i3Drefine

Version 1.0

March 2013

Debswapna Bhattacharya and Jianlin Cheng

Department of Computer Science
University of Missouri
Columbia, MO 65211, USA
Email: db279@mail.missouri.edu, chengji@missuri.edu

Contents

1 What is i3Drefine?

1.1 Introduction

i3Drefine is software for consistent and computationally efficient protein structure refinement. The goal of i3Drefine is to perform consistent and simultaneous improvement in both global and local structural qualities of the initial models to bring it closer to the native state in a computationally inexpensive manner. The name i3Drefine stands for **i**terative three-dimensional (**3D**) protein structure **refine**ment.

1.2 i3Drefine Algorithm

The i3Drefine protocol employs an iterative refinement based on two steps of process. The first step is based on optimization of hydrogen bonding (HB) network and the second step applies atomic-level energy minimization on the optimized model using a composite physics and knowledge-based force fields.

1.3 i3Drefine Force Field

The core of i3Drefine use a composite physics and knowledge-based energy function [1]. The physics based part involves the bonded interaction potential and a tethering term while the atomic pairwise potential of mean force and explicit hydrogen bonding potential constitute the knowledge-based part (Figure 1).

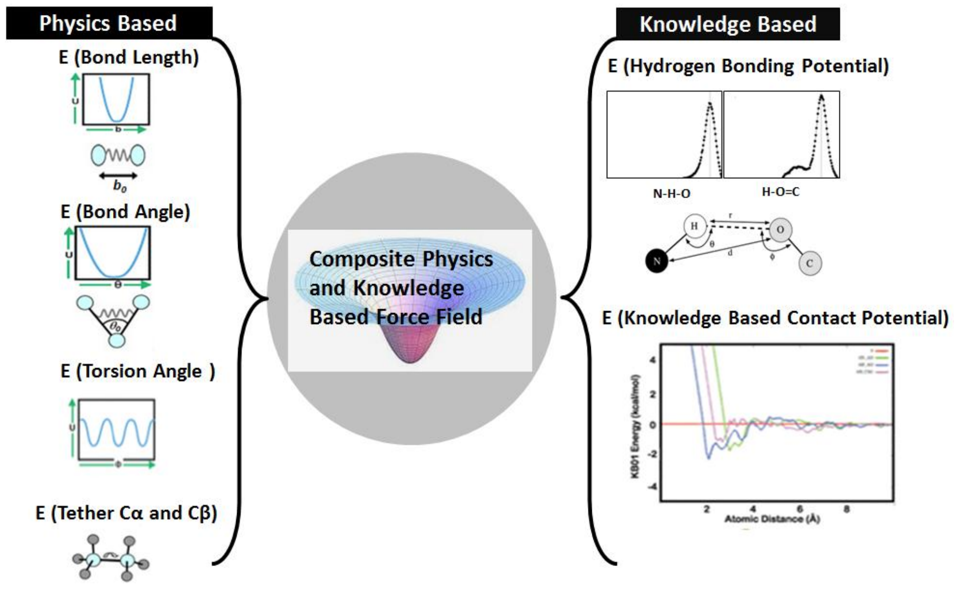


Figure 1. The composite physics-based and knowledge-based force field

2 Installation, license and other practicalities

2.1 Implementation and Platform

The core of i3Drefine is developed in Java on top of MESHI [2] software package and the command-line interface to perform the refinement is developed in Perl. i3Drefine also require DSSP [3] program in order to calculate the secondary structure of the starting structure. The software has been tested on 64-bit Intel Linux platform.

2.2 Obtaining i3Drefine

Executable version of i3Drefine software and its documentation can be downloaded from http://protein.rnet.missouri.edu/i3drefine as a gzipped tar file. Simply unzip and untar it in a suitable directory.

2.3 Prerequisites and External Programs

The required programs/libraries are:

- Java (version 6.0 or later) http://www.java.com/en
- Perl (version 5.8.8 or later) http://www.perl.org
- DSSP software http://swift.cmbi.ru.nl/gv/dssp

2.4 Installation

Installing i3Drefine requires the following steps:

- **Step 1**: Download i3Drefine.tar.gz under a suitable directory. Let's call it /home.

- **Step 2**: Unpack the compressed tar file:
```
$ gunzip  i3Drefine.tar.gz
$ tar xvf i3Drefine.tar
```

This will create a new directory called i3Drefine which contains the following files and directories:

```
bin/
script/
software/
test/
configure.pl
README
```

About the directories, subdirectories and files:
bin is an empty directory which will contain the shell script to execute the i3Drefine program once it is installed, script directory contains the perl interface for executing the refinement, software directory contains two main subdirectories: 3Drefine and one empty directory DSSP. Do not alter the location or rename these directories for a seamless i3Drefine execution. test is a directory containing a sample pdb file for testing the software. The script configure.pl is the configuration script while README includes a brief installation instruction.

- **Step 3**: Download Java version 6.0 or later from http://www.java.com/en and place it in a relevant directory. Let's assume the Java root directory is /home /jdk1.6.0_01.

- **Step 4**: Set the PATH and CLASSPATH environment variables as follows:

```
$ export PATH=$PATH:/home/jdk1.7.0_01/bin
$export CLASSPATH=$CLASSPATH:/home/i3Drefine/software/3Drefine
$ export CLASSPATH=$CLASSPATH:/home/3Drefine/software/3Drefine/programs
```

These lines can be added to .bashrc file:

- **Step 5**: Download the DSSP software from http://swift.cmbi.ru.nl/gv/dssp and place the executable version of the software inside the empty directory i3Drefine/software/DSSP/. Rename the executable to 'dssp' and don't forget the change the permission to executable mode (chmod 755).

- **Step6**: Go to the root of i3Drefine (where the script configure.pl exists). Make sure the script configure.pl is in executable mode and simply type:

  ```
  $ ./configure.pl
  ```

- **Step 7**: To verify that everything went well during the installation, check the following two things:

  - The bin directory contains a shell script named i3Drefine.sh and it is in executable mode.
  - Inside software/3Drefine directory, a new file is created named Commands. Open the file and check whether the first line contains full path to software/3Drefine/parameters.

This concludes a successful installation of i3Drefine software.

2.5 Copyright Notice

The executable software i3Drefine is distributed free of charge as it is to any non-commercial users. The authors hold no liabilities to the performance of the program.

3 Using i3Drefine

Here we describe typical way to execute i3Drefine software, input and output format and the typical execution time needed for a job and to interpret the results.

3.1 Preliminaries

Since the execution of i3Drefine needs the secondary structure assignment by DSSP, a primary condition for a successful refinement is the input structure must contain all necessary atoms in order for DSSP to correctly compute the secondary structure. For this reason, i3Drefine should not be used on a reduced model representation (e.g. missing side chain atoms or hydrogen atoms). In case the initial model is not in full-atomic representation, we strongly recommend taking the necessary steps to convert the model into all-atom structure. Users may take help of some relevant tools or programs for building side chain atoms or addition of hydrogen atoms.
Also, i3Drefine recognizes the twenty standard amino acids only. In case the starting structure contains some non-standard amino acids, it is likely that the refinement would fail. In cases like this, one may discard the residues containing non-standard amino acids before submitting the model for refinement.

3.2 Input Format

i3Drefine requires only two parameters: the starting structure in pdb format and the number of refined models one wish to generate from the iterative refinement framework. Although there are no restrictions to the number of refined models one can generate, we recommend to this number within 10 so that the energy minimization does not overshoot the global minima in the multidimensional energy landscape.

3.3 Output Format

While executing, i3Drefine automatically creates a workspace in the directory from which the program is called. This is associated with the auto-generated Job-Id of the submitted job. At the end of successful execution, RESULT/ directory created under the workspace contains the refined models numbered as REFINED_n.pdb; where n is the iteration number.

3.4 Execution Time

i3Drefine software is computationally efficient consuming only few minutes (typically less than 15 mints) in order to produce five refined models. The execution time, however, is directly proportional to the length of the starting structure and a smaller protein takes short time than a larger protein.

4 Error Handling

This section provides details on how to debug the job in case any job fails to execute by inspecting the log files generated.

4.1 Execution Log File and Log Directory

i3Drefine automatically creates one execution log file capturing activities for all iterations while a refinement job is in progress. This log file serves as a pointer to go to the detailed log file generated by DSSP and the robust error handling utility of the main energy minimization program. These detailed log files can be located under LOG/ directory created under the workspace. The execution log automatically provides the path for the detailed log files for users to inspect the possible cause of a failure. If a i3Drefine job completes successfully, a summary of the job is displayed at the end of the log file showing the name and path of the generated refined models and time taken to finish each iteration.

4.2 Debugging a Failure

Although there could be potentially many causes, failures are primarily caused by two broad classes of errors: (1) errors in executing DSSP and (2) errors in refinement. For the first cause, users are requested to check the starting structure carefully and make sure that the model is in all-atom representation (Please check section 3.1) to ensure a successful execution of DSSP program,.
For the second case, users are requested to go to the refinement log file and check for the line "*****Criminal found:**" The ATOM line displayed below this line could potentially cause a problem in refinement. The problem could be of duplicate alpha-carbon atoms (CA) present for the same residue or when some abnormality in the physicality of stating structure is observed. In case the log file contains a line with "**Minimization Error:**", users are requested to resubmit the job.

4.3 Bug Reporting

If you found some bugs or in case the failure of your job seems obscure (except for the cases mentioned in section 4.2), you can contact us at db279@mail.missouri.edu.
If you wish to report a bug, the subject line of your email should be: "BUG in i3Drefine". In case to seek our help to understand the reason for a failure, the subject line should be: "FAILURE in i3Drefine. All other emails would generally be ignored.
Please note that for bugs to be fixed or failure to be debugged, they must be reproducible. This generally means that you will need to attach all necessary input and log file(s) and screenshot(s) of the problem with your email.

5 Example

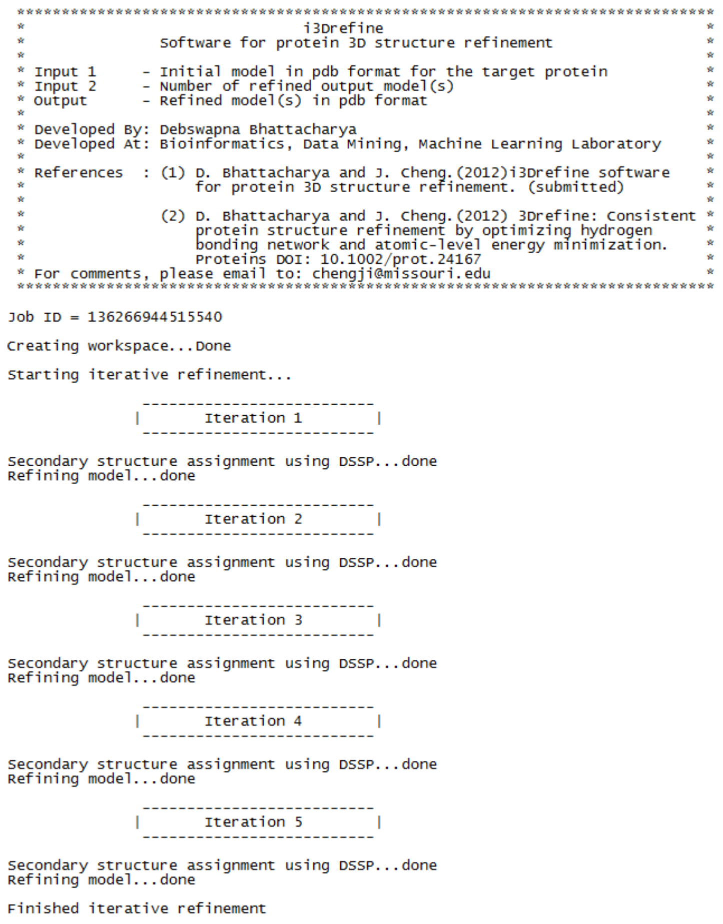
This example shows how a starting structure can be refined using i3Drefine and gives the snapshot of the log file generated.

- **Step 1**: Go to the test/ directory. An example pdb file should exist with the name start_model.pdb. Let's say we wish to generate 5 refined models for this structure.

- **Step 2**: From within the test/ directory execute the following command:

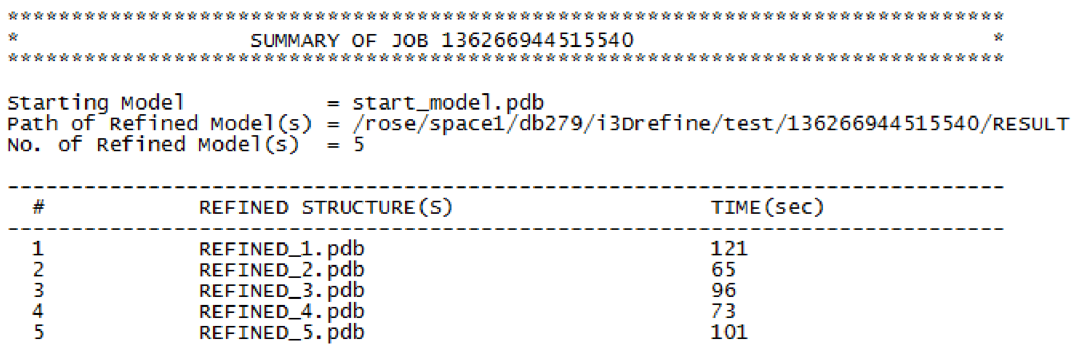
  ```
  $ ../bin/i3Drefine.sh start_model.pdb 5
  ```

- **Step 3**: As soon as the job is submitted, the auto-generated Job-Id (e.g. 136266944515540) is assigned to the job which is displayed in the execution log file displayed at the stdout. A directory with the name of the Job-Id has been created under the test/ directory will be created containing two subdirectories: LOG/ and RESULT/. As long as the execution log file does not throw any errors, users should not bother to look at the detailed log file. Following is the snapshot of the first part of execution log file:



```
*******************************************************************************
*                                  i3Drefine                                  *
*                  Software for protein 3D structure refinement              *
*                                                                             *
* Input 1      - Initial model in pdb format for the target protein          *
* Input 2      - Number of refined output model(s)                           *
* Output       - Refined model(s) in pdb format                              *
*                                                                             *
* Developed By: Debswapna Bhattacharya                                        *
* Developed At: Bioinformatics, Data Mining, Machine Learning Laboratory     *
*                                                                             *
* References   : (1) D. Bhattacharya and J. Cheng.(2012)i3Drefine software   *
*                    for protein 3D structure refinement. (submitted)        *
*                                                                             *
*                (2) D. Bhattacharya and J. Cheng.(2012) 3Drefine: Consistent *
*                    protein structure refinement by optimizing hydrogen     *
*                    bonding network and atomic-level energy minimization.   *
*                    Proteins DOI: 10.1002/prot.24167                         *
* For comments, please email to: chengji@missouri.edu                        *
*******************************************************************************

Job ID = 136266944515540

Creating workspace...Done

Starting iterative refinement...

               --------------------------
              |       Iteration 1        |
               --------------------------

Secondary structure assignment using DSSP...done
Refining model...done

               --------------------------
              |       Iteration 2        |
               --------------------------

Secondary structure assignment using DSSP...done
Refining model...done

               --------------------------
              |       Iteration 3        |
               --------------------------

Secondary structure assignment using DSSP...done
Refining model...done

               --------------------------
              |       Iteration 4        |
               --------------------------

Secondary structure assignment using DSSP...done
Refining model...done

               --------------------------
              |       Iteration 5        |
               --------------------------

Secondary structure assignment using DSSP...done
Refining model...done

Finished iterative refinement
```

The second part of execution log file providing job summary is shown below:



```
******************************************************************************
*                    SUMMARY OF JOB 136266944515540                          *
******************************************************************************

Starting Model          = start_model.pdb
Path of Refined Model(s) = /rose/space1/db279/i3Drefine/test/136266944515540/RESULT
No. of Refined Model(s)  = 5

------------------------------------------------------------------------------
  #            REFINED STRUCTURE(S)                        TIME(sec)
------------------------------------------------------------------------------
  1            REFINED_1.pdb                               121
  2            REFINED_2.pdb                               65
  3            REFINED_3.pdb                               96
  4            REFINED_4.pdb                               73
  5            REFINED_5.pdb                               101
```

- **Step 4**: As per the job summary, 5 refined models are created under the directory /test/job-id/RESULT/.

6 References

1. Bhattacharya D, Cheng J: **3Drefine: Consistent protein structure refinement by optimizing hydrogen bonding network and atomic‐level energy minimization**. *Proteins: Structure, Function, and Bioinformatics* 2012.
2. Kalisman N, Levi A, Maximova T, Reshef D, Zafriri-Lynn S, Gleyzer Y, Keasar C: **MESHI: a new library of Java classes for molecular modeling**. *Bioinformatics* 2005, **21**(20):3931-3932.
3. Kabsch W, Sander C: **Dictionary of protein secondary structure: pattern recognition of hydrogen‐bonded and geometrical features**. *Biopolymers* 1983, **22**(12):2577-2637.