

# UNRES.3.2.1 – INSTALLATION GUIDE

Dawid Jagieła, Adam Liwo  
Laboratory of Molecular Modeling  
Faculty of Chemistry  
University of Gdansk  
Wita Stwosza 63  
80-308 Gdansk, Poland

December 11, 2014

# Contents

<b>1</b>	<b>Putting the package on your system</b>	<b>3</b>
<b>2</b>	<b>Installation options</b>	<b>3</b>
2.1	Requirements . . . . .	3
<b>3</b>	<b>Installation using Cmake</b>	<b>3</b>
3.1	Requirements . . . . .	3
3.2	Basic installation . . . . .	5
3.3	Advanced installation . . . . .	5
<b>4</b>	<b>Step-by-step installation</b>	<b>6</b>
<b>5</b>	<b>TECHNICAL SUPPORT CONTACT INFORMATION</b>	<b>7</b>

# 1 Putting the package on your system

The distribution is contained in the `unrespack-v.3.2.1.tar.gz` file. To put the package on your system, copy the archive to your UNRES directory (you might want to create an environmental variable, `$UNRESROOT` or so, to define the location of UNRES on your system) and say:

```
gzip -cd unrespack-v.3.2.1.tar.gz | tar xf -
```

This will produce the directory structure shown in Figure 1.

## 2 Installation options

The most convenient way to install the package is using Cmake, as described in section 2. If your system does not run the required version of Cmake or installation does not work, do step-by-step compilation of the components of the package, as described in section 3.

### 2.1 Requirements

- C compiler
- Fortran compiler (must understand the Fortran 77 instructions)
- MPI (for CSA and MREMD)

## 3 Installation using Cmake

### 3.1 Requirements

- CMake 2.8.0 or later
- C compiler
- Fortran compiler
- MPI (for CSA and MREMD)

\$UNRESROOT

```
|
|-----doc    (documentation)
|
|-----PARAM (force field parameters)
|
|-----source
|         |
|         |-----unres (UNRES source codes; various versions)
|         |         |
|         |         |---src_MIN (only energy evaluation and minimization)
|         |         |---src_CSA (all functions except MD, includes CSA)
|         |         |---src_MD  (all functions except CSA, includes MD, single chains)
|         |         |---src_MD-M (all functions except CSA, includes MD, oligomeric proteins)
|         |-----wham (weighted analysis method source codes)
|         |         |
|         |         |---src (single chains)
|         |         |---src-M (oligomeric proteins)
|         |
|         |-----cluster (cluster analysis source coded)
|         |         |
|         |         |---clust-unres
|         |         |         |
|         |         |         |----src    (input data from UNRES)
|         |         |         |
|         |         |         |---clust-wham    (input data from WHAM)
|         |         |         |
|         |         |         |----src    (for single-chain proteins)
|         |         |         |----src-M  (for oligomeric proteins)
|         |         |
|         |-----xdrfpdb (file format conversion source codes)
|         |         |
|         |         |---src    (single chains)
|         |         |---src-M  (oligomers)
|
|-----bin (C-shell script, batch scripts, and pre-compiled binaries)
|         |
|         |-----unres
|         |         |
|         |         |---CSA
|         |         |---MD
|         |         |---MINIM
|         |-----wham
|         |-----cluster
|         |-----xdrfpdb
|
|-----examples
|         |
|         |-----unres
|         |-----wham
|         |-----cluster
```

Figure 1: Directory structure of the unres-3.2.1 package

## 3.2 Basic installation

These instructions give a very basic overview of how to configure, compile and install UNRESPACK on most systems. If you are using unique install locations and/or libraries that are not automatically detected please consult the 'Advanced' section.

1. Create a 'build' directory in the package source directory.

```
mkdir build
cd build
```

2. Configure the build system

```
cmake ..
```

3. Compile

```
make
```

4. Install

```
sudo make install
```

or

```
make install
```

## 3.3 Advanced installation

The build system (CMake) provides mechanisms for specifying non-standard build parameters.

1. Compilers & installation

```
-DCMAKE_Fortran_COMPILER=xxx equal to name of Fortran Compiler you wish to use
(ifort, gfortran)
```

```
-DCMAKE_INSTALL_PREFIX=xxx specify the binaries installation prefix
(default UNRESPACK_source_dir/bin)
```

2. Force fields

`-DUNRES_MD_FF=xxx` compiles the MD versions with given force field.  
Options are: GAB, EOLL2Y. Default: GAB

`-DUNRES_CSA_FF=xxx` compiles the CSA versions with given force field  
Options are: CASP3, ALPHA, BETA, ALPHABETA, CASP5, 3P, 4P. Default: 4P.

Please read the online documentation on force fields available at <http://unres.eu/unres#SECTION00090000000000000000>

### 3. MPI

MPI implementation on your system should be automatically detected ("MPI Found" message after running `cmake`). If not you have two options:

- (a) Try setting the path to your MPI wrapper implementation

```
-DMPI_Fortran_COMPILER=xxx MPI wrapper
```

- (b) If option 1 fails or your MPI implementation does not come with a compiler wrapper try setting both the MPI include and library paths manually. This will circumvent autodetection entirely.

```
-DMPI_Fortran_INCLUDE_PATH=xxx  
-DMPI_Fortran_LIBRARY="xxx"
```

## 4 Step-by-step installation

For this installation, you will need to visit each source directory (see `doc/UNRESPACK.txt` for directory structure). Specific installation instructions are in the documentation of the particular components of the package (UNRES, WHAM, CLUSTER, XDRFPDB). Only general instructions are given here.

1. Go to the respective source directory.
2. Determine if any of the Makefiles present there matches your needs. The Makefiles for Intel Fortran and Gnu Fortran are present everywhere and are guaranteed to work (provided that your compiler/MPI installation is correct). Use this Makefile as the working Makefile  
If your system uses a different compiler, copy the most matching Makefile to your working Makefile (e.g., to `Makefile_CRAY` if you'll be working with Cray Fortran).
3. Make a symbolic link of the working Makefile to `Makefile`, e.g.,

```
ln -s Makefile_gfortran Makefile
```

Before that, you'll need to remove the existing symbolic link (in the distribution, it points to `Makefile_ifort`).

4. Edit the Makefile to define MPI location, libraries, and the binaries directory and executable names, if you want to use different location/names from those in the distribution Makefiles. The present locations are the subdirectories of the bin directory and executable names include package component, compiler, MPI information, and force field, e.g. `unres_csa_gfort_MPICH_4P.exe` stands for the CSA component, compiled with GNU Fortran in a parallel mode (using MPICH) to run calculations with the 4P force field.
5. For the components of XDRFPDB and in the directory `cluster/unres/src`, say

```
make
```

In other directories say

```
make <force_field>
```

to create the respective executable.

The compiler flags are specified for the 4P (also covers the 3P, and CASP5 ff), GAB (also covers E0G), and E0LL2Y force fields for all components; the CASP3 force field is also specified for the CSA and MINIM component.

## 5 TECHNICAL SUPPORT CONTACT INFORMATION

Dawid Jagieła  
Faculty of Chemistry, University of Gdansk  
ul. Wita Stwosza 63, 80-308 Gdansk Poland.  
fax: +48 58 523 5012  
e-mail: [lightnir@gmail.com](mailto:lightnir@gmail.com)

Prepared by Dawid Jagieła and Adam Liwo, 4/12/2014