

Installation Guide for MEDYAN v4.0

Papoian Lab, University of Maryland

Contents

1	Unpacking MEDYAN	2
2	Setting up the Makefile	2
2.1	Compilers and libraries needed	2
2.2	Editing the Makefile	2
2.2.1	Compiler and library choices	2
2.2.2	Optimization flags	3
2.3	Command line compilation macros	3
2.4	Dependency file	3
2.5	Compilation	3
3	Running the MEDYAN executable	3

1 Unpacking MEDYAN

To unpack the MEDYAN tar file, run the following command in your terminal shell:

```
> tar -xvf MEDYAN.tar -C <InstallDirectory>
```

Once this is complete, all source code and other files will be in the chosen directory.

2 Setting up the Makefile

The Makefile for compilation of MEDYAN will be in `InstallDirectory/MEDYAN`, along with all source code that is needed for compilation.

2.1 Compilers and libraries needed

MEDYAN is a C++ program that can be compiled with the following C++11 compilers:

- GCC 4.7 and above (Full C++11 support)
- Clang 3.3 and above (Also full C++11 support, default Apple compiler)

Compiling with incomplete C++11 compatibility may result in compilation errors. MEDYAN uses the following math and utility libraries:

- Boost libraries 1.49 or above
- GSL library
- UME¹ (patched version of UMS::SIMD. Included in the external directory).

2.2 Editing the Makefile

2.2.1 Compiler and library choices

The Makefile can be edited to include a compiler or library in a non-default directory by changing the `CXX`, `CPPFLAGS`, and `LDLIBS` variables within the Makefile. To modify the default directory:

- Change the g++ directory `CXX = /usr/bin/g++ -std=c++11` to `CXX = /your g++ directory/g++ -std=c++11` or `CXX = g++ -std=c++11`.
You may need to load g++ module manually.

¹Valid only if using SIMD based pair-wise distance search protocol on an Intel chip set machine. Please refer to command line compilation macros for more details.

- Change library directory `LDLIBS = -L/usr/local/lib/ -lboost_system -lgsl` to `LDLIBS =-L/usr/local/lib/ -lboost_system -L/your boost library directory/lib/`. For some reasons, `-lgsl` may cause errors when it is by default included and removing this command does not affect installation.
- Change the third line in `CPPFLAGS`, which by default contains `-I/usr/local/include -I/usr/include/boost`, to include `-I/<installed boost directory>/include` and `-I/<installed boost directory>/include/boost`

2.2.2 Optimization flags

The code can be compiled with either `DEBUG` flags, which specifies the default debugging flags for compatibility with GDB and other debugger tools. For optimal performance, compile with the `FAST` flag, which gives a number of optimization flags. This can be edited for the system specifications.

2.3 Command line compilation macros

The command line macros can be edited in the Makefile to turn on or off certain code capabilities. See the Usage guide for more details on these macros and their implications. *While these macros are customizable, we highly recommend the default usage as previously defined in the original Makefile for full capability.* The macros available for user editing are:

2.4 Dependency file

An optional dependency file can be generated by running the command `make Makefile.dep`. This command will automatically be performed when the typical make function is executed.

2.5 Compilation

The code can be compiled into an executable file `MEDYAN` by running `make` at the command line. `make clean` will erase all object files as well as the executable in the local directory.

3 Running the MEDYAN executable

To run the executable, put the following command into the terminal shell:

```
> ./MEDYAN -s <SystemFile> -i <InputDirectory> -o <OutputDirectory>
```

More details on the system input file and directories can be found in the Usage guide.

Macro	Description
CHEMISTRY	Enable system chemistry.
Enable both MECHANICS SERIAL	Enable system mechanics. Serial implementation of conjugate gradient
Pair-wise distance protocols NLORIGINAL HYBRID_NLSTENCILLIST SIMDBINDINGSEARCH	Choose one of the macros Optimized protocol similar to MEDYAN3.2 New optimized SIMD-free protocol New SIMD based search protocol
DYNAMICRATES	Enable dynamic rate changing. This macro can only be specified if both CHEMISTRY and MECHANICS are enabled.
BOOST_MEM_POOL	Enable boost memory pool optimizations.
BOOL_POOL_NSIZ	Set boost memory pool size. Default value is 65536.
TRACK_DEPENDENTS	Track reaction dependents in system.
TRACK_ZERO_COPY_N	For activation of reactions based on copy number.
TRACK_UPPER_COPY_N	For activation of reactions based on copy number.
REACTION_SIGNALING	Enable reaction callback signaling.
RSPECIES_SIGNALING	Enable species callback signaling.
DCHECK- FORCES_INF_NAN	Checks if forces are NaN or Inf. Exists if forces become unreasonable.
PLOSFEDBACK	Uses feedback models described in Plos 2016 paper.