MiMiC

Release 0.1.0-rc

Jógvan Magnus Haugaard Olsen, Viacheslav Bolnykh, Simone Me

Apr 21, 2022

INTRODUCTION

Attention: Under construction. Expect major changes as well as missing and incomplete information.

MiMiC is a framework for the development of multiscale models and for running multiscale simulations related to computational chemistry. [1,2] The goal is to enable flexible implementations of multiscale models with support for multiple subsystems that each can be described at different resolutions and levels of theory.

Spatial resolutions may range from a quantum mechanical description of electrons, over atomistic molecular mechanics and coarse-grained modeling, to continuum models. However, MiMiC itself does not calculate any subsystem contributions (e.g., energy and forces) on its own except potentially for contributions arising from the interactions between subsystems. Instead, MiMiC relies on external programs using a multiple-program multiple-data (MPMD) model with loose coupling between programs. This allows high flexibility while at the same time retaining a high degree of computational efficiency by letting the external programs run concurrently and by exploiting the fact that different programs have been optimized for their own specific use case.

Specifically, MiMiC couples a main driver (which runs a molecular dynamics simulation or performs some other type of calculation) to a set of external programs, each of which concurrently computes the contributions that are relevant to a specific subsystem using their own optimal parallelization strategy. Crucial for this strategy is the efficient communication between programs which is achieved through the use of a lightweight communication library (MiMiC Communication Library aka MCL), which was developed specifically for this use. MCL has a C (and optionally Fortran) API that avoids major intervention in the source code of the external programs. The underlying data transfer uses MPI, which gives immediate access to high-speed interconnects such as InfiniBand, but it can be seamlessly expanded, i.e., without altering the API.

For more details about the MiMiC framework and the theory and performance of the QM/MM implementation, we refer to the following two articles:

1. J. M. H. Olsen, V. Bolnykh, S. Meloni, E. Ippoliti, M. P. Bircher, P. Carloni, U. Rothlisberger, *MiMiC: A Novel Framework for Multiscale Modeling in Computational Chemistry*, J. Chem. Theory Comput. **15**, 3810–3823 (2019). DOI: 10.1021/acs.jctc.9b00093

2. V. Bolnykh, J. M. H. Olsen, S. Meloni, M. P. Bircher, E. Ippoliti, P. Carloni, U. Rothlisberger, *Extreme Scalability of DFT-Based QM/MM MD Simulations Using MiMiC*, J. Chem. Theory Comput. **15**, 5601–5613 (2019). DOI: 10.1021/acs.jctc.9b00424

CHAPTER

ONE

ABOUT

Some history and other things about MiMiC…

CHAPTER

TWO

CURRENT FEATURES

MiMiC currently supports using CPMD as the main driver (e.g., for running an MD simulation) and for calculating subsystem contributions it has interfaces CPMD, CFOUR, and GROMACS. Using the currently supported programs it is possible to run electrostatic-embedding QM/MM MD simulations where CPMD or CFOUR can be used as the QM engine and GROMACS as the MM engine.

CHAPTER

THREE

SUPPORTED PLATFORMS

The following lists are compiled based on our own tests and thus may not be exhaustive.

Operating Systems:

- Linux
- MacOS

Compiler suites:

- GCC
- Intel

MPI implementations:

- MPICH
- OpenMPI
- Intel MPI

Todo: Add versions

CHAPTER

FOUR

RELEASE NOTES

There are no releases yet…

CHAPTER

FIVE

KNOWN ISSUES

Todo: To be added…

CHAPTER

SIX

LICENSE

CHAPTER

SEVEN

CITATION

Please cite these papers if you use MiMiC in your work:

1. J. M. H. Olsen, V. Bolnykh, S. Meloni, E. Ippoliti, M. P. Bircher, P. Carloni, U. Rothlisberger, *MiMiC: A Novel Framework for Multiscale Modeling in Computational Chemistry*, J. Chem. Theory Comput. **15**, 3810–3823 (2019). DOI: 10.1021/acs.jctc.9b00093
2. V. Bolnykh, J. M. H. Olsen, S. Meloni, M. P. Bircher, E. Ippoliti, P. Carloni, U. Rothlisberger, *Extreme Scalability of DFT-Based QM/MM MD Simulations Using MiMiC*, J. Chem. Theory Comput. **15**, 5601–5613 (2019). DOI: 10.1021/acs.jctc.9b00424

In addition, in the interest of reproducibility, we ask that you also cite the software as follows (remembering to update the version number and DOI according to the version that was used):

> J. M. H. Olsen, V. Bolnykh, S. Meloni, E. Ippoliti, P. Carloni, and U. Rothlisberger, *MiMiC: A Framework for Multiscale Modeling in Computational Chemistry (v0.1.0)*, GitLab, **2021**. DOI: 10.5281/zenodo.5024023. See https://mimic-project.org/.
>
> V. Bolnykh, J. M. H. Olsen, S. Meloni, E. Ippoliti, P. Carloni, and U. Rothlisberger, *MiMiC Communication Library (v2.0.0)*, GitLab, **2021**. DOI: 10.5281/zenodo.5035085. See https://mimic-project.org/.

For your convenience here are BibTeX entries that you can use

```
@article{mimic-1,
  title = {{MiMiC: A Novel Framework for Multiscale Modeling in Computational Chemistry}}
↪,
  author = {Olsen, J{\'o}gvan Magnus Haugaard and Bolnykh, Viacheslav and
            Meloni, Simone and Ippoliti, Emiliano and Bircher, Martin P. and
            Carloni, Paolo and Rothlisberger, Ursula},
  journal = {J. Chem. Theory Comput.},
  volume = {15},
  number = {6},
  pages = {3810--3823},
  month = {jun},
  year = {2019},
  url = {https://doi.org/10.1021/acs.jctc.9b00093},
  doi = {10.1021/acs.jctc.9b00093}
}

@article{mimic-2,
  title = {{Extreme Scalability of DFT-Based QM/MM MD Simulations Using MiMiC}},
  author = {Bolnykh, Viacheslav and Olsen, J{\'o}gvan Magnus Haugaard and
            Meloni, Simone and Bircher, Martin P. and Ippoliti, Emiliano and
            Carloni, Paolo and Rothlisberger, Ursula},
```

(continues on next page)

(continued from previous page)

```
  journal = {J. Chem. Theory Comput.},
  volume = {15},
  number = {10},
  pages = {5601--5613},
  month = {oct},
  year = {2019},
  url = {https://doi.org/10.1021/acs.jctc.9b00424},
  doi = {10.1021/acs.jctc.9b00424}
}

@misc{mimic:0.1.0,
  author = {Olsen, J{\'o}gvan Magnus Haugaard and Bolnykh, Viacheslav and
            Meloni, Simone and Ippoliti, Emiliano and Carloni, Paolo and
            Rothlisberger, Ursula},
  title = {MiMiC: A Framework for Multiscale Modeling in Computational Chemistry (v0.1.
↪0)},
  publisher = {GitLab},
  year = {2021},
  doi = {10.5281/zenodo.5024023},
  note = {See https://mimic-project.org/}
}

@misc{mcl:2.0.0,
  author = {Bolnykh, Viacheslav and Olsen, J{\'o}gvan Magnus Haugaard and
            Meloni, Simone and Ippoliti, Emiliano and Carloni, Paolo and
            Rothlisberger, Ursula},
  title = {MiMiC Communication Library (v2.0.0)},
  publisher = {GitLab},
  year = {2021},
  doi = {10.5281/zenodo.5024023},
  note = {See https://mimic-project.org/}
}
```

CHAPTER

EIGHT

REQUIREMENTS

8.1 MiMiC Framework

8.1.1 Minimum requirements

- CMake 3.12+
- Fortran compiler that supports the Fortran 2008 standard
- MiMiC Communication Library 2.0+

8.1.2 Optional dependencies

- Git (to clone the source repository)
- Fortran compiler that supports the OpenMP 3.0 standard (to enable threading)

The following optional dependencies are only needed to build the test suite:

- C++ compiler that supports the C++14 standard
- MPI library that supports the MPI 2.1 standard
- pFUnit 4.0+

8.2 MiMiC Communication Library

8.2.1 Minimum requirements

- CMake 3.12+
- C++ compiler that supports the C++14 standard
- MPI library that supports the MPI 2.1 standard

8.2.2 Optional dependencies

- Git (to clone the source repository)
- Fortran compiler that supports the Fortran 2008 standard (to build the Fortran API)

The following optional dependencies are only needed to build the test suite:

- GoogleTest (can optionally be downloaded at build time)

8.3 Supported external programs

- GROMACS 2022 series or newer (not released)
- CPMD 4.5 or higher (not released)

It is also possible to patch CPMD 4.3 and GROMACS 2021. See *Getting the Sources* for information on how to obtain the patches and *Build and Install* for instructions on how to apply the patches.

CHAPTER

NINE

GETTING THE SOURCES

9.1 MiMiC Framework

The MiMiC framework is hosted on GitLab (https://gitlab.com/MiMiC-projects/MiMiC). The released versions can be downloaded at https://gitlab.com/MiMiC-projects/MiMiC/-/releases. From the command line, a specific version can be downloaded using wget:

```
$ wget https://gitlab.com/MiMiC-projects/MiMiC/-/archive/0.1.0/MiMiC-0.1.0.tar.gz
```

Alternatively, the entire source (incl. development versions) can be obtained by cloning the git repository (developers may want to create a fork on GitLab and clone that instead):

```
$ git clone https://gitlab.com/MiMiC-projects/MiMiC.git
```

This will create a directory named `MiMiC`. Entering that directory will put you directly in the main development branch. From here, released version can be checked out by:

```
$ git checkout 0.1.0
```

where you can replace the tag with the version that you want. A released version can also be checked out directly by supplying the release tag when cloning:

```
$ git clone -b 0.1.0 https://gitlab.com/MiMiC-projects/MiMiC.git
```

Warning: Note that the development branch may be unstable or contain breaking changes that are not compatible with released versions of supported programs. Therefore we do not recommend that you use it unless you really know what you are doing.

9.2 MiMiC Communication Library

The MiMiC communication library is also hosted on GitLab (https://gitlab.com/MiMiC-projects/CommLib). The same procedure as for MiMiC can be followed by just replacing the URLs. In short, the released versions can be downloaded at https://gitlab.com/MiMiC-projects/CommLib/-/releases:

```
$ wget https://gitlab.com/MiMiC-projects/CommLib/-/archive/2.0.0/CommLib-2.0.0.tar.gz
```

while the repository can be cloned and a released version checked out as follows:

```
$ git clone https://gitlab.com/MiMiC-projects/CommLib.git
$ git checkout 2.0.0
```

Warning: Note that the development branch may be unstable or contain breaking changes that are not compatible with released versions of supported programs. Therefore we do not recommend that you use it unless you really know what you are doing.

9.3 External programs

9.3.1 CPMD

The CPMD program itself can be obtained from https://www.cpmd.org/ and we refer to their documentation for further instructions. Official support for MiMiC is only available from CPMD 4.5 and higher. However, patches for CPMD 4.1 and 4.3 can be downloaded at https://mimic-project.org/cpmd-patches. Instructions on how to apply the patches are given below in *Build and Install*.

Todo: Add CPMD patches

9.3.2 GROMACS

GROMACS can be obtained from https://www.gromacs.org/ and we refer to their documentation for further instructions. Official support for MiMiC is only available from GROMACS 2022 series and newer. However, patches for the GROMACS 2021 series can be downloaded at https://mimic-project.org/gromacs-patches. Instructions on how to apply the patches are given below in *Build and Install*.

Todo: Add GROMACS patches

CHAPTER

TEN

BUILD AND INSTALL

The MiMiC framework and communication library are configured, built, tested, and installed using the CMake build system. The following instructions will guide you through the process from download to installation. The *Quickstart guide* will get you through the process without providing details and may not work for everyone. On the other hand, the *Walkthrough installation* will go into more detail and describe each step more carefully.

10.1 Quickstart guide

The complete process from download to install using typical settings of both the MiMiC framework and communication library is as follows:

Note: Take care to check that each step completes successfully before moving on to the next step. In particular, ensure that **all** tests pass.

```
$ wget https://gitlab.com/MiMiC-projects/CommLib/-/archive/2.0.0/CommLib-2.0.0.tar.gz
$ tar -zxvf CommLib-2.0.0.tar.gz
$ cd CommLib-2.0.0
$ mkdir build
$ cd build
$ cmake -DBUILD_FORTRAN_API=YES -DBUILD_TESTS=YES -DDOWNLOAD_GTEST=YES -DCMAKE_INSTALL_
↪PREFIX=${HOME}/.local ..
$ make
$ make test
$ make install
$ cd ../../
$ wget https://gitlab.com/MiMiC-projects/MiMiC/-/archive/0.1.0/MiMiC-0.1.0.tar.gz
$ tar -zxvf MiMiC-0.1.0.tar.gz
$ cd MiMiC-0.1.0
$ mkdir build
$ cd build
$ cmake -DENABLE_OPENMP=YES -DBUILD_TESTS=YES -DCMAKE_INSTALL_PREFIX=${HOME}/.local ..
$ make
$ make test
$ make install
$ cd ../../
```

This will install everything in your user's home directory under `.local` using standard GNU locations (i.e., `lib` or `lib64`, `include`, etc.). If everything completed successfully then it is safe to delete the CommLib-2.0.0 and MiMiC-

0.1.0 directories (as well as the downloaded source packages). The next step is to enable MiMiC in the external programs that you want to use (see *Building external programs with MiMiC support*).

10.2 Walkthrough installation

It is necessary to build and install the communication library first as the MiMiC framework depends on it. Therefore we start by downloading and unpacking the communication library.

Then we change directory to the root of the communication library and from there create a build directory, enter it, configure, build, and install.

The MiMiC communication library is configured, built, and installed using CMake which in most cases can be used without alot of manual configuration. Apart from the standard CMake options, there are also the following list of options that are specific to CommLib:

- `BUILD_FORTRAN_API` - Build the Fortran API (required by MiMiC)
- `INCLUDE_TESTS` - Build the test suite
- `DOWNLOAD_GTEST` - Download GoogleTest (needed to run the test suite)

The quick and simple procedure is as follows (more detailed instructions are provided below):

```
$ mkdir <build-directory>
$ cd <build-directory>
$ cmake -DBUILD_FORTRAN_API=YES -DCMAKE_INSTALL_PREFIX=<install-path-prefix> <path-to-
↪commlib>
$ make
$ make install
```

where `<build-directory>` is the directory in which the code will be built (which is safe to delete after successful completion), `<install-path-prefix>` is the path prefix in which the compiled files will be installed in standard directories (i.e., `bin`, `include`, `lib` (or `lib64`), etc.), and `<path-to-commlib>` is the path to the root of the CommLib source code.

A typical procedure Start the process by creating a build directory and entering it:

```
$ mkdir build
$ cd build
```

From there you first configure the build using CMake:

```
$ cmake -DBUILD_FORTRAN_API=YES -DCMAKE_INSTALL_PREFIX=${HOME}/.local ..
```

Here we have chosen to build the Fortran API because it is needed by MiMiC and also specified an install path using CMake's standard `CMAKE_INSTALL_PREFIX` option. The supplied install path is a prefix and the files will be installed in standard locations below the given prefix, i.e., bin, include, lib/lib64, etc. Note that the default install path (usually `/usr/local`) will often require superuser privileges. By default only static libraries are built and to instead build shared libraries you need to add `-DBUILD_SHARED_LIBS=YES`. It is also possible to build the code completely outside of the source directory in which case you replace `..` with the path to the root of the source directory. In case CMake did not detect the compiler that you want to use then you can manually select it by adding `-DCMAKE_CXX_COMPILER=<c++-compiler>` (and `-DCMAKE_Fortran_COMPILER=<fortran-compiler>` if you chose to build the Fortran API) or by setting the environment variable `CXX` (and `FC` for the Fortran compiler).

Note: Examine the output from CMake to ensure that it completes without errors before moving on.

Once the build is configured it can be compiled and installed as follows:

```
$ make
$ make install
```

This will also install CMake package files allowing you to easily include CommLib in a CMake configuration.

These instructions assume that you are in the root folder of the MiMiC source. From here you should first create a build drectory and enter it:

Todo: This is how you build and install MiMiC

10.3 Building external programs with MiMiC support

The following instructions will focus on how to enable MiMiC in supported programs and as such will not be thorough guides on the build and install process of each program. For that we refer to the instructions given by the distributors of the programs.

10.3.1 Building CPMD with MiMiC support

Todo: Expand with patch and simple build instructions

10.3.2 Building GROMACS with MiMiC support

Todo: Expand with patch and simple build instructions

CHAPTER

ELEVEN

CPMD INPUT

CHAPTER

TWELVE

GROMACS INPUT

CHAPTER

THIRTEEN

RUNNING QM/MM-MD SIMULATIONS

13.1 Hybrid MPI/OpenMP parallelization

CHAPTER

FOURTEEN

HOW TO GET HELP

MiMiC is distributed for free and without any guarantee of reliability, accuracy, or suitability for any particular purpose. No obligation to provide technical support is expressed or implied.

14.1 Where should you post?

- **How do I . . . ?** Ask in the comments section below
- **I get this error, why?** Ask in the comments section below
- **I get this error and I am sure it is a bug.** Please file an issue on GitLab
- **I have an idea/request.** Ask in the comments section below or file an issue on GitLab
- **I have an idea/request and a plan for its implementation.** Please file an issue on GitLab
- **Why do you . . . ?** Ask in the comments section below
- **When will you . . . ?** Ask in the comments section below
- **Can I open a discussion on this part of the code?** Please file an issue on GitLab

14.2 Questions and comments

CHAPTER

FIFTEEN

RUNNING A QM/MM-MD SIMULATION USING CPMD AND GROMACS

15.1 Setting up input for CPMD

15.2 Setting up input for GROMACS

CHAPTER

SIXTEEN

CONTRIBUTION GUIDE

Todo: To be added…

CHAPTER

SEVENTEEN

DEVELOPER GUIDE

Todo: To be added…

17.1 Code documentation

Automatically generated code documentation (using FORD) is available at https://autodoc.mimic-project.org/.

CHAPTER

EIGHTEEN

REFERENCES

CHAPTER

NINETEEN

LIST OF TODOS

Todo: To be added...

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/contributing.rst, line 4.)

Todo: To be added...

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/contributing.rst, line 10.)

Todo: Add CPMD patches

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/installation.rst, line 130.)

Todo: Add GROMACS patches

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/installation.rst, line 143.)

Todo: This is how you build and install MiMiC

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/installation.rst, line 273.)

Todo: Expand with patch and simple build instructions

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/installation.rst, line 291.)

Todo: Expand with patch and simple build instructions

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/installation.rst, line 299.)

Todo: Add versions

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/introduction.rst, line 40.)

Todo: To be added...

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/mimic-project/checkouts/latest/docs/introduction.rst, line 53.)

BIBLIOGRAPHY

[1] Jógvan Magnus Haugaard Olsen, Viacheslav Bolnykh, Simone Meloni, Emiliano Ippoliti, Martin P Bircher, Paolo Carloni, and Ursula Rothlisberger. MiMiC: A Novel Framework for Multiscale Modeling in Computational Chemistry. *J. Chem. Theory Comput.*, 15:3810–3823, 2019. doi:10.1021/acs.jctc.9b00093.

[2] Viacheslav Bolnykh, Jógvan Magnus Haugaard Olsen, Simone Meloni, Martin P Bircher, Emiliano Ippoliti, Paolo Carloni, and Ursula Rothlisberger. Extreme Scalability of DFT-Based QM/MM MD Simulations Using MiMiC. *J. Chem. Theory Comput.*, 15:5601–5613, 2019. doi:10.1021/acs.jctc.9b00424.