NASCA
User Manual
Version 1.0

Jianyang Zeng and Bruce R. Donald

Contents

1 Introduction

NASCA (NOE Assignment and Side-Chain Assignment) is an automated program for side-chain resonance assignment and nuclear Overhauser effect (NOE) assignment from NOESY data. It does not require data from TOCSY experiments. NASCA is integrated with a graphical user interface (GUI). NASCA is developed in the lab of Prof. Bruce Donald at Duke University.

NASCA is free software and can be redistributed and/or modified under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (optionally) any later version. NASCA is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. For full licensing details, including citation requirements for the software, please refer to Section 2 and Section 3, respectively. This information can also be found in the document `license.pdf` enclosed with this package distribution.

The current version (Version 1.0) of NASCA casts the assignment problem into a Markov Random Field (MRF), and extends and applies combinatorial protein design algorithms to compute optimal assignments that best interpret the NMR data. The complexity of the combinatorial search is reduced by using a dead-end elimination (DEE) algorithm, which prunes side-chain resonance assignments that are provably not part of the optimal solution. Then an A* search algorithm is employed to find a set of optimal side-chain resonance assignments that best fit the NMR data. These side-chain resonance assignments are then used to resolve the NOE assignment ambiguity.

2 License Information

The source header below must be included in any modification or extension of the source code of NASCA.

Source Header

```
NASCA NOE Assignment and Side-Chain Assignment  Software Version 1.0
Copyright (C) 2009-2011 Bruce Donald Lab, Duke University

NASCA is free software; you can redistribute it and/or modify it under
the terms of the GNU Lesser General Public License as published by the Free
Software Foundation, either version 3 of the License, or (at your option) any
later version.

NASCA is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
details.

You should have received a copy of the GNU Lesser General Public License
along with this library; if not, see:
    <http://www.gnu.org/licenses/>.
```

```
There are additional restrictions imposed on the use and distribution of this
open-source code, including: (A) this header must be included in any
modification or extension of the code; (B) you are required to cite our
papers in any publications that use this code. The citation for the various
different modules of our software, together with a complete list of
requirements and restrictions are found in the document license.pdf enclosed
with this distribution.

Contact Info:
    Bruce R. Donald
    Duke University
    Department of Computer Science
    Levine Science Research Center (LSRC)
    Durham, NC 27708-0129
    USA
    email: www.cs.duke.edu/brd/

<signature of Bruce Donald>, 01 May, 2011
Bruce R. Donald, Professor of Computer Science and Biochemistry
```

3 Citation Requirements

Any publications, grant applications, or patents that use NASCA must state that NASCA was used, with a sentence such as "We used the open-source NASCA software [Ref] to compute..."

In addition, you are required to cite our papers in any publications that use this code. The papers that can be cited based-on or related-to this software are listed below.

[a] Jianyang Zeng, Pei Zhou, and Bruce Randall Donald. Protein Side-Chain Resonance Assignment and NOE Assignment Using RDC-Defined Backbones without TOCSY Data. *Journal of Biomolecular NMR*, 2011 (in press).

[b] Jianyang Zeng, Pei Zhou, and Bruce Randall Donald. A Markov Random Field Framework for Protein Side-Chain Resonance Assignment. In *Proceedings of the 14th Annual International Conference on Research in Computational Molecular Biology (RECOMB'10), Lisbon, Portugal*, 2010.

4 Installation

4.1 System Requirements

To use this software, Java Runtime Environment from Sun Microsystems (JRE) must be installed. The newest version of JRE can be downloaded from http://java.sun.com/javase/downloads/.

4.2 Download and install the program

The latest version of NASCA can be downloaded from http://www.cs.duke.edu/donaldlab/software.php (as a zipped file). The installation of NASCA is simple: just simply decompress the downloaded zipped file, and extract all files into a working directory.

4.3 Run the program

NASCA supports two types of interface: graphical user interface (GUI) and command line. To run NASCA in the GUI model, simply double click the executable jar file named `Nasca.jar` in the working directory. The following main window (Fig. 1) should appear if the NASCA GUI is run properly properly:

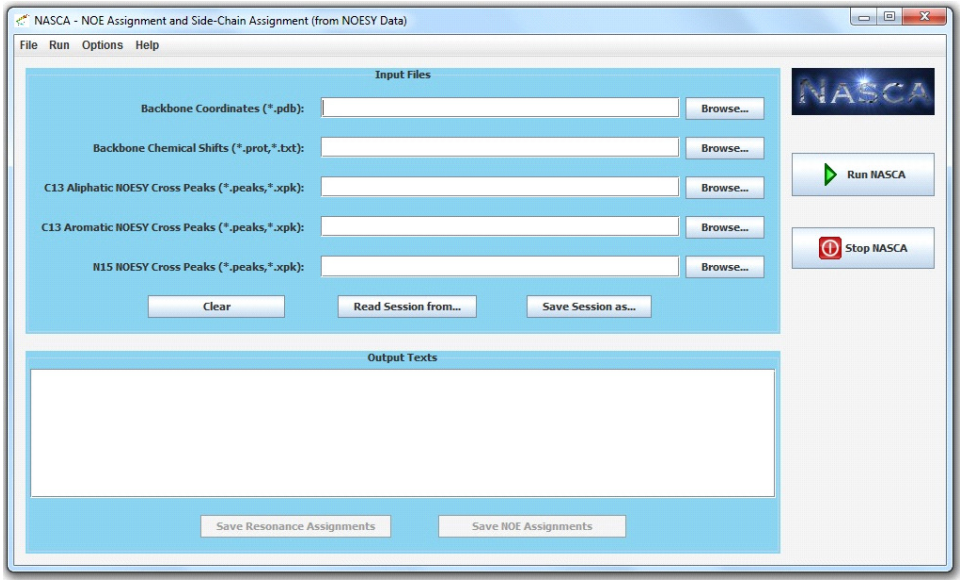

Figure 1: The main window of the NASCA GUI.

To run NASCA in the command line model, type the following command in the command window:

```
java -jar NascaCMD.jar -i Nasca.input -sc scResonancesOut.prot -noe noesOut.tbl
```

where "`-i Nasca.input`" specifies the input parameter file that gives the the paths of all input NMR files, "`-sc scResonancesOut.prot`" specifies the output file for storing the computed resonance assignments, and "`-noe noesOut.tbl`" specifies the output file for storing the computed NOE assignments. The formats of the input and output files will be described later (See Sections 5.2 and 6.2).

4.4 Compile the program (Optional)

1. Install Java Development Kit (JDK). The newest version of JRE can be downloaded from http://java.sun.com/javase/downloads/.
2. Modify the Makefile under the NASCA working directory as needed. In particular, specify your working directory and the paths to your Java compiler and resources.
3. Under the NASCA working directory, type "make".

5 System Configurations

5.1 File Organization

The executable jar files `Nasca.jar` and `NascaCMD.jar` are under the NASCA working directory. The directory structure of NASCA is organized as follows:

./src/ This directory contains the java source code.

./bin/ This directory contains the java *.class files.

./system/ This directory contains the system configuration files.

./doc/ This directory contains the documentations of the program.

./inputFiles/ This directory contains input NMR data.

./examples/ This directory contains several examples with real NMR input data for testing the program.

5.2 Input NMR Data and Formats

The input data to NASCA include: (1) The protein backbone PDB file, which also contains the protein primary sequence information; (2) The backbone chemical shifts; (3) the 3D NOESY cross peak list from both ^{15}N- and ^{13}C-edited spectra. The paths of these input files can be specified through the NASCA GUI (See Section 6), or an input parameter file (e.g., `Nasca.input`) in the command line model.

In our current protocol, we applied our recently-developed program RDC-PANDA [6, 7, 9, 2] to compute the backbone structures using two RDCs per residue (either NH RDCs measured in two media, or NH and CH RDCs measured in a single medium) and sparse NOE distance restraints. RDC-PANDA is also distributed open-source under the GNU Lesser General Public License, and can be downloaded from http://www.cs.duke.edu/donaldlab/software.php. In principle, modeling approaches, such as protein structure prediction [1], protein threading [8] or homology modeling [3, 4], could be used to compute the global fold. By using backbone chemical shift information, CS-ROSETTA [5] could also be used to predict the initial global fold.

In the input parameter file `Nasca.input` that specifies the locations of all input NMR data, lines starting with "//" are parsed as comments. Parameter names are single words; parameter values follow the corresponding parameter name on the same line and are separated by "=". Each parameter line ends with the symbol ";". The following gives an example of an input parameter file:

```
// File name of backbone PDB coordinates:
backbone = H:\workspace\ProteinNMR\inputFiles\eta_fragment.pdb;

// File name of backbone chemical shifts:
resonance = H:\workspace\ProteinNMR\inputFiles\eta_bb_bmrb.prot;

// File name of C-13 aliphatic NOESY cross peaks:
3D-C13-NOESY = H:\workspace\ProteinNMR\inputFiles\ali.peaks;

// File name of C-13 aromatic NOESY cross peaks:
3D-C13-NOESY-ARO = H:\workspace\ProteinNMR\inputFiles\aro.peaks;

// File name of N-15 NOESY cross peaks:
3D-N15-NOESY = H:\workspace\ProteinNMR\inputFiles\n15.peaks;
```

In the input backbone PDB file, the sequence of residue names should be consistent with the protein primary sequence. NASCA can support different input NMR data formats. It implements a function to automatically detect the format of each given input NMR data file. In particular, different PDB files with different atom naming schemes are supported by NASCA. For the input backbone chemical shifts, both BMRB and CYANA formats are supported by NASCA. For the NOESY cross peaks, both XEASY and NMRVIEW formats are supported by NASCA.

5.3 System Parameter Files

All system parameters and configuration files of NASCA are stored in directory `./system/`. In particular, the sub-directory `./system/rot-lib/` contains the rotamer library. The file `./system/BMRB_CS.txt` contains the BMRB statistical information. The current and default system parameters are specified in the system files `./system/nasca_system.input` and `./system/nasca_system_default.input`, respectively. The system parameter files use the same format as in the input parameter file (e.g., `Nasca.input`), that is, lines starting with "//" are parsed as comments. Parameter names are single words; parameter values follow the corresponding parameter name on the same line and are separated by "=". Each parameter line ends with the symbol ";". The following gives an example of system parameter file:

```
//error windows (in ppm unit) of chemical shifts
haErr    = 0.030;
hnErr    = 0.030;
h1Err    = 0.030;
c13Err   = 0.300;
nErr     = 0.300;

//NOE distance threshold
noeLimit = 5.00;

//default input file directory
input_directory = H:\workspace\ProteinNMR/inputFiles/;
```

```
//default output file directory
output_directory = H:\workspace\ProteinNMR/inputFiles/;
```

The system parameters can be specified through menu `Options -> Parameters` in the NASCA GUI (See Section 6) or manually editing the system parameter file `./system/nasca_system.input`.

5.4 Output File Format

The output files of NASCA include both computed side-chain resonance assignments and NOE assignments. By default, the output side-chain resonance assignments are saved in BMRB format, and the output NOE assignments are saved in XPLOR format. Other data formats may be supported for saving the output side-chain resonance assignments and NOE assignments in a future version of NASCA.

6 Graphical User Interface (GUI)

6.1 Overview

The main window of the NASCA GUI has four areas, namely input, output, command and menu areas, as shown in Fig. 2. The input area mainly deals with the input NMR data. The output area prints out the log messages and outputs the side-chain resonance assignments and NOE assignments computed by NASCA. The command area contains two buttons that are used to run and stop a NASCA thread. The menu area includes menu items for necessary operations, such as setting system parameters. Below we will show the details of each area.

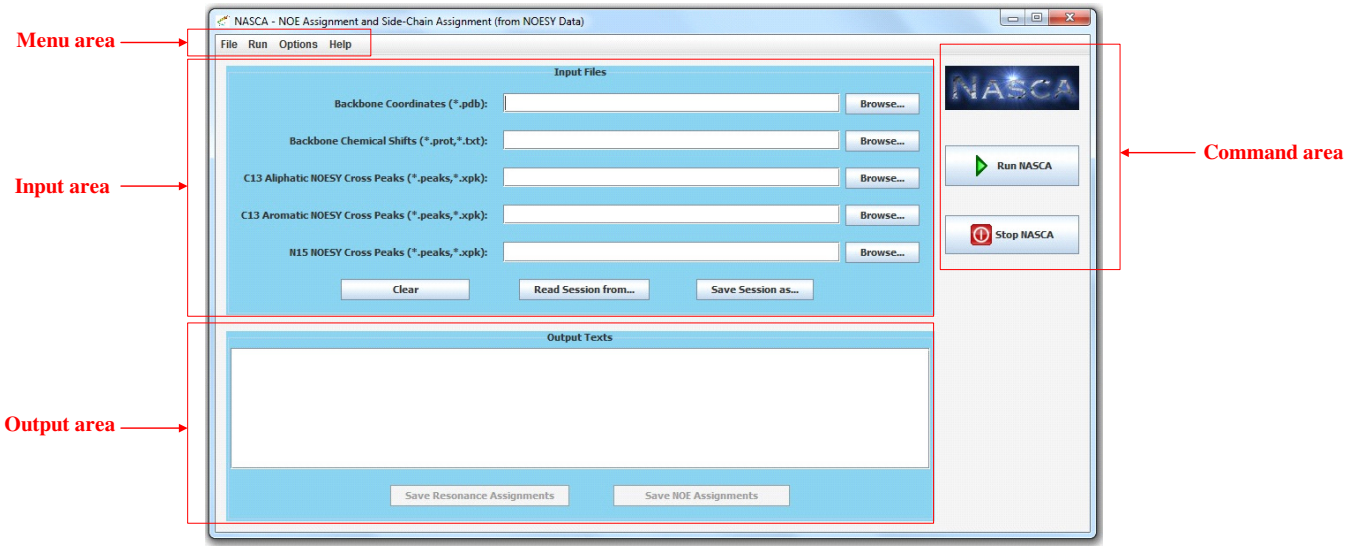

Figure 2: Overview of the NASCA GUI.

6.2 The Input Area

Fig. 3 shows the details on the input area of the NASCA GUI. The user can use the `Browse` button to open each input file. Once an input file is specified, its path name will be shown in the corresponding text field. As mentioned in Section 5.2, NASCA can support different input NMR data formats. It implements a function to automatically detect the format of each given input NMR file. In particular, different PDB files with different atom naming schemes are all supported by NASCA. For the input backbone chemical shifts, both BMRB and CYANA formats are supported by NASCA. For the NOESY cross peaks, both XEASY and NMRVIEW formats are supported by NASCA. For each dialog window for opening an input file, NASCA also incorporates a file filter (See Fig. 4 for an example).

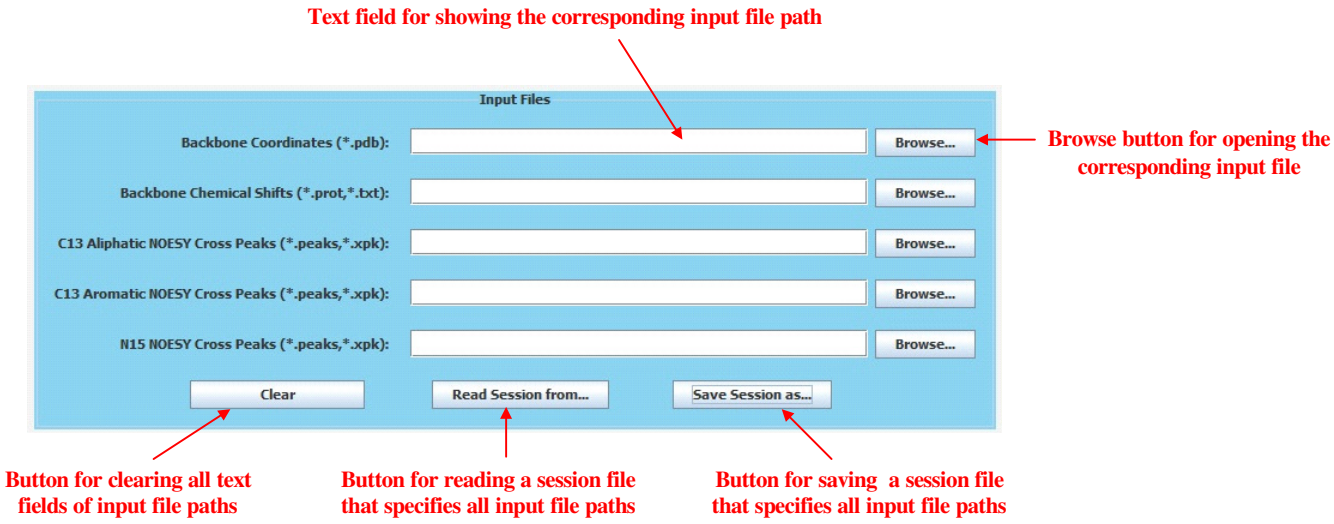

Figure 3: The input area of the NASCA GUI.

Three buttons at the bottom of the input area, namely `Clear`, `Read Session from...` and `Save Session as...`, are used for batch processing the input data files. The `Clear` button clears all text fields of input file paths. The `Read Session from...` button reads a session file that specifies all input data files. The same operation has also been implemented in the menu item `File -> Read Session from...`. The `Save Session as...` button saves a session file that specifies all input data files. The same operation has also been implemented in the menu item `File -> Save Session as...`. Note that the format of a session file is consistent with the input parameter file (i.e., `Nasca.input`) used in the command line model (See Section 5.2).

6.3 The Output Area

Fig. 5 shows the details on the output area of the NASCA GUI. The output text area is used to print out the log messages, such as the progress and error messages, while running the program. The button `Save Resonance Assignments` is used to save the computed side-chain resonance assignments, and the button `Save NOE Assignments` is used to save the computed NOE assignments. Initially these two buttons are disabled, and they are enabled when the program is finished and the computed side-chain resonance assignments and NOE assignments area available. As mentioned in

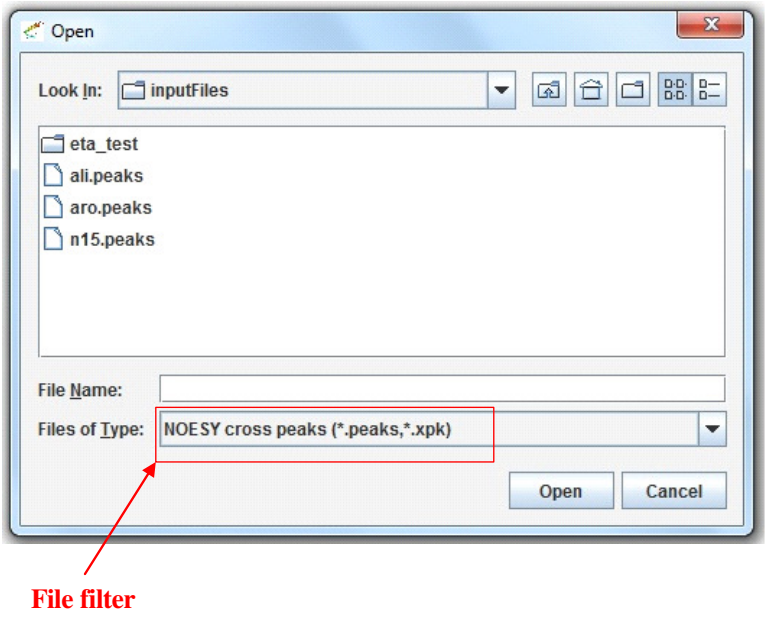

Figure 4: An example of file filter when opening an input file.

Section 5.4, by default, the output side-chain resonance assignments are saved in BMRB format, and the output NOE assignments are saved in XPLOR format. Alternatively, saving the computed side-chain resonance assignments and NOE assignments can be executed by clicking the following two menu items: `File -> Save Resonance Assignments` and `File -> Save NOE Assignments`, respectively.

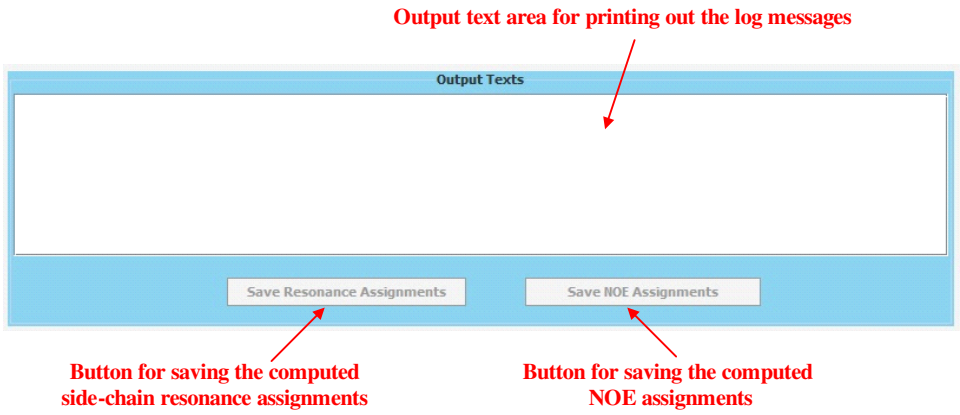

Figure 5: The output area of the NASCA GUI.

6.4 The Command Area

Fig. 6 shows the details on the command area of the NASCA GUI. Once all input data files are specified, NASCA can be performed by clicking the `Run NASCA` button. If the program runs without

being interrupted, it should compute and output the resonance assignments and NOE assignments in the output text area. The user can use buttons `Save Resonance Assignments` and `Save NOE Assignments` to save the finally-computed assignments into the specified file paths. The user can also cancel a running NASCA thread by clicking the `Stop NASCA` button.

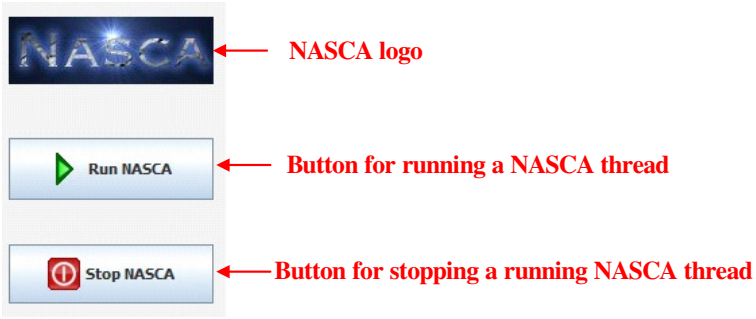

Figure 6: The command area of the NASCA GUI.

6.5 The Menu Area

The `File` menu contains the following items: `Read Session from...`, `Save Session as...`, `Save Resonance Assignments`, `Save NOE Assignments` and `Quit`. The first four menu items perform the same operations as the corresponding buttons in the NASCA GUI. Particularly, the first two menu items `Read Session from...` and `Save Session as...` read and save a session file that specified all input files, respectively. The menu items `Save Resonance Assignments` and `Save NOE Assignments` save the computed resonance assignments and NOE assignments, respectively. The menu item `Quit` quits the program.

The `Run` menu contains two items, namely `Run NASCA` and `Stop NASCA`, which are used for running and cancelling a NASCA thread. They perform the same operations as the corresponding buttons in the command area.

Currently the `Options` menu contains only one item, namely `Parameters`, which is used to set the system parameters. Once the `Parameters` menu item is clicked, the following window (see Fig. 7) will pop up. As shown in Fig. 7, the user can change the following system parameters: the error windows (in ppm unit) in each dimension for constructing the NOESY graph, distance threashold for constructing edges between proton labels, and default input and output file directories. The user can also set all parameters to the default settings by clicking the `Default` button.

The `Help` menu contains two items, namely `User Manual` and `About`. The `User Manual` directs the user to a web page in which the user manual can be found. The following dialog window (see Fig. 8) will pop up if the `About` menu item is clicked.

7 Examples

The distribution of NASCA also contains several examples for running the program, including the tests on pol η UBZ and ubiquitin proteins. The input data and parameter files for running these examples can be found in directory `./examples/`. It should be straightforward to use the NASCA

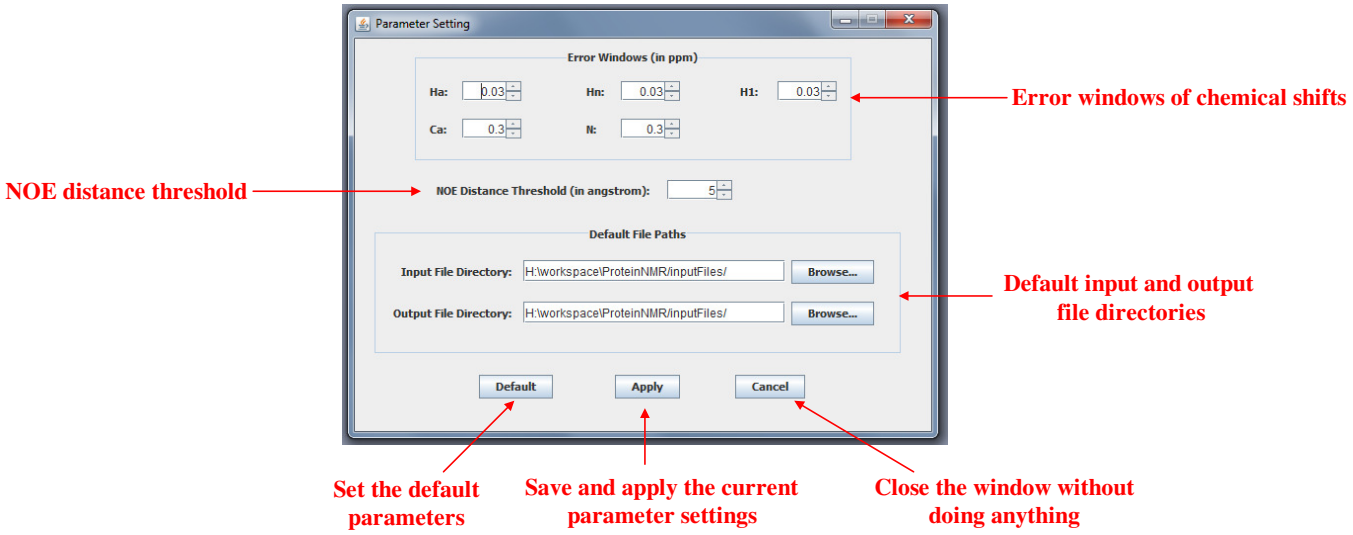

Figure 7: The dialog window for setting the system parameters.

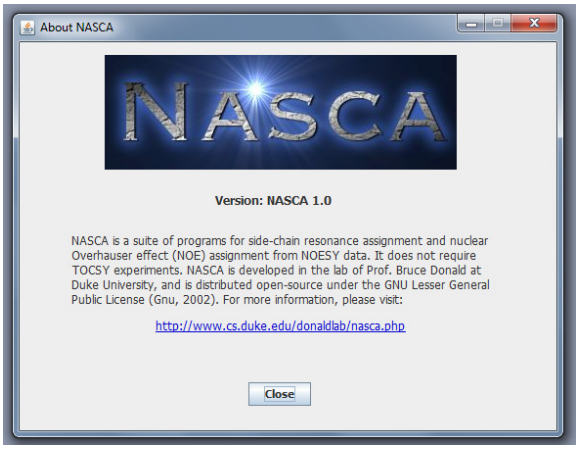

Figure 8: The `About` dialog window.

to run these examples. To run these examples in the command line model, use `Nasca.input` as the input parameter file.

[1] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294:93–96, 2001.

[2] B. R. Donald and J. Martin. Automated NMR assignment and protein structure determination using sparse dipolar coupling constraints. *Progress in NMR Spectroscopy*, 55:101–127, 2009.

[3] C. J. Langmead and B. R. Donald. 3D structural homology detection via unassigned residual dipolar couplings. In *Procedings of 2003 IEEE Comput Syst Bioinform Conf*, pages 209–217, 2003.

[4] C. J. Langmead and B. R. Donald. High-throughput 3D structural homology detection via NMR resonance assignment. In *Procedings of 2004 IEEE Comput Syst Bioinform Conf*, pages 278–289, 2004.

[5] Yang Shen, Oliver Lange, Frank Delaglio, Paolo Rossi, James M Aramini, Gaohua Liu, Alexander Eletsky, Yibing Wu, Kiran K Singarapu, Alexander Lemak, Alexandr Ignatchenko, Cheryl H Arrowsmith, Thomas Szyperski, Gaetano T Montelione, David Baker, and Ad Bax. Consistent blind protein structure generation from NMR chemical shift data. *Proc Natl Acad Sci U S A*, 105(12):4685–4690, Mar 2008.

[6] L. Wang and B. R. Donald. Exact solutions for internuclear vectors and backbone dihedral angles from NH residual dipolar couplings in two media, and their application in a systematic search algorithm for determining protein backbone structure. *Jour. Biomolecular NMR*, 29(3):223–242, 2004.

[7] L. Wang, R. Mettu, and B. R. Donald. A Polynomial-Time Algorithm for De Novo Protein Backbone Structure Determination from NMR Data. *Journal of Computational Biology*, 13(7):1276–1288, 2006.

[8] Y. Xu, D. Xu, and E. C. Uberbacher. An efficient computational method for globally optimal threading. *J Comput Biol.*, 5(3):597–614, 1998.

[9] Jianyang Zeng, Jeffrey Boyles, Chittaranjan Tripathy, Lincong Wang, Anthony Yan, Pei Zhou, and Bruce R. Donald. High-resolution protein structure determination starting with a global fold calculated from exact solutions to the RDC equations. *Journal of Biomolecular NMR*, 45(3):265–281, Nov 2009.