MORITZ BENSBERG, CHRISTOPH BRUNKEN, KATJA-SOPHIA CSIZI, MIGUEL STEINER, THOMAS WEYMUTH, AND MARKUS REIHER

USER MANUAL

SCINE SWOOSE 2.1.0

ETH ZÜRICH

Contents

Introduction 5

Installation 6

Using the Standalone Binary 11

Using the Python Library 13

The Parametrization of SFAM 15

Calculations With SFAM 24

Calculations With Other Molecular Mechanics Models 26

The Hybrid QM/MM Model 28

Automated QM Region Selection 30

Structure Preparations 34

Structure Optimizations 37

Molecular Dynamics Simulations 39

Example Calculations 41

Extensions Planned in the Future 45

References 46

Introduction

SCINE Swoose provides functionalities for treating large molecular systems in theoretical chemistry with self-parametrizing system-focused atomistic models (SFAMs).[1] This includes the full automated parametrization of such models for arbitrary systems. With the model, it is then possible to perform single point calculations, structure optimizations and simulate molecular dynamics trajectories. Furthermore, these models can be extended towards QM/MM-like hybrid models in order to, for example, describe chemical reactions at the active site of enzymes.

[1] Brunken, C.; Reiher, M. Self-Parametrizing System-Focused Atomistic Models, *J. Chem. Theory Comput.* **2020,** *16,* 1646–1665

In this manual, we describe the installation of the software, give a hands-on introduction to the program, and all of the functions and options.[2] At the end of this manual, we provide a prospect on features in future releases as well as references.

[2] Throughout this manual, the most import information is displayed in the main text, whereas useful additional information is given as a side note like this one.

For information on how to cite Swoose, check out the *References* section on https://scine.ethz.ch/download/swoose.

Installation

Conan package

Conan[3] is a Python-based package distribution solution. It integrates well with C++ code built with CMake, greatly simplifying the installation of libraries with multiple dependencies. Conan is installed with `pip`:

[3] `conan.io`

```
python3 -m pip install conan
```

Swoose packages are distributed from our research group's remote, which can be added with:

```
conan remote add scine https://scine-artifactory.ethz.ch/artifactory/api/conan/public
```

Before we install Swoose, consider the options the package provides. They are summarized below, with valid values and the default option value in bold face:

- `python=[True, `**`False`**`]`: Build the Python bindings
- `tests=[True, `**`False`**`]`: Build and run tests
- `microarch=[detect, `**`none`**`]`: Tune build to CPU instruction set
- `database=[True, `**`False`**`]`: Builds the SCINE Database module

The base command to install Swoose is:

```
conan install scine_swoose/2.1.0@
```

Changes to default options can be supplied right after `conan install` with `-o scine_swoose:<option>=<value>`.

For instance, if you want Python bindings, your install command will be:

```
conan install -o scine_swoose:python=True scine_swoose/2.1.0@
```

Conan will proceed to identify your operating system, architecture and compiler. It will scan the dependencies of Swoose and try to find a prebuilt package for your system. If no package is available in binary form for your system, `conan` can handle its build with the addition of `--build=missing` as advised by `conan`'s error message. If the package is available in binary form, it will be downloaded.

By default, Conan does not modify the environment, it will just create the package in the local cache. In order to automatically add the required libraries to the `LD_LIBRARY_PATH` environment variable and to directly use the Swoose executable without specifiying its full location, you can execute:

```
conan install scine_swoose/2.1.0@ -g virtualrunenv
```

This will generate files that can be called to activate or deactive your environment via:

```
source activate_run.sh
source deactivate_run.sh
```

After activation, the Swoose executable is present in your `PATH`.

If you also want to install the modules Sparrow and xTB via Conan, you can create a file with the name `conanfile.txt`. It should have the following content:

```
[requires]
scine_swoose/[>=2.1.0]@
scine_sparrow/[>=5.0.0]@
scine_xtb_wrapper/[>=3.0.0]@

[options]
scine_swoose:database=False

[generators]
virtualenv
```

Subsequently, you can run the following command in the directory in which the file `conanfile.txt` is located:

```
conan install . --build=missing
```

Python bindings on PyPI

If you want to experiment with just the Python bindings, those are available as a package from PyPI for Linux platforms.[4] If you are using Linux, then you can install Swoose with:

[4] Caveat: Your CPU architecture must be x86–64 or i686.

```
python3 -m pip upgrade --user pip
python3 -m pip install --user scine_swoose
```

When installed this way, the Python bindings are not optimized to your particular CPU instruction set and may therefore be slower than if you had compiled them yourself.

From source code with CMake

In order to compile Swoose from its source code (available on www.scine.ethz.ch) you need

- a C++ compiler supporting the C++17 standard (we recommend gcc 8.3.0),
- cmake (we recommend version 3.14.7),
- the Boost library (we recommend version 1.69.0),
- the Eigen3 library (we recommend version 3.3.7), and
- the mongocxx library (we recommend version 3.4.0).

In order to compile the software, clone the repository with git, change to the source directory and execute the following steps:

```
git submodule init
git submodule update
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release \
      -DCMAKE_INSTALL_PREFIX=../install \
      -DSCINE_BUILD_PYTHON_BINDINGS=ON ..
make
make test
make install
export PATH=$PATH:<source code directory>/install/bin
export PYTHONPATH=$PYTHONPATH:<source code directory>/install/lib/<python version>/site-packages:\
       <source code directory>/install/lib64/<python version>/site-packages
export SCINE_MODULE_PATH=source code directory>/install/lib
```

This will configure everything, compile the software, run the tests, and install the software into the folder "install". Moreover, it will add the Swoose binary to your `PATH` such that you can use it without having to specify its full location. In this command, you have to replace `<source code directory>` with the full path where you stored the source code of Swoose. Finally, the Python library will be added to your `PYTHONPATH`. In this last command, you have to replace `<python version>` with the correct directory name, which depends on your Python version (e.g., it may be `python3.6` or `python3.7`).

By adding the option `-DSWOOSE_COMPILE_SPARROW=ON` to the execution of `cmake`, one can automatically compile the SCINE module Sparrow[5] along with Swoose.

Similarly, the option `-DSWOOSE_COMPILE_XTB=ON` enables the download and installation of the SCINE module xTB, which is a wrapper for the xTB program[6] by Stefan Grimme.

We strongly recommend compiling with support for the Sparrow and xTB modules as this will enable calculations with a variety of semi-empirical quantum chemistry methods.

[5] Bosia, F.; Husch, T.; Müller, C. H.; Polonius, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE Sparrow 5.0.0", `https://doi.org/10.5281/zenodo.3244105`, 2023

[6] Bannwarth, C.; Caldeweyher, E.; Ehlert, S.; Hansen, A.; Pracht, P.; Seibert, J.; Spicher, S.; Grimme, S. Extended tight-binding quantum chemistry methods, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020,** e01493

The `cmake` option `-DSWOOSE_COMPILE_DATABASE=ON` can be used to turn on the support for the SCINE Database module.

Note that Swoose also links to the SCINE libraries Molassembler[7] and Utilities [8].

In case you need support with the setup of Swoose, please contact us by writing to scine@phys.chem.ethz.ch.

[7] Bensberg, M.; Grimmel, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Reiher, M. "SCINE Molassembler 2.0.1", `https://zenodo.org/doi/10.5281/zenodo.4293554`, 2023

[8] Baiardi, A. *et al.* "SCINE Utilities 9.0.0", `https://zenodo.org/doi/10.5281/zenodo.3828691`, 2023

Using the Standalone Binary

Swoose is a command-line-only binary. There is no graphical user interface. Therefore, you always work with the Swoose binary on a command line such as the Gnome Terminal or KDE Konsole.

With the command

```
swoose -h
```

the help message of the binary is printed. It gives an overview of the command-line options of the binary. These options are explained in Tab. 1.

For example, if a molecular mechanics model shall be parametrized for a molecule stored in the file "struc.xyz" and with the settings present in the file "settings.yaml", one applies the command:

```
swoose -m parametrize -s struc.xyz -y settings.yaml
```

Subsequently, a molecular dynamics trajectory starting from this the same molecular structure can be generated by

```
swoose -m md -s struc.xyz -y settings.yaml
```

and an MM-level Hessian matrix can be calculated by the following command:

```
swoose -m calculate -s struc.xyz -y settings.yaml -H
```

In the following chapters, the available settings for the different program modes ("-m" option) are presented in detail. As many settings have default values, it is sometimes optional to provide any. The exceptions are clearly described in this manual.

The settings input file is based on the YAML format[9], but as all settings are of a simple key/value pair format it can always be written in the following form:

[9] Example calculations, including YAML settings files, can be found in section Example Calculations.

```
setting_key_1: setting_value_1
setting_key_2: setting_value_2
setting_key_3: setting_value_3
```

Table 1: The command-line options of the Swoose binary.

option	description
-h [--help]	Prints the help message of the binary.
-m [--mode] *arg*	Sets the mode of the binary. The current options are: *prepare-analyze* (detects errors in enzyme structures, e.g. PDB's), *prepare-protonate* (Protonates a given input structure), *prepare-finalize* (for a detailed description, see Section), *calculate* (single point calculation, by default with the SFAM molecular mechanics model), *parametrize* (parametrizes the SFAM force field), *optimize* (optimizing the molecular structure), *md* (performing a molecular dynamics simulation), *select_qm* (performs an automated QM region selection)
-y [--settings] *arg*	Sets the path to the settings file (YAML file format).
-s [--structure] *arg*	Sets the path to the molecular structure input (XYZ file format).
-q [--quantum]	Activates the use of a QM/MM hybrid model instead of pure molecular mechanics.
-H [--hessian]	Activates the option to calculate the Hessian. This option is only available in the *calculate* mode and without the "-q" option.
-v [--verbose]	Activates a more verbose output from the binary.

Using the Python Library

After the installation, one can also use the Swoose Python library. It can be imported into a Python script or Python console via:

```
import scine_swoose as swoose
```

This library provides the same functionalities as the standalone binary. The function names correspond to the different modes of the binary combined with the "-q" option for whether a QM/MM hybrid model shall be applied. Therefore, the available functions are:

- `parametrize(structure_file, **kwargs)`
- `calculate_mm(structure_file, **kwargs)`
- `calculate_qmmm(structure_file, **kwargs)`
- `optimize_mm(structure_file, **kwargs)`
- `optimize_qmmm(structure_file, **kwargs)`
- `md_simulate_mm(structure_file, **kwargs)`
- `md_simulate_qmmm(structure_file, **kwargs)`
- `select_qm_region(structure_file, **kwargs)`
- `prepare_analyze(structure_file, **kwargs)`
- `prepare_protonate(structure_file, **kwargs)`
- `prepare_finalize(structure_file, **kwargs)`

Each function takes the path to the XYZ file with the molecular structure ("-s" option for the binary) as its first argument and several keyword arguments after that. The settings unique to each of the functions are not put into a YAML file as for the binary, instead they are provided as keyword arguments to the functions. Furthermore, the options "-H" and "-v" of the binary can also be set in the Python functions by adding keyword arguments with the keys `hessian` and `verbose`, respectively.

By typing the following into a Python console,

```
import scine_swoose as swoose
help(swoose)
```

one can also get an overview of all the available functions listed above.

For example, if you want to perform a single point calculation with the SFAM molecular mechanics model while also obtaining the Hessian matrix and a more verbose output, the function call is:

```
calculate_mm("struc.xyz", hessian=True, verbose=True, s1="test", s2=100)
```

In this example, the molecular structure was assumed to be stored in the file "struc.xyz" and two settings were added with the keys "s1" and "s2". Note that there exists an alternative way of writing the function call from above:

```
settings_dictionary = {"hessian" : True, "verbose" : True, "s1" : "test", "s2" : 100}
calculate_mm("struc.xyz", **settings_dictionary)
```

The Parametrization of SFAM

As described in the previous chapter, a SFAM molecular mechanics model is parametrized in full automation using the program mode *parametrize* ("-m" option). For large molecular systems, an automated fragmentation protocol is applied first, so that reference data can be computed for the resulting fragments.

After a successful parametrization, the system-focused parameters as well as the connectivity information of the system are written to two separate files. The paths to these files can be specified by the following settings:

- mm_parameter_file: Sets the path to the file to which the parameters are written. Default: *Parameters.dat*
- mm_connectivity_file: Sets the path to the file to which the connectivity of the system is written. Default: *Connectivity.dat*. If the file that is specified with this setting is already present before the parametrization (also if it is the default), the initial connectivity of the system is read from it instead of employing a covalent radii-based guess connectivity.

If the molecular system that is parametrized does not have a molecular charge of 0 and a spin multiplicity of 1, one has to specify the path to an atomic information file by the following setting:

- atomic_info_file: Sets the path to the atomic information file. By default, it is set to an empty string and therefore it is assumed that the molecular system has a total charge of 0 and a spin multiplicity of 1.

The atomic information file is a text file listing the formal charges of the atoms along with the associated number of unpaired electrons. Only atoms that either carry a formal charge (not equal to zero) or unpaired electrons have to be listed. Each line of this file represents the information given for one relevant atom. The first number given in that line is the atom index (starting at 0), the second one is the formal charge and the third number provides the number of unpaired electrons (separated by whitespaces). An example file looks like:

```
0     0   1
3    -1   0
172   2   3
```

In this example, atom 0 has no formal charge, but one unpaired electron. Atom 3 has a formal charge of -1 and no unpaired electrons. The atom with the index 172 carries a formal charge of 2 and 3 unpaired electrons.

For the parametrization, reference data has to be generated. This includes the optimized structures for the molecule or the molecular fragments, Hessian matrices, atomic partial charges (preferably CM5 charges[10]), and Mayer bond orders[11]. The latter does not have to be calculated, but instead the initial guess connectivity of the system based on distance rules can be employed.

[10] Marenich, A. V.; Jerome, S. V.; Cramer, C. J.; Truhlar, D. G. Charge model 5: An extension of Hirshfeld population analysis for the accurate description of molecular interactions in gaseous and condensed phases, *J. Chem. Theory Comput.* **2012,** *8,* 527–541

[11] Mayer, I. Charge, bond order and valence in the ab initio SCF theory, *Chem. Phys. Lett.* **1983,** *97,* 270–274

- refine_connectivity_qm: Sets whether the inital guess connectivity is refined by quantum-mechanically-derived bond orders provided as part of the reference calculations. By default, this setting is set to *true*.

The reference data can be calculated with a variety of programs. For instance, this includes the ORCA[12] and the TURBOMOLE[13] quantum chemistry software, as well as the SPARROW[14] and xTB[15] modules of SCINE. The availability of these options might be limited depending on the reference data generation mode selected (more details below). The following settings apply:

[12] Neese, F. The ORCA program system, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2012,** *2,* 73–78

[13] Ahlrichs, R.; Bär, M.; Häser, M.; Horn, H.; Kölmel, C. Electronic structure calculations on workstation computers: The program system Turbomole, *Chem. Phys. Lett.* **1989,** *162,* 165–169

[14] Bosia, F.; Husch, T.; Müller, C. H.; Polonius, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE Sparrow 5.0.0", `https://doi.org/10.5281/zenodo.3244105`, 2023

[15] Bannwarth, C.; Caldeweyher, E.; Ehlert, S.; Hansen, A.; Pracht, P.; Seibert, J.; Spicher, S.; Grimme, S. Extended tight-binding quantum chemistry methods, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020,** e01493

- reference_program: The program calculating the reference data. The options are: *orca* (default), *turbomole*, *sparrow*, and *xtb*. This setting is case-insensitive. Note that for the *direct* reference data generation mode, the option *orca* requires the ORCA program binary path to be set as an environment variable, ORCA_BINARY_PATH. Likewise, the option *turbomole* requires the environment variable TURBODIR to be set to the TURBOMOLE root directory. The options *sparrow* and *xtb* require prior installation of the corresponding SCINE modules (see section Installation).
- reference_method: The method for the reference calculation. The string for the method needs to be specified in the same way as it would be written in a typical input file of the corresponding program. Note that for D3 dispersion corrections, one can apply the ORCA format to the case of TURBOMOLE as well, i.e., appending

the corresponding ORCA keyword (see default string below) to the density functional. If this setting is not specified or set to an empty string, the default is set based on the selected reference program. For *orca*, it is *PBE-D3BJ*, for *turbomole* it is *pbe-D3BJ*, for *xtb* it is *gfn2*, and for *sparrow* it is *pm6*. When selecting the program option *sparrow*, check the SPARROW manual[16] for available methods (e.g., *pm6*). For the xTB module, the methods *gfn0*, *gfn1*, and *gfn2* are available methods.

[16] Bosia, F.; Husch, T.; Müller, C. H.; Polonius, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE Sparrow 5.0.0", https://doi.org/10.5281/zenodo.3244105, 2023

- reference_basis_set: The basis set for the reference calculation. The defaults for the respective programs are applied if the reference method was not specified. The defaults for *orca* and *turbomole* are *def2-SVP* and the defaults for *xtb* and *sparrow* are empty strings. The string for the basis set needs to be specified in the same way as it would be written in a typical input file of the corresponding program, in analogy to the method setting. The methods of the SPARROW and xTB modules require to specify the basis set as an empty string if a non-default reference method is selected: " ".

- external_program_nprocs: The number of processors to use in the calculations. This setting applies for the program options *orca*, *turbomole*, and *xtb*. By default, it is one, i.e., serial calculations are carried out.

- external_program_memory: The total amount of memory in MB that should be available for the external program. By default set to *1024*.

However, there also exists the option to apply the GAUSSIAN quantum chemistry software[17] to calculate the CM5 charges (only the CM5 charges, not the rest of the reference data). By default, ORCA calculates Hirshfeld charges, which are then converted to CM5 charges by SWOOSE. TURBOMOLE calculates Löwdin charges instead of Hirshfeld charges, however, these are also converted by the same algorithm to obtain approximate CM5 charges. In general, the conversion with the CM5 algorithm can be disabled with the following setting:

[17] Frisch, M. J. *et al.* "Gaussian 09, revision D. 01", 2009 Gaussian, Inc., Wallingford CT

- convert_charges_cm5: This setting decides whether to apply the CM5 algorithm to the calculated atomic charges. By default, it is *true*. For reference calculations with SPARROW or xTB (these calculate Mulliken charges), we recommend setting this option to *false* in order to obtain more accurate atomic charges for most systems.

Note that there may be small deviations in the resulting charges if GAUSSIAN is applied. The reasons for this are still investigated, however, we emphasize that our charges are in agreement with the ones from the original paper and the charges obtained by the script available at https://github.com/leelasd/ligpargen. The following settings apply for the use of GAUSSIAN:

- use_gaussian: Enables the CM5 reference calculation with GAUSSIAN instead of the reference program. By default, it is set to *false*.
- gaussian_method: The method for the CM5 calculation with GAUSSIAN. By default, it is set to *PBEPBE*. The string for the method needs to be specified in the same way as it would be written in a typical GAUSSIAN input file.
- gaussian_basis_set: The basis set for the CM5 calculation with GAUSSIAN. By default, it is set to *def2SVP*. The string for the basis set needs to be specified in the same way as it would be written in a typical GAUSSIAN input file.

As mentioned above, reference data can be calculated using one of four available modes. The mode is specified by the setting:

- ref_data_mode: The reference data generation mode. Options are: *direct* (default), *database*, *write*, and *read*. Details are given below.

Option ***database***:

Reference calculations are processed through a MONGODB database[18] and the Python program PUFFIN. The following settings apply:

[18] MongoDB Inc., “MongoDB 3.2”, `www.mongodb.com`, visited on 2024-02-13

- database_host: The hostname or IP adress of the database host. The default is set to *localhost*.
- database_port: The port at which the database is reached. Default is set to *27017*.
- database_name: The name of the MONGODB database. The default is set to *scine_swoose_mm_parametrization*.
- database_sleep_time: The time in seconds the SWOOSE binary waits after it has checked the database for results until it checks again. The default is set to *60*.

Furthermore, two more options exist, which can only be activated in *database* mode:

- reuse_database: If this setting is set to *true*, the parametrization uses all of the reference data that is already present in the specified database. This allows to terminate the SWOOSE app any time during the reference data generation stage and restart it without re-calculating any data. The default is *false*.

- ref_data_generation_only: If this setting is set to *true*, the parametrization procedure automatically terminates after the reference data generation step. It can then later be restarted with the option above. Default: *false*.

Option ***direct***:

Reference calculations are processed directly through the SWOOSE binary. The following setting applies:

- base_working_directory: The path to the directory in which the reference calculations are performed. The default is set to the current working directory.

Further settings that are available when performing calculations with the ORCA, TURBOMOLE and *xTB* SCINE wrappers or the SPARROW module, can be added to the YAML settings file. For a detailed description of these settings, check the manuals of the READUCT[19] and the SPARROW[20] modules of SCINE.

[19] Bensberg, M.; Brunken, C.; Csizi, K.-S.; Grimmel, S. A.; Gugler, S.; Sobez, J.-G.; Steiner, M.; Türtscher, P. L.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE ReaDuct 5.1.0", `https://zenodo.org/doi/10.5281/zenodo.3244107`, 2023

[20] Bosia, F.; Husch, T.; Müller, C. H.; Polonius, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE Sparrow 5.0.0", `https://doi.org/10.5281/zenodo.3244105`, 2023

Option ***write***:

Instead of calculating the reference data right away, all information needed for the calculations can be written to disk instead. The following setting applies:

- ref_data_directory: The path to the directory to which the information about the reference calculations is written or from which it is read. The default is *reference_data*.

The following directory is created for each molecular structure/fragment: `<ref_data_directory>/<fragment_index`. If no fragmentation was performed, the directory for the full system is `<ref_data_directory>/0`. In these directories, the following files are written: `fragment.xyz` (the fragment structure), `info.dat` (containing the charge and spin multiplicity values for the fragment separated by whitespaces) and optionally `constr.dat` (containing the indices of the atoms which must be constrained during the structure optimization separated by whitespaces) if any constrains apply for the corresponding fragment. For superfluous fragments, which are identical to others and therefore do not require reference calculations, the directory is created but left empty. If no fragmentation took place, the full system's structure can be found in the file `<ref_data_directory>/0/molecule.xyz` and its charge and multiplicity in `<ref_data_directory>/0/info.dat`.

Option ***read***:

If this option is chosen, the reference data is read from the same directory structure introduced for the previous option. The same setting (ref_data_directory) applies. Two options exist for the file formats from which the data is read. One can switch between the two options via the setting:

- use_csv: Decides whether to use the generic CSV file format to read in Hessians and atomic forces. Moreover, if this is set to *true*, the bond orders are read in from the Swoose connectivity file format. If the setting is set to *false*, the data is read from Orca or Turbomole output files, respectively (in this case, it is important to specify the setting reference_program to indicate the format of the output files). The default of this setting is *true*, i.e., the generic CSV format is employed.

However, the optimized structure is always read from an `opt.xyz` file in XYZ file format. This file must be present in each directory (for each fragment). Furthermore, if Turbomole is the reference program, a file with the name `coord` must also be present. The format of this `coord` file is the standard Turbomole coordinates file format.

In the case of use_csv being *false*, the following files must be present in each directory depending on the selected reference program:

type of property	**filename**	
	Orca	Turbomole
Hessian	`orca.hess`	`hessian`
atomic charges	`hirshfeld.out`	`ridft.out`
bond orders	`bonds.out`	`ridft.out`

For Turbomole, these files are the standard output files obtained from a Turbomole calculation. For Orca, the Hessian is parsed from the separate Orca Hessian output file and the atomic charges and bond orders are parsed from regular Orca output files that contain these properties. Hence, the files `hirshfeld.out` and `bonds.out` may be identical regarding their content, however, both files must be present.

The bond orders file is only needed if the option for QM connectivity refinement was set to *true*. If the option to use Gaussian was enabled, a file with the name `cm5.out` (Gaussian output file with CM5 charges) needs to be present instead of the atomic charges file. Regardless of the setting convert_charges_cm5, the Hirshfeld charges in ORCA or the Loewdin charges in TURBOMOLE are always converted to CM5 charges when provided via the output file. Note that for ORCA, the bond orders are Mayer bond orders, while for TURBOMOLE, Wiberg bond orders are calculated.

If one employs other quantum chemistry software to calculate the reference data, the generic input format can be used instead (`use_csv` should be set to *true*, which is the default). For the connectivity of each fragment, this format is the connectivity file format, which is also applied to encode the resulting connectivity of the whole system. In this format, the lines of the file represent the atoms of the system in order and each line contains the indices of the covalently bonded atoms (indices start at zero) separated by whitespaces.

An example of this file for a water molecule (oxygen is at position zero) would be:

```
1 2
0
0
```

This is because the zero-th atom (oxygen) is bonded to the atoms 1 and 2 (hydrogens) and the atoms 1 and 2 are each bonded to atom 0. Note that an incorrect connectivity file (e.g., atom 1 is bonded to atom 2, but not *vice versa*) results in an error. This connectivity file must be present in each directory (for each fragment) with the name `connectivity.dat`.

For the Hessian matrix and the atomic charges, the files `hessian.csv` and `atomic_charges.csv` must be present, respectively. These files contain the corresponding data in the CSV file format with a *comma* as the delimiter. As an example, this file could look like this for a 3x3 matrix:

```
0.14,1.83,2.8
-3.8,9.114,0.023
0.11,-0.14,2.29
```

For the atomic charges, the file `atomic_charges.csv` might look like

```
0.25
-0.12
-0.18
0.02
0.03
```

for a molecule with 5 atoms. Note that the atomic charges should ideally be Hirshfeld charges. As for the other reference data generation modes, the setting convert_charges_cm5 decides whether a CM5 correction is applied afterwards (default is *true*).

Furthermore, there exist the following additional settings:

- number_atoms_threshold: The maximum number of atoms for which reference data is still generated for the whole system and no fragmentation algorithm is applied. The default is *150*.
- subsystem_radius: The initial radius of the spheres defining the volume of the fragments. The unit is Å. The default is *6.0*.
- bond_order_threshold: Sets the threshold for which bond orders to still consider as bonds. The default is *0.5*.
- atom_type_level: Sets the atom type level, i.e. how atom types are defined for the MM model. Options are: *elements* (each element is one atom type), *low* (the number of covalently bonded atoms is included), *high* (the exact composition of the covalently bonded atoms is included, which is the default option) and *unique* (each atom is its own atom type, which is not recommended due to risk of instabilities in the parameter optimization).
- constrain_mm_parameters: Decides whether there should be constraints during the MM parameter optimization. This is recommended and the default is therefore set to *true*.
- optimize_improper_dihedral_force_constants: Decides whether the improper dihedral force constants of non-planar groups should be optimized during the MM parameter optimization or whether their value should be fixed at zero. The default is set to *true*, i.e., the force constants are optimized.
- increase_scf_safety: Decides whether to apply safer SCF convergence settings when using the ORCA or TURBOMOLE reference program. In both cases, SCF damping is switched on and for TURBOMOLE an orbital shift of 0.5 Hartree is applied additionally. This setting currently only works in the *database* mode.
- existing_parameters: The path to an already existing SFAM parameter file, which contains parameters that can be re-used for the current parametrization. If a parameter is available in that file, it is not optimized again. Reference data is only calculated for the fragments, which are still needed. This setting currently only works if the atoms and their order are identical between the two

systems that are considered and just their connectivity varies. This will be extended in the future. The default is an empty string, i.e., no existing parameters are considered and the parametrization is carried out normally.

- enable_early_termination: Decides whether the reference data generation in *database* mode will be terminated once enough data for a parametrization has been collected. This may speed up the parametrization procedure significantly. The default is set to *true*.

Calculations With SFAM

For a single point calculation with the SFAM molecular mechanics model (with already existing parameters for the molecular system), the following settings are available:

- mm_parameter_file: Sets the path to the file from which the parameters are read. The default is an empty string. This setting must therefore always exist.
- mm_connectivity_file: Sets the path to the file from which the connectivity of the system is read. The default is an empty string. This setting must therefore always exist if the bond detection by covalent radii is disabled.
- covalent_radii_bond_detection: Ignore the connectivity file setting and apply a bond detection algorithm based on distance rules. By default, it is set to *false*.
- atom_type_level: Sets the atom type level, i.e. how atom types are defined for the MM model. Make sure that the atom type level matches the one employed during parametrization. Options are: *elements* (each element is one atom type), *low* (the number of covalently bonded atoms is included), *high* (the exact composition of the covalently bonded atoms is included, which is the default option) and *unique* (each atom is its own atom type).
- covalent_contributions_only: Ignore all non-covalent potentials in the MM model. By default, this setting is set to *false*.
- hydrogen_bond_correction: Apply potentials to model hydrogen bonds. By default, it is set to *true*.
- non_covalent_cutoff: The cutoff radius for non-covalent interactions in Å. By default, this value is set to *12.0*.
- apply_cutoff_during_initialization: This setting decides whether the non_covalent_cutoff is enforced during the initialization stage of the calculation to exclude the interactions beyond the distance

cutoff. This results in a larger computational speed-up related to the cutoff setting. However, a consequence is that, for instance during an MD simulation, these interactions are not included either for structures later visited during a simulation, for which the interaction may have fallen below the distance threshold. Setting this option to *true* is recommended for single-point Hessian calculations. For MD simulations and structure optimizations, use it carefully. The default is *false*. Note that if this setting is *false*, the non-covalent cutoff is still applied, by including the interaction terms in the model and evaluating them to zero in a calculation, which results in a smaller speed-up, but allows for more flexibility during an MD simulation or structure optimization.

- print_mm_contributions: Enables a very verbose output level, where the energy contributions of all types of potentials are printed individually. By default, this is set to *false*. If the verbose output flag of the command-line app ("-v" option) is not set, this setting will be ignored.

Calculations With Other Molecular Mechanics Models

Currently, Swoose also offers the GAFF method[21] as an alternative to SFAM for molecular mechanics calculations. It can also be applied in QM/MM calculations. To enable it, one can set the following setting:

[21] Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field, *J. Comput. Chem.* **2004,** *25,* 1157–1174

- mm_model: The type of molecular mechanics model to apply. The current options are *SFAM* and *GAFF* with the default being *SFAM*.

The following settings, which are explained in the section Calculations With SFAM, are also available for GAFF:

- mm_parameter_file
- mm_connectivity_file
- covalent_radii_bond_detection
- covalent_contributions_only
- non_covalent_cutoff
- apply_cutoff_during_initialization

The default for the setting mm_parameter_file is an empty string. This default results in the standard GAFF parameters being employed (version 1.4, March 2010). Of course, one can also provide a custom parameter file as long as it complies with the standard GAFF parameter file format. Furthermore, the following settings are only available for GAFF (not for SFAM):

- gaff_atomic_charges_file: The path to a file containing the atomic charges of the system to be applied in the GAFF calculation for the electrostatic interactions. This file should be a simple text file with the atomic partial charge of each atom on a separate line. Note that the order of the atoms must be the same as in the structure file. For an example, see the information about the atomic charges file for the *read* reference data mode in section The Parametrization of SFAM. The default of this setting is an empty string, i.e., only atomic charges of zero are used in the GAFF calculation.

- gaff_atom_types_file: The path to a file containing the atom types of the system. The default is an empty string, which means that the automated atom type identifier will be applied instead. Note that our automated atom type identifier is still in beta phase and may assign some atom types incorrectly (especially for aromatic systems), which can lead to missing parameters. The format of this file is as follows: the file contains one atom type (case-insensitive) in each line, hence, there should be as many lines with atom types as there are atoms in the system (the order must be equal to the order of atoms in the provided XYZ file). Note that an empty line will result in the file parsing to stop, which means one can add additional comments to the file after such an empty line was inserted.

The Hybrid QM/MM Model

When applying the QM/MM hybrid model of SFAM ("-q" option), the same settings apply as for a regular MM-only calculation. Furthermore, the settings of the chosen QM method can also be set.[22] However, there are some additional settings, namely:

[22] For example, the DFT method of choice can be specified with the method and basis_set settings, as described in the manual of the SCINE module ReaDuct.

- qm_atoms: The atoms in the QM region. This setting is given as a list (e.g., `[0, 7, 12, 32, 42]`) containing the atom indices of these atoms (indices start at 0). There is no default for this setting. It must be set by the user. Note that one should only cut through single bonds at the QM–MM boundary when choosing the QM region as only these can be saturated correctly by hydrogen link atoms in the current state of the program.

- qm_module and qm_model: The SCINE module from which to take the QM calculator and what model (calculator name) should be taken from that module. If you want to use Orca or Turbomole, you can set both the module and the model to *orca* or *turbomole*, respectively. If PM6 is supposed to be the QM method, the strings *sparrow* and *PM6* need to be set. If GFN2-xTB is supposed to be the method, set the strings *xtb* and *GFN2*. The modules that are currently available are *Orca*, *Turbomole*, *Sparrow* and *xTB*. These settings are case-insensitive.

- electrostatic_embedding: Sets whether electrostatic embedding is applied. The alternative is mechanical embedding. The default is *true* (currently only available with Orca or Turbomole as the QM program). Note that Swoose automatically writes different point charges file formats for the two available calculators.

- qm_region_file: The path to the file to which the QM region is written in XYZ format. By default it is set to an empty string, which results in no such file being created.

- charge_redistribution: The scheme to apply for the redistribution of charges close to the QM–MM boundary. The two options are *rc* and *rcd*. The former implements a simple redistribution of charges (RC) scheme, which is described in Fig. 2 of the review article by Senn and Thiel[23]. The latter implements a redistribution of charges and dipoles (RCD) scheme as described by Truhlar et al.[24]. The default is set to *rc*.

- reduced_qmmm_energy: Calculate the reduced QM/MM energy as it is defined in our recent publication[25], i.e., that any covalent MM contributions and the noncovalent interactions within the environment are neglected. The default is *false*.

[23] Senn, H. M.; Thiel, W. QM/MM methods for biomolecular systems, *Angew. Chem. Int. Ed.* **2009,** *48,* 1198–1229

[24] Lin, H.; Truhlar, D. G. Redistributed charge and dipole schemes for combined quantum mechanical and molecular mechanical calculations, *J. Phys. Chem. A* **2005,** *109,* 3991–4004

[25] Brunken, C.; Reiher, M. Automated Construction of Quantum–Classical Hybrid Models, *J. Chem. Theory Comput.* **2021,** *17,* 3797–3813

Automated QM Region Selection

A QM region can be generated in full automation using the program mode *select_qm* ("-m" option). One can choose either to generate only a single QM region deterministically or to generate multiple QM region candidates via our sampling method, for which the optimal QM region is then evaluated based on the atomic forces close to the center of the QM region.[26] The important setting that decides which option is applied, is the following:

[26] Brunken, C.; Reiher, M. Automated Construction of Quantum–Classical Hybrid Models, *J. Chem. Theory Comput.* **2021,** *17*, 3797–3813

- cutting_probability: The probability that a bond that is considered cleavable is actually cut during the generation of the QM region. If it is set to *1.0* (which is the default), the deterministic option is chosen, and only a single QM region is generated. If the value of this setting is less than one, several QM regions are generated. Subsequently, reference data is calculated with each of the generated models to evaluate the optimal selection. The random seed for this generation can be set via qm_region_selection_seed (default is *42*).

Furthermore, one can set the initial radius for the QM region generation:

- initial_radius: The initial radius of the sphere defining the volume of the QM region. The unit is Å. The default is *6.0*. The value must be between 2 and 12 Å.

If the setting cutting_probability is set to a value less than *1.0*, N generation attempts are performed with the initial radius and then the radius is iteratively increased by 0.1 Å until each of the N attempts per radius generates a QM region that is larger than the maximum allowed size for the reference models or until each of the N attempts generates the full system as the QM region. The corresponding settings are:

- num_attempts_per_radius: The number of attempts N described above. The default is *100*.

- ref_max_size: The maximum number of atoms that the QM regions of the reference models are allowed to contain. If the full system contains fewer atoms than this number, a QM calculation on the full system will be the only reference. Otherwise, the minimum number of atoms in the QM region of the reference models is automatically set to 95% of this value. The default is *200*.

If cutting_probability is set to the value *1.0*, only a single QM region is generated with the initial fragmentation radius of initial_radius. Regardless of the choice for cutting_probability, an important setting is:

- qm_region_center_atoms: The indices of the atoms (indices start at zero) around which the QM region will be constructed. The default is [*0*]. Note that you can also add multiple atoms to this list of QM atoms to be included in the QM region. If one chooses more than one QM atom to be included in the QM region, the symmetry of the generated overall QM region will not be considered.

If more than one QM region is generated, the following settings apply for the QM region sizes of the candidate models:

- qm_region_min_size: The minimum number of atoms in the QM region (including link atoms) for the candidate models. The default is *100*.

- qm_region_max_size: The maximum number of atoms in the QM region (including link atoms) for the candidate models. The default is *120*. This number must always be larger than or equal to the minimum number of atoms. Furthermore, it must be smaller than or equal to the maximum number of atoms in the QM regions of the reference models.

The number of reference models that one wants to employ can be limited. The corresponding setting is:

- max_num_ref_models: The maximum number of reference models that are employed to obtain the reference forces. The default is *10*. Note that the last N_{ref} models that were generated are used, while the first ones are discarded.

Moreover, one must specify the path to the connectivity file:

- mm_connectivity_file: Sets the path to the file to which the connectivity of the system is written. Default: *Connectivity.dat*.

The reference data can be calculated via a database mode similar to the database mode of the SFAM parametrization or in a direct mode (default). If the mode *database* is chosen via the setting ref_data_mode, the four database-related settings database_host, database_port, database_name and database_sleep_time also apply for the QM region selection task. The same defaults are available as described in the section The Parametrization of SFAM. One must also set all the settings that shall be applied for the QM/MM reference calculations. These settings are described section The Hybrid QM/MM Model. The most important ones are qm_module, qm_model, and the mm_parameter_file. Obviously, the settings qm_atoms, molecular_charge and spin_multiplicity must not be set manually, but are set by the program automatically for each QM/MM candidate or reference model.

As mentioned above, the mode *direct* is the default option for the setting ref_data_mode. With this option, the QM/MM calculations are directly excuted by the Swoose binary. The settings described in the section The Hybrid QM/MM Model also apply. The QM/MM reference calculations can be parallized by setting the OMP_NUM_THREADS environment variable to the number of desired parallel calculations. Note that the number of threads set by this variable multiplied by the number of cores chosen for one reference calculation (via setting external_program_nprocs) should not exceed the system limit.

Per default, the candidate models have a symmetry score m_{sym}. If this score is 50% larger than the minimum among all candidates, the candidate is discarded. The calculations for the discarded calculations are written to the database, but their status is set to *hold*. The symmetry score m_{sym} is defined as,

$$m_{\text{sym}} = \frac{\bar{r}_{\text{LDM}}}{\bar{r}_{\text{MDQ}}} \quad , \tag{1}$$

where $\bar{r}_{\text{LDM}}$ is the mean distance of the central atom to the three least distant MM atoms (LDM) and $\bar{r}_{\text{MDQ}}$ is the mean distance of the central atom to the three most distant QM atoms (MDQ). This tolerance percentage of 50% can be modified via the setting:

- tol_percentage_sym_score: Tolerance for the symmetry score of a candidate model in percent of the minimum symmetry score among all of the candidate models. *50.0* is set as the default.

After the reference data is calculated, the model with the QM region that contains the smallest number of link atoms is selected, if that model's mean error on the forces is small enough. This is the case if this error is within an error tolerance of the candidate with the minimum error. This tolerance is calculated as a percentage of the minimum error. The setting is:

- tol_percentage_error: Tolerance for mean error of the forces in percent of the minimum error among all of the candidate models. *20.0* is set as the default. Note that errors below $1 \cdot 10^{-4}$ are considered zero when checking against this tolerance.

Structure Preparations

The parametrization of an SFAM molecular mechanics model requires an atomistically resolved, error-free molecular structure file. To process biomolecular structures in Swoose, the program modes *prepare-analyze*, *prepare-protonate*, and *prepare-finalize* can be called in a consecutive manner. Alternatively, if no operator interaction is intended, a fully automated mode can be called with the program mode *prepare-automate*. The mode *prepare-analyze* automatically detects errors in a given input structure, and separates the structure into a NRMC (non-regular-module-container), and a RMC (regular-module container). These containers hold the erroneous, and error-free substructures of the full system. The mode *prepare-protonate* protonates these subsystems separately, and *prepare-finalize* merges the structures back together, corrects the boundaries, assigns partial charges and generates a connectivity file required for a subsequent SFAM parametrization. In between each step, the operator can check the generated results, apply changes to both containers by directly manipulating the output structure files, and then start the next step. If the operator is confident in the quality of the input structure and wants to carry out the preparation in a fully automated manner, all 3 modes are called via *prepare-automate*. Any file format supported by the Scine infrastructure can be provided as input. The following settings can be applied:

- preparation_directory: Sets the path to the parent directory for structure preparation. The default is *preparation_data*. The following directory is generated:
  `<preparation_directory>/<system_name>`
  to which all files and output data are written. `system_name` is the input structure file name without trailing separator and suffix.

For example, if the file `plastocyanin.pdb` shall be prepared, one applies the following commands:

```
swoose -m prepare-analyze -s plastocyanin.pdb
```

This action generates a directory `preparation_data/plastocyanin/`, and therein, the files `nonregular_container.xyz` and `rmc.pdb`.

```
swoose -m prepare-protonate -s plastocyanin.pdb
```

This action writes the files `nonregular_container_H.xyz`, `rmc_H.xyz`, `atomic_info.dat`, `atomic_info_nrmc.dat` to the preparation directory generated in the first step.

```
swoose -m prepare-finalize -s plastocyanin.pdb
```

This action writes the final processed structure to a file `system.xyz`, and the corresponding connectivity information to `Connectivity.dat`. Both files are also stored in the preparation directory generated in the first step. Furthermore, the information stored in `atomic_info_nrmc.dat` are correctly appended to the final atomic information file `atomic_info.dat` accounting for index mapping of the NRMC atoms in the final structure.

Alternatively, one may call:

```
swoose -m prepare-automate -s plastocyanin.pdb
```

This action executes the above three protocols automatically one after the other.

If the input structure file contains multiple structural conformers or ensembles of molecular models that are condensed in a single file (as it is often the case for PDB files), these models/conformers can be separated via the setting:

- separate_overlaying_substructures: Splits up substructures. Note that this setting is only supported for PDB input files, as the alternate location indicator in the PDB file format is parsed.

Protonation of the NRMC is carried out using the protonation functionality provided by OpenBabel. Therefore, the OpenBabel binary must be present in the PATH. For protonation, the following settings can be applied:

- pH_value_for_protonation: Sets the pH value. By default, it is set to 7.0.
- charged_termini: Whether to protonate C and N termini in their charged (NH_3^+, COO^-) form. The default is set to *true*.

If atoms in the molecular system have partial charges and/or spin multiplicities unequal to 1, these information are written to an atomic information file. The path to this file can be set by the following setting:

- atomic_info_file: Sets the path to the atomic information file. By default, it is set to *atomic_info.dat*.

Additionally, the user can add charge and multiplicities for atoms in the NRMC to the automatically generated file *atomic_info_nrmc.dat*. Please use the corresponding indices of the cofactor structure starting from 0.

After protonation and merge of the substructures, the resulting molecular system can be solvated. The following settings apply:

- solvate_structure: Sets whether the structure is solvated using the solvation algorithm as implemented in the SCINE framework. By default, this setting is set to *false*.
- num_solvent_shells: The number of solvent shells. By default, this is set to 1. Attention: Multiple solvent shells may be computationally demanding and thus time consuming.

Furthermore, all solvent-specific settings provided within the SCINE infrastructure can be set.

The final structure can be found in: `<preparation_directory>/<system_name>/system.xyz`

Structure Optimizations

For structure optimizations in Swoose, the BFGS algorithm is applied as described in the manual of SCINE ReaDuct[27]. All the settings available in ReaDuct for this algorithm as well as all general optimization settings are also available in Swoose. Furthermore, the settings of the methods discussed in the previous chapters are available.

[27] Bensberg, M.; Brunken, C.; Csizi, K.-S.; Grimmel, S. A.; Gugler, S.; Sobez, J.-G.; Steiner, M.; Türtscher, P. L.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE ReaDuct 5.1.0", `https://zenodo.org/doi/10.5281/zenodo.3244107`, 2023

After a completed optimization, the optimized molecular structure is written to the file `opt_structure.xyz` and the trajectory of the optimization procedure is written to `opt_trajectory.xyz`. These files can be found in a results directory, which can be specified by the following setting:

- results_directory: The directory to which the results of structure optimizations are written to. By default, it is set to *opt_results*.

For QM/MM structure optimizations, a special microiteration-based algorithm is applied, for which the following settings can be set:

- qmmm_opt_max_macroiterations: The maximum number of macrocycles allowed. One macrocycle contains one full system optimization and one environment-only optimization. Default: *30*.
- qmmm_opt_max_full_microiterations: The maximum number of full system optimization microcycles (with the full QM/MM gradients) allowed per macrocycle. The default is set to *15*.
- qmmm_opt_max_env_microiterations: The maximum number of environment-only (MM-only) optimization microcycles allowed per macrocycle. The default is set to *1000*.
- qmmm_opt_boundary_distance_thresh: The distance threshold in Å determining which atoms of the environment are also frozen during the environment-only microiterations. By default, it is set to *4.0*.
- qmmm_opt_env_switch_off: The number of macrocycles after

which the environment-only optimization is switched off. The default value is *12*.

- qmmm_opt_env_start: Decides whether the MM-only optimization of the environment is performed at the very beginning. By default, it is set to *true*.

Molecular Dynamics Simulations

When performing an MD simulation, all settings for the single point calculations are also available. Furthermore, the following MD-specific settings are:

- md_time_step: MD integration time step in femtoseconds. The default is *1.0*.
- number_md_steps: The number of MD steps that will be performed. The default is *10*.
- md_integration_scheme: The integration algorithm used in the MD simulation. Options are: *leap_frog* (default), *euler*, *velocity_verlet*, and *stochastic_dynamics*. The latter[28] includes temperature control.
- md_thermostat: The thermostat applied in the MD simulation. Currently, two options are available: *none* (no thermostat) and *berendsen* (Berendsen thermostat[29]). By default, it is set to *none*.
- generation_temperature: The temperature in Kelvin for which initial velocities are drawn from a Boltzmann distribution. The default is *300*. If it is set to zero, all initial velocities are set to zero.
- target_temperature: The target temperature in Kelvin, which is only a valid setting if the thermostat is *berendsen* or the integration scheme is *stochastic_dynamics*. The default is *0*, which means that the generation_temperature is used as the target_temperature.
- temperature_coupling_time: The temperature coupling time in femtoseconds. If a value of *0.0* is chosen (which is also the default), the defaults of the two different thermostats are applied. These defaults are: *10.0* (Berendsen), *2000.0* (Stochastic Dynamics).
- record_frequency: The frequency with which structures and their energies are written to disk during the simulation. The default is *1*, which means that the structure is written in every step. The structures are written to the file `MD_trajectory.xyz` and the energies to `MD_energies.dat`.

[28] Goga, N.; Rzepiela, A. J.; de Vries, A. H.; Marrink, S. J.; Berendsen, H. J. C. Efficient Algorithms for Langevin and DPD Dynamics, *J. Chem. Theory Comput.* **2012,** *8,* 3637–3649

[29] Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* **1984,** *81,* 3684–3690

- linear_momentum_removal_frequency: The frequency with which the linear momentum of the center of mass is removed. If zero, no action is taken. The default is *1*.
- angular_momentum_removal_frequency: The frequency with which the angular momentum of the center of mass is removed. If zero, no action is taken. The default is *1*.

Example Calculations

In this section, we provide some example YAML settings files and execution commands for a typical workflow with Swoose using the standalone binary.

Let's start by parametrizing a SFAM molecular mechanics model for a water molecule. First, one has to provide the coordinates in an XYZ file, which may look like this (unoptimized):

```
3
water molecule
H    0.75    0.00    0.45
O    0.00    0.00   -0.15
H   -0.75    0.00    0.45
```

Let's store it in a file with the name `water.xyz`. Second, create the following example YAML settings file with the name `settings.yaml`:

```
mm_parameter_file: final_parameters.dat
mm_connectivity_file: final_connectivity.dat
reference_program: ORCA
reference_method: PBE0-D3BJ
reference_basis_set: def2-TZVP
ref_data_mode: direct
refine_connectivity_qm: false
external_program_nprocs: 2
base_working_directory: /tmp/reference_calculations
max_scf_iterations: 150
```

Note that the directory `reference_calculations` will be created by Swoose automatically if it does not exist yet. In this example, we disabled the calculation of quantum chemically derived bond orders for the evaluation of the system connectivity, because the distance-based guess connectivity will be sufficient for such a simple molecule. The setting `max_scf_iterations` is an additional setting that can be set for Orca calculations in SCINE (to increase the maximum number of SCF iterations). The descriptions of such additional settings can be found in the manual of the SCINE ReaDuct module[30].

[30] Bensberg, M.; Brunken, C.; Csizi, K.-S.; Grimmel, S. A.; Gugler, S.; Sobez, J.-G.; Steiner, M.; Türtscher, P. L.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE ReaDuct 5.1.0", https://zenodo.org/doi/10.5281/zenodo.3244107, 2023

With the two files we created, one can start a parametrization via the console:

```
swoose -m parametrize -s water.xyz -y settings.yaml -v
```

The verbose output (requested via "-v" option) is then printed to the console and once the parametrization is completed, the files `final_parameters.dat` and `final_connectivity.dat` will be available in the directory from which the command above was executed.

Subsequently, we can perform a single-point calculation with our generated model for the unoptimized structure in the file `water.xyz`. The settings file may now look like this:

```
mm_parameter_file: final_parameters.dat
mm_connectivity_file: final_connectivity.dat
```

Additional settings are not necessary for our simple system. However, see section Calculations With SFAM for a list of all available settings. The calculation (including the Hessian matrix) can be started via:

```
swoose -m calculate -s water.xyz -y settings.yaml -v -H
```

The command-line output will contain the energy and the gradients that were calculated. The Hessian will be written to the file `hessian.csv`.

A structure optimization may require additional settings for the optimizer, which are decribed in detail in the ReaDuct manual. An example settings file may look like this:

```
mm_parameter_file: final_parameters.dat
covalent_radii_bond_detection: true
results_directory: optimization_results
convergence_max_iterations: 250
bfgs_use_trust_radius: true
bfgs_trust_radius: 1.0
```

This example file also demonstrates that one can allow for a distance-based connectivity detection instead of specifying a connectivity file. The structure optimization can be executed with the following command:

```
swoose -m optimize -s water.xyz -y settings.yaml -v
```

The results (optimized structure and optimization trajectory as XYZ files) can be found in the directory `optimization_results` as specified in the YAML settings file.

Finally, we provide an example of a QM/SFAM calculation. Obviously, the water molecule is not a suitable example for this. Let's assume that our large molecular system, e.g., a protein, is stored in the file `protein.xyz` and the generated parameters are stored in the file `protein_parameters.dat`.

As a first step, one can evaluate a QM region automatically for our system with the following settings file:

```
mm_parameter_file: protein_parameters.dat
covalent_radii_bond_detection: true
ref_data_mode: direct
program: xtb/swoose
qm_model: GFN2
method_family: GFN2/SFAM
method: GFN2
qm_region_center_atoms: [7]
cutting_probability: 0.9
ref_max_size: 300
qm_region_min_size: 80
qm_region_max_size: 100
```

The selected QM region will be between 80 and 100 atoms and the center atom of the QM region was specified as the atom with index 7 (indices start at zero). The command

```
swoose -m select_qm -s protein.xyz -y settings.yaml -v
```

executes the QM region selection. The console output provides you with the resulting QM region in a format which can be directly copied to the settings file of a QM/SFAM calculation:

```
mm_parameter_file: protein_parameters.dat
covalent_radii_bond_detection: true
ref_data_mode: direct
program: xtb/swoose
qm_model: GFN2
method_family: GFN2/SFAM
```

```
method: GFN2
qm_atoms: [3, 18, 19, ..., 89, 97]
```

For the sake of brevity, we did not write out the 80 − 100 atom indices for the `qm_atoms` setting. To perform a calculation, one can employ the "-q" option to signal the quantum–classical hybrid character of the model:

```
swoose -m calculate -q -s protein.xyz -y settings.yaml -v
```

For additional settings that are available for QM/MM structure optimizations or molecular dynamics simulations, check the detailed descriptions of these modes in the previous sections of this manual.

Extensions Planned in the Future

- Export SFAM force field to OpenMM format

References

[1] Brunken, C.; Reiher, M. Self-Parametrizing System-Focused Atomistic Models, *J. Chem. Theory Comput.* **2020,** *16,* 1646–1665.

[2] Bosia, F.; Husch, T.; Müller, C. H.; Polonius, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE Sparrow 5.0.0", `https://doi.org/10.5281/zenodo.3244105`, 2023.

[3] Bannwarth, C.; Caldeweyher, E.; Ehlert, S.; Hansen, A.; Pracht, P.; Seibert, J.; Spicher, S.; Grimme, S. Extended tight-binding quantum chemistry methods, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020,** e01493.

[4] Bensberg, M.; Grimmel, S.; Sobez, J.-G.; Steiner, M.; Unsleber, J. P.; Reiher, M. "SCINE Molassembler 2.0.1", `https://zenodo.org/doi/10.5281/zenodo.4293554`, 2023.

[5] Baiardi, A. *et al.* "SCINE Utilities 9.0.0", `https://zenodo.org/doi/10.5281/zenodo.3828691`, 2023.

[6] Marenich, A. V.; Jerome, S. V.; Cramer, C. J.; Truhlar, D. G. Charge model 5: An extension of Hirshfeld population analysis for the accurate description of molecular interactions in gaseous and condensed phases, *J. Chem. Theory Comput.* **2012,** *8,* 527–541.

[7] Mayer, I. Charge, bond order and valence in the ab initio SCF theory, *Chem. Phys. Lett.* **1983,** *97,* 270–274.

[8] Neese, F. The ORCA program system, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2012,** *2,* 73–78.

[9] Ahlrichs, R.; Bär, M.; Häser, M.; Horn, H.; Kölmel, C. Electronic structure calculations on workstation computers: The program system Turbomole, *Chem. Phys. Lett.* **1989,** *162,* 165–169.

[10] Frisch, M. J. *et al.* "Gaussian 09, revision D. 01", 2009 Gaussian, Inc., Wallingford CT.

[11] MongoDB Inc., "MongoDB 3.2", `www.mongodb.com`, visited on 2024-02-13.

[12] Bensberg, M.; Brunken, C.; Csizi, K.-S.; Grimmel, S. A.; Gugler, S.; Sobez, J.-G.; Steiner, M.; Türtscher, P. L.; Unsleber, J. P.; Vaucher, A. C.; Weymuth, T.; Reiher, M. "SCINE ReaDuct 5.1.0", `https://zenodo.org/doi/10.5281/zenodo.3244107`, 2023.

[13] Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field, *J. Comput. Chem.* **2004,** *25,* 1157–1174.

[14] Senn, H. M.; Thiel, W. QM/MM methods for biomolecular systems, *Angew. Chem. Int. Ed.* **2009,** *48,* 1198–1229.

[15] Lin, H.; Truhlar, D. G. Redistributed charge and dipole schemes for combined quantum mechanical and molecular mechanical calculations, *J. Phys. Chem. A* **2005,** *109,* 3991–4004.

[16] Brunken, C.; Reiher, M. Automated Construction of Quantum–Classical Hybrid Models, *J. Chem. Theory Comput.* **2021,** *17,* 3797–3813.

[17] Goga, N.; Rzepiela, A. J.; de Vries, A. H.; Marrink, S. J.; Berendsen, H. J. C. Efficient Algorithms for Langevin and DPD Dynamics, *J. Chem. Theory Comput.* **2012,** *8,* 3637–3649.

[18] Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* **1984,** *81,* 3684–3690.