. MDRUN

2.1.1. MD Execution

The `MDRUN` module manages the execution of simulations, through a unified interface for the *ORCA* and *GROMACS* programs. The module can be run from the main program or directly from the command:

> mdrun ⟨simbox.ext⟩ [⟨-options⟩]

where *ext* is one of the extensions recognized by *Solvate Suite* (`inp` [*ORCA*] or `gro` [*GROMACS*]), and “-options” a set of the configurations for the simulation execution.

Note

Simply type `mdrun` in the terminal and press *enter* to see the full list of execution options for the module.

> **Note**
>
> In the case of using the *GROMACS* program, the user can generate a default settings file with the `-chk` option. The edited file can then be read by the program with the `-cfg` option. Alternatively, the user can provide his own settings file with the `-cfg ⟨file.mdp⟩` command.

For *GROMACS*, it is possible to check whether the options automatically configured after packaging are appropriate to start the simulation:

```
mdrun ⟨simbox.gro⟩ -chk
```

In this case, a configuration file (mdp) is generated with a default set of execution options, which can be edited by the user to apply custom configurations. This file can then be read using the following command:

```
mdrun ⟨simbox.gro⟩ -cfg
```

Alternatively, the user can provide a custom settings file via the following command:

```
mdrun ⟨simbox.gro⟩ -cfg config.mdp
```

Before starting the simulation, it is necessary to carry out an initialization step for the initial box, which includes energy minimization and box optimization. For example, to initialize the simulation box at a pressure of 1.0 bar and a temperature of 298.15 K, you can use the following command:

```
mdrun ⟨simbox.gro⟩ -ini -prs 1.0 -tmp 298.15
```

After initialization, temperature equilibration can be performed in an NVT ensemble. For example, running 1.0 ns for equilibration in the NVT ensemble can be executed with the following command:

```
mdrun ⟨simbox.gro⟩ -nvt -tmp 298.15 -tot 1.0
```

and the pressure equilibration, in an NPT ensemble, with:

> mdrun ⟨simbox.gro⟩ -npt -prs 1.0 -tmp 298.15 -tot 1.0

The equilibration steps involve saving the trajectory every 100 steps (in a trr file) to conserve disk space.

> **Note**
>
> To assess the equilibrium, the properties acquired during the simulation can be analyzed using the `DATAS` module.

After achieving equilibration, the production step can proceed. In this step, the trajectory is saved at each step of the simulation (in an xtc file). Production is run with the following command:

> mdrun ⟨simbox.gro⟩ -prd -prs 1.0 -tmp 298.15 -tot 5.0

> **Note**
>
> The trajectory extraction procedure can be performed using the `DATAS` module, which has several extraction filters available.

Finally, after extraction, it is possible to remove several files that are unnecessary for the final step of the microsolvation treatment, which can be done with the command:

> mdrun ⟨simbox.gro⟩ -cls

2.1.2. Execution Options

> mdrun ⟨file.ext⟩ [⟨-options⟩]

```
--------------------------------------------------------------------------------
 ORCA/GROMACS
--------------------------------------------------------------------------------
   auto : Execute the MD simulation with automatic settings.
        » Description
          ORC: [Default: -tot = 50.0 ps (prod.) | -dts = 1.0 fs]
          GMX: [Default: -tot = 2.50 ns (prod.) | -dts = 2.0 fs]
        + ------------------------------------------------------
» Automation Options:
  -auto : Start MD simulation with automatic settings.
  -init : Init. energy minimization with simulation box optimization procedure.
  -equi : Exec. MD equilibration procedure (with auto restart).
  -prod : Exec. MD production procedure (with auto restart).
  -bomd : Run a BOMD simulation at GFN2-xTB level.
» Procedure Options:
   -min : Execute an energy minimization of the simulation box.
   -opt : Execute an optimization of the simulation box.
   -nvt : Run an equilibration simulation on NVT ensamble.
   -npt : Run an equilibration simulation on NPT ensamble.
   -prd : Run a production simulation on NVT/NPT ensamble.
» Settings Options:
   -met = MD simulation method (GFF|GF1|GF2).                        [Default: GFF
   -prs = Pressure for the MD simulation (in bar).                   [Default: 1.0
   -tcp = Time constant for pressure coupling (in ps).               [Default: 0.5
   -bar = Type of barostat pressure coupling.                        [Default: Ber
          * BRDS = Berendsen | CRES = C-rescale | PARR = Parrinello  | MTTK
   -tmp = Temperature for MD simulation (in K).                      [Default: 298
   -tct = Time constant for temperature coupling (in ps).            [Default: 0.5
   -ter = Type of thermostat temperature coupling.                   [Default: V-r
          * BRDS = Berendsen | VRES = V-rescale | NOSE = Nose-Hoover | ADSN = Ande
   -kap = Isothermal compressibility value (in bar⁻¹).               [Default: 1.0
   -nst = Simulation steps to update neighbor list.                  [Default: 100
   -dmp = Number of steps for dump positions.                        [Default: 1]
   -dts = Simulation time step (in fs).                              [Default: 1.0
   -tot = Simulation additional (ORCA) or total (GMX) time (in ps).  [Default: Aut
   -nrs = Number of simulation run steps.                            [Default: Aut
» Additional Options:
   -prc = Number of parallel CPU processes to run.                   [Default: 4]
   -chk : Check generated files before submit.                       [Default: No]
   -cfg = Execute with user defined input configurations.            [Default: No]
   -fix : Performs MD without atom constraints.                      [Default: No]
   -nhr : Exec. with no H-bonds. restraints (for production steps).  [Default: No]
   -clr : Cleaning unnecessary files (after extraction).             [Default: No]
   -cpt : Compact output files.                                      [Default: No]
   -ext : Extract output files.                                      [Default: No]
   -for : Execute in forced mode (suppressing upper to 2 warnings).  [Default: No]
--------------------------------------------------------------------------------
(*) Configurable options in automated mode. (●) ORCA available. (■) GROMACS availa
```

2.1.3. Quickstart Examples

- Example #1: Submitting a simulation with ORCA
- Example #2: Submitting a simulation with GROMACS

2.2. DATAS

2.2.1. MD Analysis

The `DATAS` module performs graphical and statistical analysis of simulated properties, through a unified interface for the *ORCA* and *GROMACS* programs. The module can be run from the main program or directly from the command:

> datas ⟨simbox.ext⟩ [⟨-options⟩]

where *ext* is one of the extensions recognized by *Solvate Suite* (out [*ORCA*], or log [*GROMACS*]), and “-options” a set of the configurations for the simulation execution.

> **Note**
>
> Simply type `datas` in the terminal and press *enter* to see the full list of execution options for the module.

To assess the equilibrium, the properties acquired during the simulation can be analyzed with the following basic command structure:

> datas ⟨simbox.ext⟩

Properties such as pressure, temperature, density, and potential energy are evaluated graphically and statistically (descriptive statistics include the calculation of standard error, standard deviation and drift of properties; RMSD and RDF analysis are also available).

After a production step, simulation data extraction can be performed with trajectory truncation. Several options are available, including the ability to perform RMSD and RDF analyses of the

truncated trajectory. The simplest procedure is to run the command:

> datas ⟨simbox.ext⟩ -ext

It is possible to apply a set of filters to the extraction of configurations. For example, it is possible to extract configurations whose pressure is around the target pressure *P*, within a narrow window of ±*D*:

> datas ⟨simbox.ext⟩ -ext -ptr [*P*:*D*]

2.2.2. Execution Options

> datas ⟨file.ext⟩ [-dat [-opt]][-plt [-opt]] [-trj [-opt]]

```
-------------------------------------------------------------------------------
   auto : Execute an automated properties extraction from simulation files.
        » Description
          Properties extraction + statistical and graphical analysis
        » NOTE: this option is recommended in queue systems.
        + ------------------------------------------------------
   -dat : Generate a dat file only. [for GMX: -min|-opt|-nvt|-npt]
   -cpt : Generate a dat file in compact format.
   -dot : Convert dots to commas.
   -psa : Performs statistical analysis.
   -for : Force new property data extraction.
        + ------------------------------------------------------
   -rms : RMSD analysis for SLT molecular residue (full trajectory).
   -rdf : RDF analysis for center of masses of ref SLT and sel SVT residues.
        = [I:J] : RDF analysis for ref SLT/I and sel SVT/J labels.
   -pdf : PDF analysis for center of masses of ref SLT and sel SVT residues.
        = [I:J] : PDF analysis for ref SLT/I and sel SVT/J indexes.
   -sma : Calculate simple moving average at frequency 5 to the extracted RDF data
   -tim = Final simulation interval (in ps) to extract RDF data (def.: 100 ps).
   -cnv = Cumulative numerical value to be highlighted in the RDF/PDF analysis.
   -shw : Visualize the performed SMA analysis.
        + ------------------------------------------------------
   -plt : Generate graphs for params vs simulation time.
   -stp : Generate graphs for params vs simulation step.
        + ------------------------------------------------------
   -trj : Analize the trajectory file.
   -ext : Extract a truncated trajectory file.
   -ctr = [I:J] : Skip I equilibration steps and extract J production steps.
   -btr = [B:X] : Select frames from B trajectory blocks into X% final steps.
   -ptr = [P:D] : Select configurations aroud pressure P with tolerance ±D (in bar
   -prd : Select configuration production steps only and the corresponding propert
   -grp : Extrac short-ranged non-bonded potential energies.
   -nbx : Do not make corrections for “fragmented” molecules near the box edges.
-------------------------------------------------------------------------------
(*) Configurable options in automated mode. (●) ORCA available. (■) GROMACS availa
(†) RDF = Radial Distribution Function. (from TRJ.XTC file)
(‡) PDF = Pair Distribution Function.   (from TRY.XYZ file)
```

2.2.3. Quickstart Examples

- Example #1: Data analysis from ORCA simulation
- Example #2: Data analysis from GROMACS simulation

2.3. STATS

2.3.1. MD Statistics

The `STATS` module performs statistical analysis of simulated properties, through a unified interface for the *ORCA* and *GROMACS* programs. It can be used from the command:

> stats ⟨simbox.ext⟩ [⟨-options⟩]

where *ext* is one of the extensions recognized by *Solvate Suite* (out [*ORCA*], or log [*GROMACS*]), and “-options” a set of the configurations for the simulation execution.

> **Note**
>
> Simply type `stats` in the terminal and press *enter* to see the full list of execution options for the module.

2.3.2. Execution Options

> stats ⟨datafile.dat⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   -cpt : Read a generated dat file in compact format.
   -for : Force new property data extraction.
-------------------------------------------------------------------------------
```

2.3.3. Quickstart Examples

- Under construction…

2.4. PLOTS

2.4.1. MD Graphics

The `PLOTS` module performs graphical analysis of simulated properties, through a unified interface for the *ORCA* and *GROMACS* programs. It can be used from the command:

> plots ⟨simbox.ext⟩ [⟨-options⟩]

where *ext* is one of the extensions recognized by *Solvate Suite* (out [*ORCA*], or log [*GROMACS*]), and “-options” a set of the configurations for the simulation execution.

> **Note**
>
> Simply type `plots` in the terminal and press *enter* to see the full list of execution options for the module.

2.4.2. Execution Options

> plots ⟨inputs.ext⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
Formatted plots:
        + ------------------------------------------------------------
   -tim : Plot graphs for params vs simulation time.
   -stp : Plot graphs for params vs simulation step.
   -cnv = Cumulative value to be highlighted in RDF analysis.
        + ------------------------------------------------------------
General plots:
        + ------------------------------------------------------------
   -csv : Load an unformated general plot.
   -plt : Plot y versus x as lines (default).
   -sct : A scatter plot of y vs x with varying marker size/color.
   -bar : Make a bar plot.
   -stm : Create a stem plot.
        + ------------------------------------------------------------
   -sma : Simple moving average to the extracted RDF data.
        + ------------------------------------------------------------
Saving plots:
        + ------------------------------------------------------------
   -bak : Save generated graphs (without iterative interface).
-------------------------------------------------------------------------------
(*) ext = dat, rms, rdf, csv, nmr
```

2.4.3. Quickstart Examples

- Under construction…

2.5. TRAJS

2.5.1. MD Trajectories

The `TRAJS` module performs the extraction of simulated trajectory from *GROMACS* program. It can be used from the command:

> trajs ⟨simbox.gro⟩ [⟨-options⟩]

> **Note**
>
> Simply type `trajs` in the terminal and press *enter* to see the full list of execution options for the module.

2.5.2. Execution Options

> trajs ⟨trajectory.gro/xtc⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   auto : Execute an automated trajectory extraction from simulation files.
        » Description
          btrj = 1000 blocks with 10% extraction
        » NOTE: this option is recommended in queue systems.
        + #1: Individual Options - (no data analysis) ----------------------------
   -itr = I     : Extracts position structure #I from the trajectory.
   -ntr = N     : Extract N equaly spaced MD steps for truncated trajectory.
   -str = M     : Extract equaly spaced frames at every M steps from trajectory.
        + #2: Combinable Options - (with data analysis) --------------------------
   -ctr = [I:J] : Skip I equilibration steps and extract J production steps.
   -btr = [B:X] : Select frames from B trajectory blocks into X% final steps.
   -ptr = [P:D] : Select configurations around pressure P with tolerance ±D (in ba
        + #3: Output file formats - (GROMACS only) -------------------------------
   -xyz : Output in XYZ file format. [default]
   -pdb : Output in PDB file format.
   -gro : Output in GRO file format.
        + #4: Other Options ------------------------------------------------------
   -prd : Select configuration production steps only and the corresponding propert
   -grp : Extract short-ranged non-bonded potential energies.
   -nbx : Don´t make corrections for “fragmented” molecules near the box edges.
   -bak : Save intermediate PDB and XVG files for alternative convertions.
   -deb : Save intermediate XVG file and create a reconstructed ENT file.
   -for : Force new extraction and override previous PDB file.
-------------------------------------------------------------------------------
(*) Configurable options in automated mode. (●) ORCA available. (■) GROMACS availa
(†) Structure position, not configuration step.
```

2.5.3. Quickstart Examples

- Under construction…

3. **Microsolvation**

SETTING UP THE MICROSOLVATION

In the **third stage**, after extracting the trajectory, the truncated trajectory file *simbox.try.xyz* is obtained. The `MICRO` module can then select *N* number of lower-energy clusters, each containing *M* solvent molecules. The value of *M* can be determined from a previous RDF analysis (using the DATAS module), set based on the solute-solvent ratio (at a concentration specified by the user), or calculated automatically (based on a cutoff distance corresponding to the diameter of the sphere encompassing the solvent molecule). The extraction is performed using the following command structure:

> solvate ⟨simbox.trj.xyz⟩ -micro [⟨-options⟩]

After this extraction, the selected clusters are saved in a file *solute.trj.xyz*. This file can be used to analyze hydrogen bonds with the `BONDS` module. Alternatively, one can treat microsolvation at the level of the first solvation shell with the `GCALC` module, simultaneously treating all types of short-range interactions present in the system under investigation, regardless of the nature of the solvent. These modules are executed in two steps. In the first, all clusters present in the input file are characterized, aiming to select several *C* clusters with the lowest energy. In the second step, the *C* clusters are prepared for optimization and the Gibbs free energy extrapolation, considering one of the treatment options to remove any imaginary frequencies. In each of them, the procedure uses the following command structure:

> solvate ⟨solute.trj.xyz⟩ -bonds [⟨-options⟩]

or:

> solvate ⟨solute.trj.xyz⟩ -gcalc [⟨-options⟩]

In the optimization step, the input files are saved in a separate jobrunning folder. In this folder, the module `QUEUE` can be executed directly to perform the optimization and the Gibbs free energy extrapolation in a queueing system. This procedure sequentially executes the `ONION` module, which is the resource responsible for performing geometry optimization, frequency calculation, and obtaining the extrapolated Gibbs free energy.

> **Note**
>
> Several additional options for treating clusters are available. The complete list can be accessed in the help menu of the MICRO, BONDS/GCALC, and QUEUE modules (just type the program name on the command line, without additional options). Finally, all the processes can be automated through the MSRUN module. For examples, visit the MICRO, BONDS/GCALC, and QUEUE tutorials page.

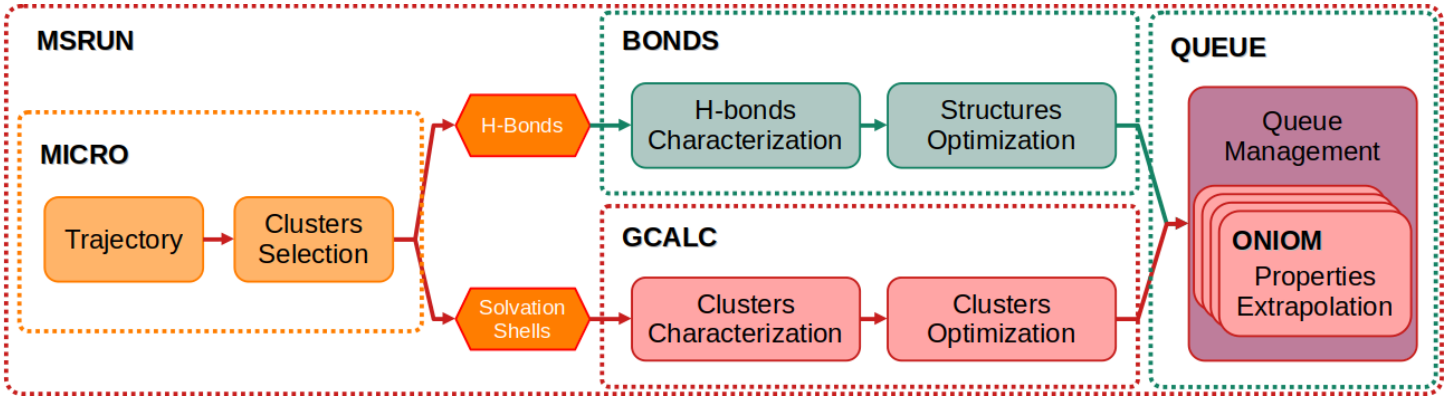
Once all the necessary settings for executing the microsolvation treatment are known, the MSRUN module makes it possible to perform the entire procedure from a single command:

```
solvate ⟨input.trj.xyz⟩ -msrun [⟨-options⟩]
```

where *input.trj.xyz* can be the file *simbox.trj.xyz* (from extracted trajectory file with TRAJS module, which corresponds to the complete procedure by default), or the file *solute.trj.xyz* (from extracted clusters file with MICRO module, in which the partial procedure is executed, without MICRO reexecution). The settings provided via *-options* make it possible to adjust the execution of the MICRO, BONDS/GCALC, QUEUE, and ONIOM modules independently.

> **Note**
>
> The list of options can be seen from the applications menu, which is accessible from the command line by typing the module name, without any other parameters.

STAGE STRUCTURE



3.1. MSRUN

3.1.1. Microsolvation Execution

The `MSRUN` module manages the entire execution of microsolvation treatment. The module can be run from the main program or directly from the command:

> msrun ⟨input.trj.xyz⟩ [⟨-options⟩]

where *input.trj.xyz* can be the file *simbox.trj.xyz* (from extracted trajectory file with TRAJS module, which corresponds to the complete procedure by default), or the file *solute.trj.xyz* (from extracted clusters file with MICRO module, in which the partial procedure is executed, without MICRO reexecution). The settings provided via *-options* make it possible to adjust the execution of the MICRO, BONDS/GCALC, QUEUE, and ONIOM modules independently.

> **Note**
>
> Simply type `msrun` in the terminal and press *enter* to see the full list of execution options for the module.

3.1.2. Execution Options

> msrun ⟨simbox.trj.xyz⟩ [⟨-options⟩][(a)]

or

> msrun ⟨solute.trj.xyz⟩ [⟨-options⟩][(b)]

```
-------------------------------------------------------------------------------
  -auto : Execute an automated microsolvation optimization.
        + ---------------------------------------------------------
   -mic = MICRO module configuration. [ Ex.: -mc "-str 1000 -bco 1.0" ]
   -bnd = BONDS module configuration. [ Ex.: -bd "-ini -opt -max 100 -fco" ]
   -gcc = GCALC module configuration. [ Ex.: -gc "-ini -opt -max 100 -fca" ]
   -que = QUEUE module configuration. [ Ex.: -qc "-key “#ONIOM(M062X/6-311+G(d,p):
        + ---------------------------------------------------------
   -prc = Number of parallel CPU processes to run.                  [Default: 8]
   -mem = Amount of memmory to be used (in GB).                     [Default: 16]
-------------------------------------------------------------------------------
Total CPUs: 12 (max. MPI-procs.) :: System Memory: 5 GB
-------------------------------------------------------------------------------
(a) From extracted tractory file (with TRAJS). [Complete procedure by default]
(b) From extracted clusters file (with MICRO). [Partial procedure, without MICRO r
(*) Each module can be run independently.
(†) The “-key” option can be passed separately.
```

3.1.3. Quickstart Examples

- Example #1: Treating clusters from a simulation with ORCA
- Example #2: Treating clusters from a simulation with GROMACS

3.2. MICRO

3.2.1. Clusters Extraction

The `MICRO` module manages the extraction of microsolvation clusters. It is possible to select *N* low-energy clusters, each containing *M* solvent molecules. The value of *M* can be determined from a previous RDF analysis (using the DATAS module), defined based on the solute-solvent ratio (at a user-specified concentration), or calculated automatically (based on a cutoff distance corresponding to the diameter of the sphere surrounding the solvent molecule). The extraction is performed using the following command:

> micro ⟨simbox.trj.xyz⟩ -str *N* -nsv *M*

The extraction based on molal concentration *C* is performed using the command:

> micro ⟨simbox.trj.xyz⟩ -str *N* -bco *C*

The extraction based on the number *S* of solvation shells (estimated from the solvent radius) is performed using the command:

> micro ⟨simbox.trj.xyz⟩ -str *N* -sol *S*

Note

Simply type `micro` in the terminal and press *enter* to see the full list of execution options for the module.

3.2.2. Execution Options

> micro ⟨trajectory.trj.xyz⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   auto : Execute an automated microsolvation clusterization from simulation files
        » Description
          bCon = 1 mol·kg⁻¹ (for setting the number of solvent molecules)
          Nstr = 1% of trajectory structures (1000 min, 10000 max)
          cTRJ = All production steps
        + ------------------------------------------------------
   -str = Number of clustered structures selected trajectory.        [Default: 100
   -nsv = Number of solvent molecules in the clustering process.     [Default: 0]
   -sol = Number of solvent layers in the clustering process.        [Default: 1]
   -bco = Molal conc. of the microsolvation cluster (in mol·kg⁻¹).   [Default: 0.0
   -cfg = Nº of MD equilibration steps [:Nº of MD production steps]. [Default: 0:0
   -chr = Value of microsolvated solute charge.                      [Default: 0]
   -mlt = Value of microsolvated solute multiplicity.                [Default: 1]
   -prs = Value of simulation pressure (in bar).                     [Default: 1.0
   -tmp = Value of simulation temperature (in K).                    [Default: 298
   -svt = Solvent to be used in ALPB solvation model.                [Default: Aut
   -rtn : Retain the original orientation of the clusters.           [Default: No]
   -psv : Preserve the original trajectory file.                     [Default: No]
   -for : Execute in forced mode.                                    [Default: No]
        + ------------------------------------------------------
   -prc = Number of parallel CPU processes to run.                   [Default: 8]
   -mem = Amount of memmory to be used (in GB).                      [Default: 16]
-------------------------------------------------------------------------------
(*) Configurable options in automated mode.
```

3.2.3. Quickstart Examples

- Example #1: Extracting clusters from a simulation with ORCA
- Example #2: Extracting clusters from a simulation with GROMACS

3.3. BONDS

3.3.1. H-Bonds Treatment

The `BONDS` module manages the H-bond clusters treatment. After extraction with MICRO module, the selected clusters are saved in a file *solute.trj.xyz*. This module are executed in two steps. In the first, all clusters present in the input file are characterized, aiming to select several *C* clusters with the lowest energy, with the command:

> bonds ⟨solute.trj.xyz⟩ -ini -max *C*

In the second step, the *C* clusters are prepared for optimization and the Gibbs free energy extrapolation, considering one of the treatment options to remove any imaginary frequencies:

> bonds ⟨solute.trj.xyz⟩ -opt -max *C*

In this step, the input files are saved in a separate jobrunning folder.

> **Note**
>
> Simply type `bonds` in the terminal and press *enter* to see the full list of execution options for the module.

3.3.2. Execution Options

> bonds ⟨solute.trj.xyz⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   auto : Execute an automated microsolvation optimization.
        » NOTE: this option is recommended in queue systems.
        » HINT: -auto    [starts processing 100 clusters]
              : -auto N  [starts processing N clusters]
              : -auto +M [add M clusters to the previous analysis]
-------------------------------------------------------------------------------
STEPS CONTROL
-------------------------------------------------------------------------------
        ● Stage #1: Clusters Characterization
   -ini : Execute clusters characterization and selection.          [Default: Aut
          Characterization for structures selection.
          Procedure performed at xTB-GFF level.

        ● Stage #2: Clusters Treatment
   -opt : Optimization and free energy calculation. (FCA|FCO)       [Default: FCO
          Calculation after structures selection.
          Procedure performed at xTB-GFN2 level combined with DFT.
        » FCA: [F]ull [C]luster [A]ccommodated geometry (with SPH).¹ «xTB level»
        » FCO: [F]ull [C]luster [O]ptimized geometry (with CHM).²    «xTB level»
-------------------------------------------------------------------------------
ADITIONAL OPTIONS
-------------------------------------------------------------------------------
        ● Selection Options
   -max = Maximum number of structures into energy window.          [Default: 100
   -add = Add M clusters to GFE procedure from INI step.            [Default: Not

        ● System Options
   -chr = Value of microsolvated solute charge.                     [Default: Aut
   -mlt = Value of microsolvated solute multiplicity.               [Default: Aut
   -prs = Pressure for thermochemistry calculation (in atm).        [Default: Aut
   -tmp = Temperature for thermochemistry calculation (in K).       [Default: Aut

        ● Solvation Options
   -svt = Solvent type (for ALPB/CPCM solvation method).            [Default: Aut
   -nis = No implicit solvation.                                    [Default: Nor

        ● Configuration Options
   -prc = Number of parallel CPU processes to run.                  [Default: 8]
   -mem = Amount of memmory to be used (in GB).                     [Default: 16]

        ● Additional Options
   -key = “List of keywords” (for ONIOM program).                   [Default: Nor
   -fcc = Final concentration after compression.                    [Default: 1.0
   -hbd : Include H-bonds at extrapolation (for ORCA final step).   [Default: No]
   -nmr : Config. for NMR calculation (for ORCA final step).        [Default: No]
   -for : Force reexecution stages.                                 [Default: No]
   -fil : Filter high energy clusters (above 100.0 kcal·mol⁻¹).     [Default: No]
        + ------------------------------------------------------
   -dat : Extract data for selected stage.                          [Default: No]
   -plt : Plot extracted data.                                      [Default: No]
   -bak : Save generated graphs (without iterative interface).      [Default: No]
        + ------------------------------------------------------
```

```
    -exc = List of structural clusters to exclude. (inp) [i,j,k,...]  [Default: Nor
    -clr = List of optimized clusters to clean.    (out) [i,j,k,...]  [Default: Nor
         + -------------------------------------------------------
    -del = Delete specified previous optimization.                    [Default: No]
    -ext = Extract specified previous calculation.                    [Default: No]
    -ren = Rename specified previous description. "DSC" / [OLD:NEW]   [Default: No]
    -chk : Check all previous calculations. (= spec. to extract)      [Default: No]
    -log : Summary of performed calculations.                         [Default: No]
         + -------------------------------------------------------
    -out = Force reading specified ONIOM output file.                 [Default: No]
         : Force reading default ONIOM output file.                   [Default: No]
-------------------------------------------------------------------------------
Total CPUs: 12 (max. MPI-procs.) :: System Memory: 5 GB
-------------------------------------------------------------------------------
(*) Configurable options in automated mode.
(▪) Try "-sN" to view stage N only.
(†) The "ith" cluster identified from trj.xyz input file.
(‡) The "ith" cluster identified through graphical analysis.
(1) SPH = Single Point Hessian.
(2) CHM = Conventional Hessiam Method.
```

3.3.3. Quickstart Examples

- Example #1: Treating clusters from a simulation with ORCA

3.4. GCALC

3.4.1. Clusters Treatment

The `GCALC` module manages the H-bond clusters treatment. After extraction with MICRO module, the selected clusters are saved in a file *solute.trj.xyz*. This module are executed in two steps. In the first, all clusters present in the input file are characterized, aiming to select several *C* clusters with the lowest energy, with the command:

> gcalc ⟨solute.trj.xyz⟩ -ini -max *C*

In the second step, the *C* clusters are prepared for optimization and the Gibbs free energy extrapolation, considering one of the treatment options to remove any imaginary frequencies:

> gcalc ⟨solute.trj.xyz⟩ -opt -max *C*

In this step, the input files are saved in a separate jobrunning folder.

> **Note**
>
> Simply type `gcalc` in the terminal and press *enter* to see the full list of execution options for the module.

3.4.2. Execution Options

> gcalc ⟨solute.trj.xyz⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   auto : Execute an automated microsolvation optimization.
        » NOTE: this option is recommended in queue systems.
        » HINT: -auto    [starts processing 100 clusters]
              : -auto N  [starts processing N clusters]
              : -auto +M [add M clusters to the previous analysis]
-------------------------------------------------------------------------------
STEPS CONTROL
-------------------------------------------------------------------------------
        ● Stage #1: Clusters Characterization
   -ini : Execute clusters characterization and selection.          [Default: Aut
          Characterization for structures selection.
          Procedure performed at xTB-GFF level.

        ● Stage #2: Clusters Treatment
   -opt : Optimization and free energy calculation. (FSA|FCA)       [Default: FCA
          Calculation after structures selection.
          Procedure performed at xTB-GFN2 level combined with DFT.
        » FSA: [F]ull [S]olutes [A]ccommodated geometry (with SPH).[1] «xTB level»
        » FCA: [F]ull [C]luster [A]ccommodated geometry (with SPH).[1] «xTB level»
-------------------------------------------------------------------------------
ADITIONAL OPTIONS
-------------------------------------------------------------------------------
        ● Selection Options
   -max = Maximum number of structures into energy window.          [Default: 100
   -add = Add M clusters to GFE procedure from INI step.            [Default: Not

        ● System Options
   -chr = Value of microsolvated solute charge.                     [Default: Aut
   -mlt = Value of microsolvated solute multiplicity.               [Default: Aut
   -prs = Pressure for thermochemistry calculation (in atm).        [Default: Aut
   -tmp = Temperature for thermochemistry calculation (in K).       [Default: Aut

        ● Solvation Options
   -svt = Solvent type (for ALPB/CPCM solvation method).            [Default: Aut
   -nis = No implicit solvation.                                    [Default: Nor

        ● Configuration Options
   -prc = Number of parallel CPU processes to run.                  [Default: 8]
   -mem = Amount of memmory to be used (in GB).                     [Default: 16]

        ● Additional Options
   -key = “List of keywords” (for ONIOM program).                   [Default: Nor
   -fcc = Final concentration after compression.                    [Default: 1.0
   -hbd : Include H-bonds at extrapolation (for ORCA final step).   [Default: No]
   -nmr : Config. for NMR calculation (for ORCA final step).        [Default: No]
   -for : Force reexecution stages.                                 [Default: No]
   -fil : Filter high energy clusters (above 100.0 kcal·mol⁻¹).     [Default: No]
        + ------------------------------------------------------
   -dat : Extract data for selected stage.                          [Default: No]
   -plt : Plot extracted data.                                      [Default: No]
   -bak : Save generated graphs (without iterative interface).      [Default: No]
        + ------------------------------------------------------
```

```
    -exc = List of structural clusters to exclude. (inp) [i,j,k,...]  [Default: Nor
    -clr = List of optimized clusters to clean.    (out) [i,j,k,...]  [Default: Nor
         + -------------------------------------------------------
    -del = Delete specified previous optimization.                    [Default: No]
    -ext = Extract specified previous calculation.                    [Default: No]
    -ren = Rename specified previous description. “DSC” / [OLD:NEW]   [Default: No]
    -chk : Check all previous calculations. (= spec. to extract)      [Default: No]
    -log : Summary of performed calculations.                         [Default: No]
         + -------------------------------------------------------
    -out = Force reading specified ONIOM output file.                 [Default: No]
         : Force reading default ONIOM output file.                   [Default: No]
  ------------------------------------------------------------------------------
  Total CPUs: 12 (max. MPI-procs.) :: System Memory: 5 GB
  ------------------------------------------------------------------------------
  (*) Configurable options in automated mode.
  (▪) Try “-sN” to view stage N only.
  (†) The “ith” cluster identified from trj.xyz input file.
  (‡) The “ith” cluster identified through graphical analysis.
  (1) SPH = Single Point Hessian.
  (2) CHM = Conventional Hessiam Method.
```

3.4.3. Quickstart Examples

- Example #1: Treating clusters from a simulation with GROMACS

3.5. QUEUE

3.5.1. Clusters Submission

The `QUEUE` module manages the entire process in the system queue (if available). It can be used from the command:

> queue [(-options)]

in which all the settings are provided via *-options*, that makes it possible to configure the ONIOM calculation.

> **Note**
>
> Simply type `queue` in the terminal and press *enter* to see the full list of execution options for the module.

3.5.2. Execution Options

> queue [(-options)]

```
-------------------------------------------------------------------------------
   -key = Complete list of keywords for ONIOM calculation configuration.
   -chr = Charge value of the microsolvation cluster.
   -mlt = Multiplicity value of the microsolvation cluster.
   -des = Optional output configuration description.
   -fcc = Final concentration after compression. [Default: 1.0 mol·kg⁻¹]
   -inf : Show current ONIOM calculation configuration.
   -con : Force concatenation of results over warnings.
   -clr : Clean up run folder after archiving results.
   -run : Perform the calculation procedure.
   -srm : Execute calculation in a serial mode.
   -for : Execute calculation in forced mode.
        + ------------------------------------------------------
   -bnd : Execution for BONDS module.
   -gcc : Execution for GCALC module. (default)
        + ------------------------------------------------------
   -prc = Number of parallel CPU processes to run.                [Default: 8]
   -mem = Amount of memmory to be used (in GB).                   [Default: 16]
-------------------------------------------------------------------------------
```

3.5.3. Quickstart Examples

- Under construction…

3.6. ONIOM

3.6.1. Clusters Calculation

The `ONIOM` module manages the optimization and ONIOM calculation for Gibbs free energy extrapolation. It can be done from the command:

> oniom ⟨input.gjf⟩ [⟨-options⟩]

where *input.gjf* is a file with *GaussView* format for microssolvation calculation, and “-options” a set of the configurations for the calculation execution.

> **Note**
>
> Simply type `oniom` in the terminal and press *enter* to see the full list of execution options for the module.

3.6.2. Execution Options

> oniom ⟨input.gjf⟩ [⟨-options⟩]

```
   ------------------------------------------------------------------------------
   ⟨input.gjf⟩ = File with GaussView format for Solvate/Microssolvation calculation
   ------------------------------------------------------------------------------
      -opt : Execute geometry optimization in forced mode.              [Default: No]
      -ext : Execute ONIOM extrapolation in forced mode.                [Default: No]
           + ------------------------------------------------------------
      -srm : Execute calculation in a serial mode.                      [Default: No]
      -for : Execute optimization and extrapolation in forced mode.     [Default: No]
      -deb : No extrapolation execution: only tests.                    [Default: No]
           + ------------------------------------------------------------
      -prc = Number of parallel CPU processes to run.                   [Default: 8]
      -mem = Amount of memmory to be used (in GB).                      [Default: 16]
   ------------------------------------------------------------------------------
```

3.6.3. Quickstart Examples

- Under construction…

4. **Management**

FILE MANAGEMENT

In the **fourth stage** (and final), after all the steps of the computational experiment have been executed, it is possible to manage the significant volume of files from a single command. The `FILES` module can perform file management, archiving or unarchiving the files handled throughout the simulation and calculation of the electronic structure. Management is performed using the following command structure:

> solvate ⟨solute.trj.xyz⟩ -files [⟨-options⟩]

> **Note**
>
> The list of options can be seen from the applications menu, which is accessible from the command line by typing the module name, without any other parameters.

4.1. FILES

4.1.1. File Management

The `FILES` module manages the collection of files of the computational experiment from a single command, archiving or unarchiving the files handled throughout the simulation and calculation of the electronic structure. Management is performed using the following command structure:

> files ⟨solute.trj.xyz⟩ [⟨-options⟩]

> **Note**
>
> Simply type `files` in the terminal and press *enter* to see the full list of execution options for the module.

4.1.2. Execution Options

> files ⟨solute+solvent_molN.ext⟩ [⟨-options⟩]

```
-------------------------------------------------------------------------------
   auto : Execute an automated file management.
        » Description
          Compress if intermediate files are present
          Expand if intermediate files are absent
        + ------------------------------------------------------
   -cpt : Compact the intermediate files.                        [Default: No]
   -ext : Extract the intermediate files.                        [Default: No]
-------------------------------------------------------------------------------
```

4.1.3. Quickstart Examples

- Under construction…

5. **Additional Modules**

5.1. **SOLVS**

- List of available solvents and ions for packaging

```
-------------------------------------------------
solvs [<solvent>] [<-option>]

OPTIONS/DEFAULTS
   -field T   [Force field parameters]
               ► ALL = All force fields
               ● OPL = OPLS-AA/L all-atom force field
               ○ LPG = OPLS-AA/L (from LigParGen)
               ○ QFC = OPLS-AA/L (from Q-Force)
               ● GRO = GROMOS96 54a7 force field
               ○ ATB = GROMOS96 54a7 (from ATB)
               ● CHM = CHARMM36 all-atom force field
               ● AMB = AMBER-GS force field
-------------------------------------------------
```

5.2. **MDPAR**

- List of available configuration files for MD simulations

```
-------------------------------------------------
mdpar [⟨-step N⟩]

OPTIONS/DEFAULTS
     -step N   [Step for extracting MDP parameters]
               ► 0/ALL = List all MDP files
               ○ 1/OPT = Box Optimization
               ○ 2/MIN = Energy Minimization
               ● 3/NVT = NVT Equilibration
               ● 4/NPT = NPT Equilibration
               ● 5/PRD = NPT Production
-------------------------------------------------
```

5.3. **CHECK**

- Monitoring of procedures progress

```
-----------------------------------------------------------------------
check ⟨input.ext⟩ [⟨compt.ext⟩] [⟨-options⟩]

PARAMETERS
     input.ext [Input file]
     compt.ext [Compacted file]
               [Extensions: COM/INP/XYZ/LOG/OUT/XTB/TMP]

OPTIONS
    -ext       [Extract compacted internal file]
               (ex.: check inpfile.zip cptfile.cpt -ext)
    -del       [Deletes compacted internal file][zip only]
               (ex.: check inpfile.zip cptfile.cpt -del)
    -ren       [Renames compacted internal file][zip only]
               (ex.: check inpfile.zip oldfile.cpt -ren newfile.cpt)
-----------------------------------------------------------------------
```

5.4. **COMPT**

- Compression of output files

```
---------------------------------------
compt ⟨output.log/out/xtb⟩ [-chk]

PARAMETERS
    output.ext [Output LOG/OUT/XTB file]
---------------------------------------
```

5.5. **CONFS**

- Conformational analysis automator

```
-------------------------------------------------------------------
confs ⟨input.ext⟩ [⟨-options⟩]

PARAMETERS
      input.ext [Output COM/LOG/INP/OUT/XYZ/XTB/GRO/PDB file]

OPTIONS
-inp       [Output file in ORCA format (default)]               [-in]
-gro       [Output file in GROMACS format]                      [-gr]
-pdb       [Output file in PDB format (for LigParGen)]          [-pd]
-xtb       [Use GFN2-xTB in conformational analysis (slower)]   [-xt]
-sep       [Split the output file into its conformers]          [-sp]
-key       [List of keywords for final optimization]            [-kw]
-for       [Force new extraction and override previous XTB file] [-fm]
-------------------------------------------------------------------
```

5.6. **GCOPT**

- gCalc SOC optmization procedure

```
-----------------------------------------------------------------------------
gcopt ⟨slt.xyz/slt.slt/solute.gjf⟩ [⟨-options⟩] &

PARAMETERS
    slt.xyz    [Input XYZ/SLT/GJF file]

OPTIONS/DEFAULTS
    -key K     [List of keywords]                     [Default: "! M062X 6-31+G
    -svt A     [Solvent type for CPCM solvation]      [Default: None]
    -prs p     [Pressure for thermochemistry (in bar)] [Default: 1.0]
    -tmp T     [Temperature for thermochemistry (in K)][Default: 298.15]
    -cut N     [Cut value for structural RMSD]        [Default: 0.075]
    -opt       [Perform a single geometry optimization][Default: Yes]
    -fgo       [Perform a full geometry optimization]  [Default: None]

ADDITIONAL
    -prc P     [Number of processors to be used]      [Default: 4]
    -mem M     [Amount of memory to use in GB]        [Default: 8]
-----------------------------------------------------------------------------
```

5.7. **IMAGS**

- Imaginary mode distortion

```
---------------------------------------
imags <output.log/out/xtb> [-img fator]

PARAMETERS
    output.ext [Output LOG/OUT/XTB file]
---------------------------------------
```

5.8. **MOLAP**

- Structural alignment for RMSD minimization

```
--------------------------------------------------------------------------------
molap ⟨reference.ext⟩ ⟨structure.ext⟩ [⟨trajectory.trj.xyz⟩] [⟨-options⟩]

PARAMETERS
     files.ext [Input  COM/LOG/INP/OUT/XYZ/XTB/GRO/PDB file]

OPTIONS
    -bak       [Save modified “structure” file with minimized RMSD (at system or
    -ref       [Save modified “structure” file with minimized RMSD (at “referenc
    -str       [Save modified “structure” file with minimized RMSD (at “structur
    -slt       [Reorientation relative to the central solute fragment]
--------------------------------------------------------------------------------
```

5.9. **RMSDE**

- Calculation of structural RMSD and solute energy

```
--------------------------------------------------------------------------------
rmsde ⟨reference.ext⟩ ⟨structure.ext⟩ [⟨preoptimization.ext⟩] [⟨trajectory.trj.x

PARAMETERS
     input.ext [Output COM/LOG/INP/OUT/XYZ/XTB/GRO/PDB file]

OPTIONS
    -min       [Maximum structural superposition for RMSD minimization]
    -bak       [Save RMSD information in the input “structure” file]
    -for       [Force new calculation and override previous output]
--------------------------------------------------------------------------------
```

5.10. **SCALC**

- Chemical shifts calculator

```
------------------------------------------------------------------------
scalc ⟨input.trj.xyz⟩ [⟨-options⟩]

PARAMETERS
 input.trj.xyz [Output TRJ file]

OPTIONS
    -ato A     [Atomic symbol of the species to be analyzed] [-at]
    -ref N     [Reference value for chemical shifts (in ppm)] [-rf]
    -eqv [L]   [List of equivalent atoms. (note: -eqv=[A1=A2:B1=B2=B3...])] [-eq
               +-----------------------------------------------------------+
    -out F     [Force reading specified ONIOM output file]                  [-of
    -out       [Force reading default ONIOM output file]                    [-of
               +-----------------------------------------------------------+
    -hbd       [Use BONDS data]                                             [-hb
    -gcc       [Use GCALC data]                                             [-gc
    -plt       [Plot the NMR spectrum]                                      [-pt
------------------------------------------------------------------------
* Note : Requires “gcc” file, and optionally “pcm” and “exp” files.
```

5.11. **SOLVX**

- Solvent identifier in clusters

```
-----------------------------------
solvx ⟨file.trj.xyz⟩

PARAMETERS
  file.trj.xyz [Output TJR.XYZ file]
-----------------------------------
```

5.12. **STAND**

- Modify input to standard orientation

```
----------------------------------------
stand ⟨input.ext⟩ &

PARAMETERS
     input.ext [Input GJF/COM/INP/XYZ/GRO]
----------------------------------------
```

5.13. **STATX**

- Full statistical analysis directly from GMX program

```
--------------------------------
statx ⟨output.log⟩

PARAMETERS
    output.log  [Output LOG file]
--------------------------------
```

5.14. **SUBST**

- Modification of texts in editable files

```
--------------------------------
subst ⟨input.ext⟩ "text1" "text2"

PARAMETERS
    input.ext [Extensions: all]
--------------------------------
```

5.15. **SUMMY**

- Extraction of properties from output files

```
------------------------------------------------------------------------------
summy ⟨output.log/out/xtb⟩

PARAMETERS
    output.ext  [Output LOG/OUT/XTB file]

OPTIONS/DEFAULTS
       -prs N   [Pressure configuration for thermodinamic calculation]
       -tmp N   [Temperature configuration for thermodinamic calculation]
       -fcc N   [Final concentration after compression (in mol·L⁻¹)]
       -frq N   [Scale factor for frequencies (for zero-point energy correction)
       -qha <N> [Quasi-Harmonic Approximation (with optional cutoff frequency)]
------------------------------------------------------------------------------
```

5.16. **SUPPY**

- Extraction of supplementary data from output files

```
------------------------------------------------------------------------------
suppy ⟨output.log/out/xtb⟩

PARAMETERS
    output.ext  [Output LOG/OUT/XTB file]

OPTIONS/DEFAULTS
       -prs N   [Pressure configuration for thermodinamic calculation]
       -tmp N   [Temperature configuration for thermodinamic calculation]
       -fcc N   [Final concentration after compression (in mol·L⁻¹)]
       -frq N   [Scale factor for frequencies (for zero-point energy correction)
       -qha <N> [Quasi-Harmonic Approximation (with optional cutoff frequency)]
------------------------------------------------------------------------------
```

5.17. **SVIEW**

- Visualization of input/output files

```
--------------------------------------------------------------------------------
sview ⟨input.ext⟩ [⟨-options⟩] &

PARAMETERS
    input.ext  [Input file](*)

GENERAL OPTIONS
    -try I     [Generate a try file with N truncated steps (.trj file)]
    -vmd       [Force file open with VMD program]
    -cpt       [Perform compression of input/output files for GVW program]

JMOL OPTIONS
    -noc       [Run without input convertion]

VMD OPTIONS
    -cpk       [Show solute with VDW representation]
    -vdw       [Show solute with VDW representation]
    -lico      [Activate Licorice for solvent shell]
    -dyna      [Activate DynamicBonds for solvent shell]
    -surf      [Activate QuickSurf for solvent shell]
    -text      [Execute in a text mode]
--------------------------------------------------------------------------------
(*) Conversions: COM/GJF/LOG/INP/OUT/XYZ/XTB/PDB/GRO/TRJ
```

5.18. **PDBIO**

- PDB processor for Biomolecular systems

```
-------------------------------------------------------------
pdbio ⟨input.pdb⟩ [⟨-options⟩]

PARAMETERS
     input.pdb [Input PDB file]
output.gro/itp [Output GRO/ITP file]

OPTIONS/DEFAULTS
    -ffp T     [Force field parameters]  [Default: OPLS]
               * GRO = GROMOS96 54a7 force field
               * OPL = OPLS-AA/L all-atom force field

ADDITIONAL
    -pdb       [Archives PDB input]
    -gro       [Generate GRO geometry file]
    -itp       [Produces ITP forcefield file]
    -ion       [Don't do ITP files of ionic species]
    -lap       [Override PDB input file]
    -cls       [Wipe all PDB comment lines]
    -bak       [Create a backup of auxiliary files]
-------------------------------------------------------------
```

5.19. **PDB2X**

- PDB cleaner module

```
--------------------------------
pdb2x ⟨input.pdb⟩ [⟨-options⟩]

PARAMETERS
     input.pdb [Input PDB file]
    output.ent [Output ENT file]
--------------------------------
```

5.20. **IONIC**

- PDB ions forcefield constructor

```
-----------------------------------
ionic ⟨input.pdb⟩ [⟨-options⟩]

PARAMETERS
     input.pdb [Input PDB file]
output.gro/itp [Output GRO/ITP file]
-----------------------------------
```

5.21. **GROUT**

- GRO files processor module

```
-------------------------------------------------------------
grout ⟨input.gro/itp⟩ [⟨-options⟩]

PARAMETERS
 input.gro/itp [Input GRO/ITP file]
output.gro/itp [Output GRO/ITP file]

OPTIONS
    -gro       [Archives GRO input/output]
    -itp       [Produces a concatenated ITP forcefield file]
    -pdb       [Converts a PDB file to GRO input geometry]
-------------------------------------------------------------
```

5.22. **GRO2X**

- GRO cleaner module

```
-----------------------------------
gro2x ⟨input.gro/itp⟩ [⟨-options⟩]

PARAMETERS
 input.gro/itp [Input GRO/ITP file]
output.gro/itp [Output GRO/ITP file]
-----------------------------------
```

5.23. **GROTC**

- GRO total charge calculator

```
------------------------------
grotc ⟨input.gro⟩

PARAMETERS
 input.gro     [Input GRO file]
------------------------------
```

5.24. **OUTIN**

- Manipulation of input/output files

> outin ⟨file_inp.ext⟩ ⟨file_out.ext⟩ [⟨-options⟩]

```
--------------------------------------------------------------------------------
FILE FORMATS
--------------------------------------------------------------------------------
.ext: com : Gaussian    (read/write)[gjf]
      inp : ORCA        (read/write)
      xyz : xTB         (read/write)
      gro : GROMACS     (read/write)
      pdb : PDB         (read/write)
      log : Gaussian    (read)
      out : ORCA        (read)
      xtb : xTB         (read)
      xtc : GROMACS     (read)
      trj : XYZ/GRO/PDB (read)
          + ------------------------------------------------------
      cpt : Data Comp   (write)[Compress calculated data]
      dat : Calc Data   (write)[Extract calculated data (option: nogeom)]
      sup : Supl Data   (write)[Extract supplementary data]
      sep : Sepa Data   (write)[Separate input/output data]
      job : Jobs Exec   (write)[Separate input/output jobs]
      frq : Calc Frqs   (write)[Extract vibrational frequency data]
      img = Imag Dist   (write)[Imaginary mode distortion (factor of distortion)]
          + ------------------------------------------------------
      slt : SLTs Extn   (write)[Solute  extraction from trajectory frames]
      svt : SVTs Extn   (write)[Solvent extraction from trajectory frames]
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
CONVERTION OPTIONS
--------------------------------------------------------------------------------
     -key = “List of keywords” (for Gaussian or ORCA).
     -add = “List of keywords/additional” (for Gaussian).
     -svt = Include solvent type for implicit (with ALPB or PCM solvation method).
     -nmr : Include configuration for NMR calculation (for ORCA).
          + ------------------------------------------------------
     -itr = Extract structure nº N of a trajectory.                          [
     -ntr = Number N of equaly spaced MD steps for truncated trajectory.     [
     -str = Extract equaly spaced frames at every M steps from trajectory.   [
     -ctr = Skipe I equilibration and extract J production MD steps (auto mode). [
     -btr = Select frames from I trajectory sections into J% final steps.    [
     -ptr = Select frames aroud pressure P with tolerance ±D (in bar).       [
     -nsm = Number of solvent molecules in the clustering process.           [
     -bmc = Molal conc. of the microsolvation cluster (in mol·kg⁻¹).         [
     -grp : Extract short-ranged non-bonded potential energies.
     -nbx : Don´t make corrections for “fragmented” molecules near the box edges.
          + ------------------------------------------------------
     -prs = Include pressure configuration for thermodinamic calculation.    [
     -tmp = Include temperature configuration for thermodinamic calculation. [
     -fcc = Final concentration after compression (in mol·L⁻¹).              [
     -frq = Scale factor for frequencies (for zero-point energy correction). [
     -qha = Quasi-Harmonic Approximation (with optional cutoff frequency).   [
--------------------------------------------------------------------------------
(*) I,J = Integers; N,P,D = Real. (○) xTB available. (●) ORCA available. (■) GROMA
```

5.25. **SUBMIT**

- Run the program in the system queue

```
--------------------------------------------------
submit ⟨program⟩ ⟨input.ext⟩ ⟨-options⟩ ⟨-ntasks I⟩

PARAMETERS
     program   [gsn/orca/xtb/gmx program]
     input.ext [Input GJF/COM/INP/XYZ/GRO file]

OPTIONS/DEFAULTS
    -options   [Program specific options]
    -queue  T  [Name of available queue]
    -ntasks I  [Number of processors]    [-prc]
    -qmemor I  [Amount of memory (in GB)][-mem]
    -preserve  [Preserve the submission job file]
    -debug     [Check generated job before submit]

AVAILABLE SYSTEM BATCH
    »Local Machine

WARNING
     This is an open-source script that may require
     adjustments depending on the available queue
     system
--------------------------------------------------
```

5.26. **INFOS**

- Displaying system information

5.27. **CLEAN**

- Cleaner for temporary files

5.28. **DEPLY**

- Full deployment dependencies check

5.29. **runGSN**

- *Gaussian* program interface

5.30. **runGVW**

- *GaussView* program interface

5.31. **runORCA**

- *ORCA* program interface

5.32. **runXTB**

- *xTB* program interface

5.33. **runCREST**

- *CREST* program interface

5.34. **runGMX**

- *GROMACS* program interface

5.35. **runBABEL**

- *oBabel* program interface

5.36. **runJMOL**

- *JMol* program interface

5.37. **runVMD**

- VMD program interface

5.38. **runPACKMOL**

- *PackMol* program interface

5.39. **runQFORCE**

- *Q-Force* program interface