BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 12 2011, pages 1713–1714
doi:10.1093/bioinformatics/btr255

Structural bioinformatics

Advance Access publication May 5, 2011

Grid computing for improving conformational sampling in NMR structure calculation

Fabien Mareuil[1,2], Christophe Blanchet[2], Thérèse E. Malliavin[1,*] and Michael Nilges[1,*]

[1]Unité de Bioinformatique Structurale, CNRS URA 2185, Institut Pasteur 25-28 rue du Dr Roux, F-75724 Paris Cedex 15 and [2]Université Lyon 1, Univ Lyon, France; CNRS, FR 3302 ; Institut de Biologie et Chimie des Protéines, IBCP, 7 passage du vercors, F-69367, France

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Methods for automatic nuclear magnetic resonance (NMR) structure determination need to face a high level of ambiguity encountered in NMR spectra recorded by solid-state NMR and by solution NMR of partially unfolded proteins, leading to time-consuming calculations. The software package Ambiguous Restraints for Iterative Assignment (ARIA) allows for straightforward parallelization of the calculation, as the conformers can be generated in parallel on many nodes.

Results: Due to its architecture, the adaptation of ARIA to grid computing can be easily achieved by using the middleware glite and JDL (Job Description Language) scripts. This adaptation makes it possible to address highly ambiguous datasets, because of the much larger conformational sampling that can be generated by use of the grid computational power.

Availability: The version 2.3.1 of ARIA implemented on the grid is freely available from the ARIA web site: aria.pasteur.fr/downloads.

Contact: nilges@pasteur.fr; tere@pasteur.fr

Received and revised on April 1, 2011; accepted on April 8, 2011

Nuclear Magnetic Resonance (NMR) is one of the major techniques for biomolecular structure determination (Wüthrich, 1986), with decisive advantages for particular cases. NMR structure determination makes it necessary to assign the contacts (NOEs for liquid state NMR, other dipolar couplings for ssNMR), a step that is performed during the structure calculation in an iterative protocol within Ambiguous Restraints for Iterative Assignments (ARIA) (Nilges *et al.*, 1997). In each iteration, the contacts are assigned based on structures from the previous iteration, and the data points most inconsistent with the conformations are removed with a simple statistical analysis (Guntert, 2004; Rieping *et al.*, 2006). Protein conformations are then calculated with distance restraints, which are based on the current set of assignments but usually retain a large level of ambiguity. Convergence of the structures improve from iteration to iteration, and the calculation is terminated when a converged set of conformations is obtained with good restraints fit.

The efficiency and even the feasibility of the automatic NOE assignment depends on the spectral resolution, i.e. on the level of ambiguity in the NMR spectra. NMR started recently to tackle problems where high levels of ambiguity are encountered, as the study of membrane or fibrillar proteins with solid-state NMR (ssNMR), and the study of partially unfolded proteins with solution NMR. The analysis of these structural landscapes is essential to understand important biological phenomena (Böckmann and Meier, 2010; Korzhnev *et al.*, 2010; Leroy *et al.*, 2010; Montserret *et al.*, 2010).

The iterative ARIA cycle is based on three major steps: (i) input preparation for the generation of conformations; (ii) generation of conformations by a simulated annealing procedure using the software CNS (Brunger *et al.*, 1998); and (iii) analysis of the obtained conformations to generate a new set of NOE assignments and restraints. ARIA is straightforwardly parallelized by distributing the step (ii) over a few or many CPU units.

A significant increase in the number of calculated protein conformations improves the statistics on the NMR conformations and can help to overcome the ambiguity bottleneck. Grid computing is an attractive option for the parallel computing of conformations, because of the large number of computing elements, and since it brings parallel computing even to laboratories that lack the in-house infrastructure.

We have adapted ARIA to grid computing in the frame of GRISBI (Grid Support to Bioinformatics: www.grisbio.fr) (Blanchet *et al.*, 2006a, b), which is a transverse project of the French Bioinformatics network RENABI (REseau NAtional des plateformes BIoinformatiques: http://www.renabi.fr/).

In order to use the grid, the user needs to obtain a digital certificate from a trusted Certification Authority, to register on a virtual organization (VO) and to have an account on a user interface (UI) computer. A proxy certificate, valid by default for 24 h, has to be created to authenticate the user in subsequent secure interactions. The UI performs the steps (i) and (iii) of the ARIA iteration, and dispatches tasks related to step (ii) on a Workload Management System (WMS) that submit each task to an appropriate computing element (CE). The job submission and management on the grid computing elements is performed through the middleware glite, developed by the EGEE project (www.eu-egee.org).

The glite commands, glite-wms-job-submit, glite-wms-job-status, glite-wms-job-output and glite-wms-job-cancel, are used in the ARIA grid implementation. The job submission is done through a Job Description Language (JDL) file generated by ARIA. The command glite-wms-job-submit submits a job to the WMS, by uploading the CNS input files to transfer them to the most appropriate computing element (Fig. 1).

After a successful submission, ARIA uses the glite-wms-job-status command to query the WMS about the state of each job

*To whom correspondence should be addressed.

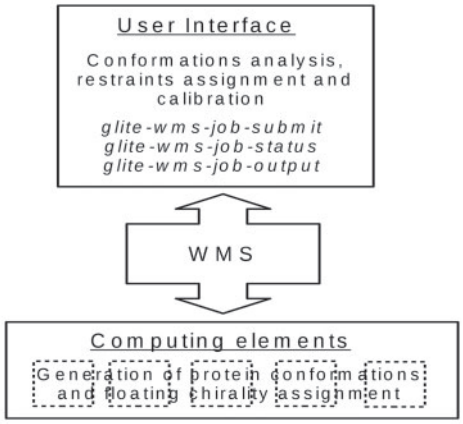

Fig. 1. Implementation of ARIA in the frame of grid computing.

every 2 min. If the job state is 'done', the glite-wms-job-output command allows to download an archive of the CNS outputs from the computing element to the UI. Once all jobs are correctly finished, ARIA performs the step (iii) on the UI in order to prepare the updated restraints for the next iteration.

A grid is not as reactive and safe as a cluster, and delays in computing element response can slow down the conformer generation. To avoid these delays, two jobs are submitted for each CNS calculation, the output of the faster job being conserved and the other job being canceled. Several python procedures have been implemented in ARIA to handle errors during the computing element calculation, during the job submission. The processed errors are as follows: (i) submission or network failure, (ii) job abortion by WMS, (iii) unsuccessful job output download. In these cases, ARIA resubmits automatically the job with an output message. The input and output files of the conformation generation are uploaded and downloaded through the Sandbox, that is a space storage on the WMS restrained to 10 MB.

Three python classes of ARIA have undergone most of the changes to adapt ARIA to the grid: (i) class Job for managing the CNS jobs, (ii) class cns for preparing the CNS input files, (iii) class jobmanagerpanel for connecting to the graphical user interface. The running mode of ARIA is defined in the panel jobmanager of the graphical interface or in the element jobmanager of the project xml file. ARIA can run on a single computer (LOCAL mode), on a cluster of processors related by a fast local area network and sharing the same data and storage information (CLUSTER mode) or on a grid (GRID mode). The panel jobmanager contains also the management job commands. The state_command and the output_command attributes have to be filled only if ARIA is run in GRID mode.

The state and the id of each job submitted to the grid are traced in files .run located in the ARIA run directory. 'Not run' means that the job is not launched yet or aborted. The two submitted jobIDs are given if they are running. 'Cleared' means that the fastest job ended successfully and was retrieved on the UI.

The future perspective is the adaptation of ARIA to cloud computing, where external customers buy on demand CPU, storage, computing power from a supplier. This approach increases the reliability and the flexibility, and is certainly the future of intensive computation for parallel calculations with no or little communication. In the framework of the cloud computing approach, ARIA faces the challenge of the encapsulation into a virtual machine, due to the large number of libraries and software packages involved [CCPN: Fogh *et al.* (2002); Vranken *et al.* (2005), CNS: (Brunger *et al.*, 1998), TkInter].

ACKNOWLEDGEMENTS

The authors thank Dr Anja Böckmann for fruitful discussions at the beginning of the ARIA grid implementation.

Funding: ANR THALER 'Massively parallel simulation and analysis of protein structure and dynamics', CNRS, Institut Pasteur, GIS IBiSA (www.ibisa.net).

Conflict of Interest: none declared.

REFERENCES

Blanchet,C. *et al*. (2006a) Integrating bioinformatics resources on the EGEE Grid platform. In *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid*, p. 48.

Blanchet,C. *et al*. (2006b) Grid deployment of legacy bioinformatics applications with transparent data access. In *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing (GRID 2006), September 28–29, 2006, Barcelona, Spain*. IEEE.

Böckmann,A. and Meier,B. (2010) Prions: En route from structural models to structures. *Prion*, **4**, 72–79.

Brunger,A. *et al*. (1998) Crystallography & NMR system: a new software suite for macromolecular structure determination. *Acta Crystallogr. D Biol. Crystallogr.*, **54**, 905–921.

Fogh,R. *et al*. (2002) The CCPN project: an interim report on a data model for the NMR community. *Nat. Struct. Biol.*, **9**, 416–418.

Guntert,P. (2004) Automated NMR structure calculation with CYANA. *Methods Mol. Biol.*, **278**, 353–378.

Korzhnev,D. *et al*. (2010) A transient and low-populated protein-folding intermediate at atomic resolution. *Science*, **329**, 1312–1316.

Leroy,A. *et al*. (2010) Spectroscopic studies of GSK3beta phosphorylation of the neuronal tau protein and its interaction with the N-terminal domain of apolipoprotein E. *J. Biol. Chem.*, **285**, 33435–33444.

Montserret,R. *et al*. (2010) NMR structure and ion channel activity of the p7 protein from hepatitis C virus. *J. Biol. Chem.*, **285**, 31446–31461.

Nilges,M. *et al*. (1997) Automated NOESY interpretation with ambiguous distance restraints: the refined NMR solution structure of the pleckstrin homology domain from beta-spectrin. *J. Mol. Biol.*, **269**, 408–422.

Rieping,W. *et al*. (2006) ARIA2: automated NOE assignment and data integration in NMR structure calculation. *Bioinformatics*, **23**, 381–382.

Vranken,W. *et al*. (2005) The CCPN data model for NMR spectroscopy: Development of a software pipeline. *Proteins*, **59**, 687–696.

Wüthrich,K. (1986) *NMR of Proteins and Nucleic Acids*. Wiley, New York, USA.