ALMOST: An All Atom Molecular Simulation Toolkit for Protein Structure Determination

Biao Fu,[a] Aleksandr B. Sahakyan,[a] Carlo Camilloni,[a] Gian Gaetano Tartaglia,[b] Emanuele Paci,[c] Amedeo Caflisch,[d] Michele Vendruscolo,[a] and Andrea Cavalli*[a,e]

Almost (**al**l atom **mo**lecular **s**imulation **t**oolkit) is an open source computational package for structure determination and analysis of complex molecular systems including proteins, and nucleic acids. **Almost** has been designed with two primary goals: to provide tools for molecular structure determination using various types of experimental measurements as conformational restraints, and to provide methods for the analysis and assessment of structural and dynamical properties of complex molecular systems. The methods incorporated in **Almost** include the determination of structural and dynamical features of proteins using distance restraints derived from nuclear Overhauser effect measurements, orientational restraints obtained from residual dipolar couplings and the structural restraints from chemical shifts. Here, we present the first public release of **Almost**, highlight the key aspects of its computational design and discuss the main features currently implemented. **Almost** is available for the most common Unix-based operating systems, including Linux and Mac OS X. **Almost** is distributed free of charge under the GNU Public License, and is available both as a source code and as a binary executable from the project web site at http://www.openalmost.org. Interested users can follow and contribute to the further development of **Almost** on http://sourceforge.net/projects/almost. © 2014 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23588

Introduction

The determination of the structure of molecules from experimental observables is one of the essential aspects of structural biology.[1,2] Characterizing the structure and dynamics of proteins, nucleic acids and their complexes is, however, a very difficult task. The major problems in this challenge are, at least in part, stemming from the inherent complexity associated with an accurate modeling of some of the most important experimental observables, as well as from the acquisition of a working knowledge of a variety of protocols for structure determination and assessment that are required in the modeling process.[3–13]

In this context, the aim of **Almost** (**al**l atom **mo**lecular **s**imulation **t**oolkit)**,** the computational package presented here, is to provide a unified platform for the structure determination of proteins and nucleic acids to facilitate the usage and further development of a wide range of algorithms and protocols for structure determination from experimental observables.

The methods incorporated within **Almost** include both well established methodologies, such as the calculation of bundles of three-dimensional (3D) structures derived from nuclear Overhauser effects (NOEs) distance restraints, or through the automated assignment of NOEs, and approaches based on more recent developments, such as CHESHIRE[14,15] that uses only nuclear magnetic resonance (NMR) chemical shifts (CS) as experimental input.

To achieve its objectives, **Almost** has been designed with a flexible architecture. The **Almost** source code consists of three application layers, as illustrated in Figure 1, which can be accessed through an integrated scripting language. The innermost layer contains the core data structures for the computational representation of molecules, such as proteins and nucleic acids, and routines for the data import and export. The second layer contains the core algorithms required for an efficient calculation of energies and interaction forces, and fast modules to perform molecular dynamics or Monte Carlo simulations. The third layer consists of methods and algorithms for the structural analysis and assessment of structures and trajectories. All structures and algorithms in the three layers are exposed to an integrated object-oriented scripting language (A++). Finally, higher-level protocols are written using the integrated scripting language A++ and are part of the ALMLIB library which is distributed together with the source code.

In addition to the core algorithms and protocols for structure determination of proteins and nucleic acids, **Almost** provides an extensive support for the study of structural and

[a] *B. Fu, A. B. Sahakyan, C. Camilloni, M. Vendruscolo, A. Cavalli*
Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge, CB2 1EW, United Kingdom
[b] *G. G. Tartaglia*
Center for Genomic Regulation (CRG) and Universitat Pompeu Fabra (UPF), Barcelona, Spain
[c] *E. Paci*
School of Molecular and Cellular Biology, University of Leeds, Leeds, United Kingdom
[d] *A. Caflisch*
Department of Biochemistry, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
[e] *A. Cavalli*
Institute for Research in Biomedicine (IRB), via Vincenzo Vela 6, 6500, Bellinzona, Switzerland
E-mail: andrea.cavalli@irb.usi.ch

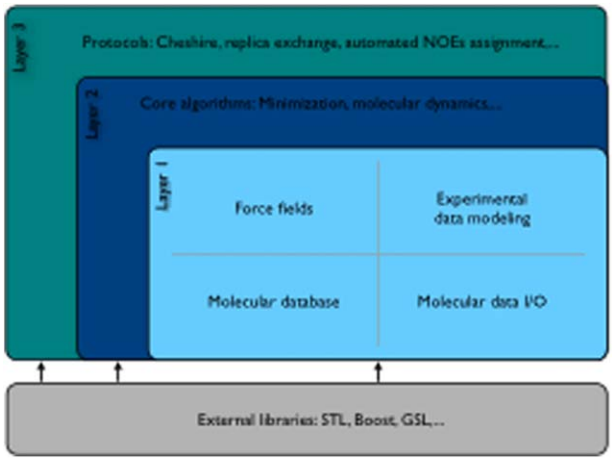

Figure 1. Organization of the **Almost** package, with its three application layers and the interfacing higher-level protocols.

dynamical properties of biomolecules using wide range of experimental observables including residual dipolar couplings (RDCs), CS, chemical shift anisotropies, and small angle X-ray scattering (SAXS) data. Finally, **Almost** implements tools and methods for the structural analysis and assessment.

All the functionalities of **Almost** can be accessed via the embedded scripting language A++. The use of an embedded scripting language, instead of exposing functionalities to existing languages or using "input configuration files," presents at least two major advantages. First, it makes the use of the code in heterogeneous contexts easier because the user does not have to be concerned with third party software versions, installation of modules and the configuration steps required to use shared modules. Second, it is very effective in providing the required flexibility for the implementation of complex protocols on top of the core features provided by the **Almost** platform. The hurdle associated with learning yet another scripting language can be minimized by choosing a syntax that resembles the one of the most common languages, such as C, C++, and JavaScript. A++ is, in fact, a simple, object-oriented scripting language, which was designed with the purpose to serve as a glue language for classes and functions written in C++. This means, for example, that A++ uses the same fundamental types (string, integer, float, etc.) and the same typecasting rules as C++, which makes exposing complex C++ classes relatively easy and safe. A++ is developed and maintained together with the rest of the **Almost** platform. Further details are covered in the **Almost** manual available online (http://www.open-almost.org).

Implementation

Almost is written in C++, with some C and functions in the innermost loops of the energy and force calculation routines to achieve a better performance. The choice of C++, as the implementation language was made because of its object-oriented nature, which encourages the development of clearly defined modules and interfaces. In turn, this aspect greatly facilitates the development and maintenance of a relatively large code base. Moreover, performance penalties usually ascribed to C++ can be avoided by a careful implementation of data structures and a proper programming style. Avoiding certain C++ features, such as virtual functions, minimizing the number of conditional jumps in the innermost loops and the usage of modern optimizing compilers is often sufficient to guarantee performances that are almost optimal. Finally, C++ provides advanced constructs, such as the template mechanism, which can be used to write self-optimizing code. Moreover, the use of template classes makes it possible to reuse large portion of code in a clean way. In **Almost**, for example, all force fields and restraints are implemented as template classes with the type of float used as a parameter. This makes it easy to use single or double precision algorithms depending on the hardware and the accuracy requirements.

The scripting language A++ was implemented using the lexical analyzer LEX and the parser generator BISON, provided by the Free Software Foundation.[16]

Results and Discussion

We discuss here the usage of **Almost,** first by presenting an overview of its functionalities (see Table 1) and then by providing representative examples of its usage. The functionalities of **Almost** are not limited to those areas and span a much wider range. The small selection of the examples is presented as just a case study to illustrate the potential of the **Almost** platform.

Overview of functionality

Almost contains many general-purpose tools to generate and manipulate molecular structures. The initial configurations of proteins and nucleic acids can be generated from their amino acid and nucleotide sequences. Coordinates and trajectories can be analyzed using the commands of the built-in scripting language, which include commonly performed tasks, such as fitting a structure to another one using a subset of atoms, computing the variance within a bundle of structures or calculating the average structure from an ensemble. Further details are covered in the **Almost** manual available online (http://www.open-almost.org).

The generation of structures or ensembles of structures that corresponds to a given set of experimental observables has been one of the driving forces behind the development of **Almost**. For this reason, **Almost** implements several algorithms for energy minimization and molecular dynamics with structural restraints. In Cartesian coordinates, steepest-descent and conjugated gradient energy minimization routines are provided and molecular dynamics can be carried out at constant temperature using the Berendsen thermostat [17] or Langevin dynamics.[18] In addition to the methods operating in Cartesian coordinates, **Almost** implements a fast algorithm for energy minimization and molecular dynamics in internal (dihedral angle) coordinates.[19]

Force fields. **Almost** implements several force fields including CHARMM,[6] AMBER,[20] and OPLS,[21] together with several

implicit solvation models, such as SASA,[22] EEF1,[24] and several flavors of generalized Born solvation.[25]

Experimental observables. Several NMR and X-ray derived observables can be used to model structural features of proteins and other biomolecules with the current version of **Almost**. To achieve this goal, experimental observables are imposed as structural restraints using suitable energy terms that can be added to background molecular mechanics force fields**. Almost** implements restraining energy terms for commonly used geometrical features, such as distances, angles, dihedral angles, and for the quantities directly measured in experiments, such as CS[14,26–28] and CS anisotropies, paramagnetic contact shifts, SAXS[29] and RDC.[23]

Structure determination from NMR chemical shifts

Almost provides support for the determination of the structure of proteins using the CHESHIRE pipeline.[14,15,30]

CHESHIRE uses the $^{[15]}N$, $^{[1]}H^N$, $^{[1]}H\alpha$, $^{[13]}C\alpha$, $^{[13]}C\beta$, and $^{[13]}C'$ backbone CSs for the determination of the structure of proteins. The whole CHESHIRE pipeline, which was described in detail previously,[14] consists of three main steps. In the first one, the backbone CSs are used to select small protein fragments with a similar structure from a database. In the second step, the selected fragments are assembled using molecular fragment replacement. Finally, the best scoring models from the fragment replacement phase are refined using CS and the CHARMM/EEF1[24] molecular dynamics force field. Despite the complexity of the CHESIRE pipeline, a structure determination with CHESHIRE is actually fairly straightforward. CHESHIRE requires only a sequence file (in FASTA format) and a table with the backbone CSs as an input, with the following format:

```
12 ASP 4.273 8.586 126.315 57.608 41.161 178.089
13 GLU 4.026 8.289 119.325 59.006 28.709 178.893
14 LEU 3.985 7.577 121.683 57.261 40.105 178.309
15 ILE 3.219 8.238 119.483 66.590 37.424 177.138
16 LYS 3.843 7.604 117.596 60.169 32.164 179.577
17 LYS 3.992 7.583 118.832 59.840 33.065 179.380
18 ILE 3.372 8.785 120.961 65.474 37.556 176.967
19 LYS 3.680 8.532 118.324 60.853 32.326 177.746
```

The script cheshire_template.z can then be used to generate the necessary scripts and directory structure (see the manual for the complete list of parameters):

```
almost -f cheshire_template.z -fasta test.fasta
-cs test.cs -pdb test.pdb
```

The template file generator will create, besides the appropriate directory tree, two scripts, cheshire.z and generate.z. The first script is used to validate the input by performing a series of consistency crosschecks and to perform the first step of the CHESHIRE protocol, which is the determination of the secondary structure and the generation of the fragment libraries. The second script is used to compute the structure of the protein, that is, it will perform the steps two and three of the pipeline. Because of the computational cost of the whole calculation, the structures can be computed in parallel:

```
for i in 1 2 3 4 5 6 7 8; do
almost -f generate.z &
done
```

Almost automatically performs the synchronization between the various instances.

Structure determination from unassigned NOESY spectra

The typical protocol for protein structure determination by NMR spectroscopy involves a number of sequential steps.[31] In the first step, the CSs observed in multidimensional NMR spectra are assigned to their corresponding atoms. Then, in a second step, thousands of peaks in multidimensional NOEs spectra are identified, assigned, and converted into interatomic distance restraints (peak-peaking and assignment). Finally, molecular dynamics is used to generate a set of protein structures that should satisfy these experimental restraints (structure generation step). The NOE assignment and structure generation steps are usually performed over several iterations to maximize the number of distance restraints obtained from the spectra, while guaranteeing that the restraints are self-consistent. These iterations are time-consuming, error-prone, and require a considerable bookkeeping to ensure that the final set of structures satisfy all experimental data. To enhance productivity and reproducibility of the NMR structure determination process, several protocols aimed at the automation of the aforementioned steps have been proposed.[32]

Almost currently provides support for the automation of steps 2 and 3, that is, the structure determination from unassigned peaks lists obtained from multidimensional Overhauser effects NOEs spectroscopy spectra. The method is expected to gain considerable interest in the near future, as it has been shown that unassigned NOE data can be used to determine structures of very good quality.[3] **Almost** implements two different protocols for this purpose. The two algorithms are based on the CANDID and CYANA protocol.[33,34] The main difference is that **Almost** provides the possibility to use a variant of the original algorithm which is more robust in cases where the peak lists contain a considerable amount of noise. The latter variant of the algorithm has been used for the automated calculations structure determination form raw peak lists in the third round of the CASD-NMR experiment and will be described in more details in a forthcoming article.

A general input for the automated structure determination from unassigned NOE peak lists is the following:

```
calc = nevermind.calcnmr("HR2876C","HR2876C.
seq","HR2876C-final.prot",0.03,0.03,0.3);
//Read unassigned peak lists
calc.add_peaklist("C13.peaks");
calc.add_peaklist("N15.peaks");
calc.add_peaklist("ARO.peaks");
//Add torsion angle restraints
calc.set_aco("talos.aco");
//Run (7) cycles of NOE assignment-structure
generation
calc.plan7(240,20);
```

Table 1. Overview of features implemented in **Almost**.

Core algorithms	Basic protocols
Molecular dynamics in Cartesian coordinates	CHESHIRE
Molecular dynamics in torsion angles coordinates	YAPP
Replica exchange	CHESHIRE-YAPP
Fragment replacement	CAMSHIFT-MD
Monte Carlo	**Experimental data**
Metadynamics	Chemical Shifts for back bone and side chains
	Residual dipolar couplings
	Chemical shift anisotropy
	Small X-ray scattering
	Basic: bond, angle and so forth.

Chemical shift refinement using CamShift

In addition to the methods presented above, **Almost** also provides support for molecular dynamics simulations using CSs as structural restraints.

The use of CSs for protein structure determination has been a long-standing goal is structural biology, since these NMR observables are measurable under very general conditions. One of the major obstacles, however, has been the difficulty to understand in sufficient detail the complicated conformational dependencies of the CSs. A step forward in this direction was the development of CamShift chemical shift predictor,[26] which demonstrated that CSs can be expressed through differentiable functions of the atomic positions in a sufficiently accurate manner. This initial model was subsequently modified to extend the chemical shift predictions toward side-chain methyl and aromatic ring protons.[27,28]

Chemical shift based molecular dynamics simulations, through their implementation in **Almost** platform, have already been used to study protein structure and dynamics both in their folded[35] and low-populated intermediate[36] states.

Almost has also been used, together with PLUMED,[37] as a source for introducing experiment-based collective variables (CS, NOE, etc.) in molecular simulations. In particular, the implementation of CamShift has been used for the characterization of protein structure and dynamics through the use of replica-averaged restrained simulations.[38,39]

Required input

A molecular simulation (e.g., molecular dynamics or refinement) that involves the use of CSs with CamShift is very easy to perform, with the only required inputs being the starting coordinates of the protein model and a table with the CSs.

A general script for an energy minimization using CSs as restraints is the following:

```
//Read force field and protein topology parameters
MDB = mdb.MDB("aa03.mdb");
//Build protein adding missing atoms, usually H-atoms,
//if necessary
PDB = pdb.pdb(sys.argv(1));
p = molecules.protein("GRD");
p.build_missing(PDB[0][0],MDB);
m = molecules.molecules();
m.add_protein(p);
//Setup restraints
//Create an empty collection of restraints/constraints
cc = const.collection();
//Add upper and lower distance restraints
upl_rest = const.upl(m,"dist.upl");
const.add_upl(cc,upl_rest);
lol_rest = const.lol(m,"dist.lol");
const.add_lol(cc,lol_rest);
//Setup chemical shift restraints
cs2 =const.camshift2(m,"camshift.db");
cs2.read_cs_table("experimental_CS.dat");
const.add_camshift2(cc,cs2);
//Force field setup
options = energy.amber_defaults();
//Perform minimization and save coordinates
minimize.const_sd(m,o,cc,MDB,5000,0.001);
molecules.molecules2pdb(m,"mini.pdb");
```

Benchmarks

Although the main purpose of the **Almost** project was not the development of a very fast molecular dynamics code, performance is an important factor in making many of the important biomolecular problems accessible for the all atom simulation protocols in a reasonable CPU timescale. We thus discuss here representative benchmarks that illustrate that **Almost** is, in the single CPU mode, as fast as widely used molecular codes such as NAMD 2.9, CHARMM 35b2, and GROMACS 4.6. Table 2 shows the time required to perform 20,000 steps of molecular dynamics at 300 K for a protein (ubiquitin) in vacuo. In all cases a cutoff of 12 Å was used for both electrostatic and van der Waals interactions. Simulation times where measured on a computer equipped with a 2.7 GHz Intel Core i7 processor running Mac OSX 10.8.4.

Almost is about 10% slower that GROMACS when using fast AVX vector extensions provided by modern CPUs. It should, however, be noted that while **Almost** uses a slightly more time-consuming smooth cutoff functions for electrostatics and van der Waals interactions (79 vs. 39 FLOPS per interaction),

Table 2. Time required to perform 20,000 steps of molecular dynamics at 300 K for ubiquitin in vacuo.

Ubiquitin in vacuum	**Almost** AVX	**Almost** SSE	**Almost** plain	NAMD	CHARMM	GROMACS
Simulation time (s)	58.29	79.69	235.50	131.43	180.08	52.81
Relative performance	1.10	1.50	4.45	2.49	3.40	1.00

GROMACS truncates the interactions at the cutoff distance, which is faster but may not conserve the energy.

Conclusions

We have described the molecular dynamics and modeling package **Almost**, which is an open source framework for the determination of the structure and dynamics of proteins and nucleic acids. **Almost** has been designed to provide support for a wide range of different structure determination protocols that make use of experimental observables as conformational restraints. Moreover, its object-oriented architecture can be used for the rapid development of further structural biology and bioinformatics applications.

Availability and Requirements

Project name: **Almost—**all atom molecular simulation toolkit.
Project home page: http://www.open-almost.org.
Operating systems: platform independent (Unix, Linux, MacOS X).
Programming languages: C++, C.
License: GNU Public License (GPL).
Any restriction to use by nonacademic: none.

Acknowledgments

*Over the years, numerous people have contributed to the development of **Almost** using it, finding bugs and suggesting improvements. The authors want to express their gratitude to all of them.*

Keywords: Molecular simulations · molecular dynamics · NMR spectroscopy · CHESHIRE · chemical shifts · residual dipolar couplings

How to cite this article: B. Fu, A. B. Sahakyan, C. Camilloni, G. G. Tartaglia, E. Paci, A. Caflisch, M. Vendruscolo, A. Cavalli *J. Comput. Chem.* **2014**, *35*, 1101–1105. DOI: 10.1002/jcc.23588

[1] B. Alberts, Molecular Biology of the Cell; Garland Publications, New York, **2008**.
[2] A. Fersht, Structure and Mechanism in Protein Science; Macmillan, **1999**.
[3] A. Rosato, J. M. Aramini, C. Arrowsmith, A. Bagaria, D. Baker, A. Cavalli, J. F. Doreleijers, A. Eletsky, A. Giachetti, P. Guerry, A. Gutmanas, P. Güntert, Y. He, T. Herrmann, Y. J. Huang, V. Jaravine, H. R. A. Jonker, M. A. Kennedy, O. F. Lange, G. Liu, T. E. Malliavin, R. Mani, B. Mao, G. T. Montelione, M. Nilges, P. Rossi, G. van der Schot, H. Schwalbe, T. A. Szyperski, M. Vendruscolo, R. Vernon, W. F. Vranken, S. de Vries, G. W. Vuister, B. Wu, Y. Yang, A. M. J. J. Bonvin, *Structure* **2012**, *20*, 227.
[4] A. T. Brünger, P. D. Adams, G. M. Clore, W. L. DeLano, P. Gros, R. W. Grosse-Kunstleve, J. S. Jiang, J. Kuszewski, M. Nilges, N. S. Pannu, R. J. Read, L. M. Rice, T. Simonson, G. L. Warren, *Acta Crystallogr. D Biol. Crystallogr.* **1998**, *54*, 905–921.
[5] A. Sali, T. L. Blundell, *J. Mol. Biol.* **1993**, *234*, 779.
[6] B. Brooks, R. Bruccoleri, B. Olafson, D. States, S. Swaminathan, M. Karplus, *J. Comp. Chem*. **1983**, *4*, 187.
[7] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435.
[8] A. -P. E. Kunz, J. R. Allison, D. P. Geerke, B. A. C. Horta, P. H. Hünenberger, S. Riniker, N. Schmid, W. F. van Gunsteren, *J. Comput. Chem.* **2012**, *33*, 340.
[9] D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, A. Onufriev, C. Simmerling, B. Wang, R. J. Woods, *J. Comput. Chem.* **2005**, *26*, 1668.
[10] C. D. Schwieters, J. J. Kuszewski, N. Tjandra, G. M. Clore, *J. Magn. Reson.* **2003**, *160*, 65.
[11] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y.-E. A. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popović, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker, P. Bradley, *Methods Enzymol.* **2011**, *487*, 545.
[12] G. Vriend, *J. Mol. Graph.* **1990**, *8*, 52.
[13] R. A. Laskowski, M. W. MacArthur, D. S. Moss, J. M. Thornton, *J. Appl. Crystallogr.* **1993**, *26*, 283.
[14] A. Cavalli, X. Salvatella, C. M. Dobson, M. Vendruscolo, *Proc. Natl. Acad. Sci. USA* **2007**, *32*, 204.
[15] A. Cavalli, R. W. Montalvao, M. Vendruscolo, *J. Phys. Chem. B* **2011**, *115*, 9491.
[16] J. Levine, Flex & Bison; O'Reilly Media Inc., San Francisco, CA, **2009**.
[17] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, J. R. Haak, *J. Chem. Phys.* **1984**, *81*, 3684.
[18] R. D. Skeel, J. A. Izaguirre, *Mol. Phys.* **2002**, *100*, 3885.
[19] A. Jain, N. Vaidehi, G. Rodriguez, *J. Comput. Phys.* **1993**, *106*, 258.
[20] D. Pearlman, D. Case, J. Caldwell, W. Ross, T. Cheatham, S. Debolt, D. Ferguson, G. Seibel, P. Kollman, *Comput. Phys. Commun.* **1995**, *91*, 1–41.
[21] W. Jorgensen, D. Maxwell, J. TiradoRives, *J. Am. Chem. Soc.* **1996**, *118*, 11225.
[22] D. Eisenberg, A. D. McLachlan, *Nature* **1986**, *319*, 199.
[23] N. Tjandra, J. G. Omichinski, A. M. Gronenborn, G. M. Clore, A. Bax, *Nat. Struct. Mol. Biol.* **1997**, *4*, 732.
[24] T. Lazaridis, M. Karplus, *Proteins* **1999**, *35*, 133.
[25] D. Bashford, D. A. Case, *Annu. Rev. Phys. Chem.* **2000**, *51*, 129.
[26] K. J. Kohlhoff, P. Robustelli, A. Cavalli, X. Salvatella, M. Vendruscolo, *J. Am. Chem. Soc.* **2009**, *131*, 13894.
[27] A. B. Sahakyan, W. F. Vranken, A. Cavalli, M. Vendruscolo, *J. Biomol. NMR* **2011**, *50*, 331.
[28] A. B. Sahakyan, W. F. Vranken, A. Cavalli, M. Vendruscolo, *Angew. Chem. Int. Ed. Engl.* **2011**, *50*, 9620.
[29] F. Förster, B. Webb, K. A. Krukenberg, H. Tsuruta, D. A. Agard, A. Sali, *J. Mol. Biol.* **2008**, *382*, 1089.
[30] R. W. Montalvao, A. Cavalli, X. Salvatella, T. L. Blundell, M. Vendruscolo, *J. Am. Chem. Soc.* **2008**, *130*, 15990.
[31] K. Wüthrich, NMR of Proteins and Nucleic Acids; Wiley-Interscience, **1986**.
[32] P. Güntert, *Eur. Biophys. J.* **2009**, *38*, 129.
[33] T. Herrmann, P. Güntert, K. Wüthrich, *J. Mol. Biol.* **2002**, *319*, 209.
[34] P. Güntert, *Methods Mol. Biol.* **2004**, *278*, 353.
[35] P. Robustelli, K. Kohlhoff, A. Cavalli, M. Vendruscolo, *Structure* **2010**, *18*, 923.
[36] P. Neudecker, P. Robustelli, A. Cavalli, P. Walsh, P. Lundström, A. Zarrine-Afsar, S. Sharpe, M. Vendruscolo, L. E. Kay, *Science* **2012**, *336*, 362.
[37] M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Martinelli, F. Pietrucci, R. A. Broglia, M. Parrinello, *Comput. Phys.* **2009**, *180*, 1961.
[38] C. Camilloni, P. Robustelli, A. De Simone, A. Cavalli, M. Vendruscolo, *J. Am. Chem. Soc.* **2012**, *134*, 3968.
[39] C. Camilloni, A. Cavalli, M. Vendruscolo, *J. Phys. Chem. B* **2013**, *117*, 1838.

Received: 14 January 2014
Revised: 16 February 2014
Accepted: 2 March 2014
Published online on 27 March 2014