PHAISTOS: A Framework for Markov Chain Monte Carlo Simulation and Inference of Protein Structure

Wouter Boomsma,[a,b]* Jes Frellsen,[a] Tim Harder,[a,c] Sandro Bottaro,[d,e] Kristoffer E. Johansson,[a] Pengfei Tian,[f] Kasper Stovgaard,[a] Christian Andreetta,[a,g] Simon Olsson,[a] Jan B. Valentin,[a] Lubomir D. Antonov,[a] Anders S. Christensen,[h] Mikael Borg,[a,i] Jan H. Jensen,[h] Kresten Lindorff-Larsen,[a] Jesper Ferkinghoff-Borg,[e] and Thomas Hamelryck[a]

We present a new software framework for Markov chain Monte Carlo sampling for simulation, prediction, and inference of protein structure. The software package contains implementations of recent advances in Monte Carlo methodology, such as efficient local updates and sampling from probabilistic models of local protein structure. These models form a probabilistic alternative to the widely used fragment and rotamer libraries. Combined with an easily extendible software architecture, this makes PHAISTOS well suited for Bayesian inference of protein structure from sequence and/or experimental data. Currently, two force-fields are available within the framework: PROFASI and OPLS-AA/L, the latter including the generalized Born surface area solvent model. A flexible command-line and configuration-file interface allows users quickly to set up simulations with the desired configuration. PHAISTOS is released under the GNU General Public License v3.0. Source code and documentation are freely available from http://phaistos.sourceforge.net. The software is implemented in C++ and has been tested on Linux and OSX platforms. © 2013 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23292

Introduction

Two methods dominate the field of molecular simulation: molecular dynamics (MD) and Markov chain Monte Carlo (MCMC). The main difference between the methods lies in the way the system is updated in each iteration. MD involves iterating between calculating the forces exerted on each particle in the system and using Newton's equations of motion to update their positions. In contrast, MCMC is a statistical approach, where the goal is to generate samples from a probability distribution associated with the system, typically a Boltzmann distribution. MD has generally been regarded as best-suited for exploring dense molecular systems such as the native ensemble of proteins, while MCMC methods can be more efficient for longer time scale simulations involving large structural rearrangements.[1] Using optimized move sets it has, however, been demonstrated that even in the densely packed native state, MCMC can serve as an efficient alternative to MD.[2–4] In addition, the statistical nature of MCMC methods make them particularly well-suited for Bayesian inference of protein structure from experimental data.[5]

The freedom in the choice of moves in Monte Carlo simulations means that there is potential progress to be made in designing new, improved move types, thereby further increasing the time scales and molecular sizes amenable to simulation. In this article, we present a software framework designed with this goal in mind. The PHAISTOS framework contains implementations of recently developed tools that increase the

[a] *W. Boomsma, J. Frellsen, T. Harder, KE. Johansson, K. Stovgaard, C. Andreetta, S. Olsson, JB. Valentin, L. D. Antonov, M. Borg, K. Lindorff-Larsen and T. Hamelryck*
Department of Biology, University of Copenhagen, Copenhagen, 2200, Denmark
E-mail: wb@bio.ku.dk

[b] *W. Boomsma*
Department of Astronomy and Theoretical Physics, University of Lund, Lund, SE-223 62, Sweden

[c] *T. Harder*
Center for Bioinformatics, University of Hamburg, Hamburg, 20146, Germany

[d] *S. Bottaro*
Scuola Internazionale Superiore di Studi Avanzati, Trieste, 34136, Italy

[e] *S. Bottaro and J. Ferkinghoff-Borg*
Department of Biomedical Engineering, DTU Elektro, DTU, Kongens Lyngby, 2800, Denmark

[f] *P. Tian*
Niels Bohr Institute, University of Copenhagen, Copenhagen, 2100, Denmark

[g] *C. Andreetta*
Computational Biology Unit, Uni Computing, Uni Research, Norway

[h] *AS. Christensen and JH. Jensen*
Department of Chemistry, University of Copenhagen, Copenhagen, 2100, Denmark

[i] *M. Borg*
BILS, Science for Life Laboratory, Box 1031, Solna, 171 21, Sweden

Contract/grant sponsor: Danish Council for Independent Research; Contract/grant numbers: FNU272-08-0315 (to W.B.); FTP274-06-0380 (to K.S.); FTP09–066546 (to S.O. and J.V.), and FTP274-08-0124 (to K.E.J.).
Contract/grant sponsor: Danish Council for Strategic Research; Contract/grant number: NABIIT2106-06-0009.
Contract/grant sponsors: Novo Nordisk STAR Program (to A.S.C.), Novo Nordisk Foundation (to K.L.L.), and Radiometer (DTU) (to S.B.).

efficiency and scope of MCMC-based simulations. Through a modular design, the software can easily be extended with new move types and force-fields, making it possible to experiment with novel Monte Carlo strategies. Finally, using flexible configuration file and command line options, users can quickly set up simulations with any combination of moves, energy terms, and other simulation settings.

By making our methods available in an easily extendible, open source framework, we hope to further encourage the use of MCMC for protein simulations and promote the development of new MCMC methodologies for the simulation, prediction, and inference of protein structure.

Methodology

The framework is split into four main types of components: moves, energy terms, observables, and Monte Carlo methods. For each of these types, a number of algorithms are available. Moves and energies are normally used in sets: a weighted set of moves is referred to as a move collection, while an energy function is composed of a weighted sum of energy terms. Observables are similar to energy terms, but are typically only evaluated at certain intervals to extract statistics during a simulation. In the following description, each algorithm is annotated with its corresponding command line option name in a monospace font.

Moves

One of the main distinguishing features of the PHAISTOS package is efficient sampling, obtained through an elaborate set of both established and novel Monte Carlo moves. Each move stochastically modifies a protein chain in a specific way. Weighted sets of these moves can be selected from the command line, allowing the user to easily experiment and fine-tune the set of moves for a given simulation scenario. All moves in PHAISTOS can be applied such that detailed balance is obeyed, which ensures, if the sampling is ergodic, that simulations sample from a well-defined target distribution (e.g., the canonical or multicanonical ensemble).

The framework contains many of the established moves from the literature, including various pivot moves (`move-pivot-uniform`, `move-pivot-local`), the crankshaft/backrub local move (`move-crankshaft`),[6,7] the CRA local move (`move-cra`),[8] and the semilocal biased Gaussian step (BGS) (`move-semilocal`).[9] Side-chain conformational sampling can be done either from Gaussian distributions given by rotamer libraries (`move-sidechain-rotamer`)[10] or through Gaussians centered around the current side-chain conformation (`move-sidechain-local`).

Moves using Probabilistic Models. PHAISTOS has broad support for sampling using biased proposals. Usually, if an MCMC simulation were to be conducted without the presence of a force-field, a uniform distribution in configurational space would be obtained. In the case of biased sampling, moves are instead allowed to follow a specific distribution during the simulation. This bias can then, optionally, be divided out, so that it does not influence the final statistical ensemble, but only serves to increase sampling efficiency by focusing on the most important regions of conformational space. If the bias is left in, it corresponds to an implicit extra term in the energy function.

Typically, the bias is chosen to reflect prior knowledge about the local structure of the molecule. A good example is the common use of fragment and rotamer libraries for structure prediction.[11,12] These methods are used strictly for sampling, and the introduced bias is not easily quantifiable, which also makes it difficult to ensure detailed balance for the Markov chain. In contrast, PHAISTOS includes a number of moves based on probabilistic models, which support both sampling of conformations and the evaluation of the bias introduced with those moves. This makes them uniquely suited for use in MCMC simulations.

Four different structural, probabilistic models are available: FB5HMM models the Cα trace of a protein,[13] COMPAS models a reduced single-particle representation of amino acid side-chains, whereas TORUSDBN and BASILISK, respectively, model backbone and side-chain structure in atomic detail.[14,15] All models can be applied both as proposal distributions in the form of Monte Carlo moves (`move-backbone-dbn`, `move-sidechain-basilisk`, `move-sidechain-compas`) and as probabilistic components of an energy function (`energy-backbone-dbn`, `energy-basilisk`, `energy-compas`). Figure 1 illustrates how dihedral angles are sampled from a TORUSDBN-like model of the protein backbone. The practical details on how probabilistic models can be incorporated in an energy function are discussed in the "Energies" section.

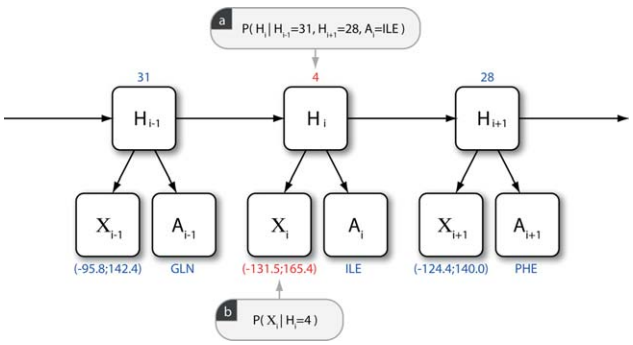

Figure 1. An illustration of a simplified version of the TORUSDBN model of backbone local structure, showing the architecture of the dynamic Bayesian network (DBN) and an example of values for the individual nodes. Each *A* node is a discrete distribution over amino acids, whereas each *X* node is a bivariate distribution over (ϕ, ψ) angle pairs. The hidden node (*H*) sequence is a Markov chain of discrete states, representing the sequence of residues in a protein chain. Each hidden node state corresponds to a particular distribution over angle pairs and amino acid labels. The values highlighted in red are the result of a single resampling step of the (ϕ, ψ) angle pair at some position *i* in the chain: a) The hidden node state H_i is resampled based on the current values of the values of neighboring *H* values and the amino acid label at position *i* ($P(H_i|H_{i-1}, H_{i+1}, A_i) \propto P(H_i|H_{i-1})P(H_{i+1}|H_i)P(A|H_i)$); b) A (ϕ, ψ) value is drawn from the bivariate angular distribution corresponding to the sampled *H* value ($P(X_i|H_i)$). A full description of the TORUSDBN model can be found in the original publication.[14]

Efficient Local Updates. An important challenge in Monte Carlo simulations is to ensure efficient sampling in dense states where proposed conformational changes will have a high probability of containing self-collisions. In particular, pivot moves will typically have very poor acceptance rates in this scenario. The solution is typically to expand the move set to include local moves, which only change the atom positions within a small segment of the chain.

In addition to various established local move methods from the literature, PHAISTOS includes a novel method, called CRISP[4] (`move-crisp`), which is particularly well-suited for this problem. Unlike other local move approaches,[7,8] CRISP is able to generate updates to a segment of the chain without disrupting its local geometry. Often, local move algorithms are designed as a two-step process, where some angular degrees of freedom are modified stochastically (prerotation), whereas others are modified deterministically to bring the chain back to a closed state (postrotation). From the work of Gō and Scheraga,[16] it is known that in general, six degrees of freedom are required for the postrotation step. CRISP distinguishes itself from previous methods by merging these two steps, modifying the stochastic prerotation step so that it takes the resulting postrotation step into account. More precisely, for each application of the move, a random segment of the protein is selected, and a multivariate Gaussian distribution is constructed over the angular change in the $n-6$ prerotation degrees of freedom $\delta\bar{\chi}_{\text{pre}}$

$$P(\delta\bar{\chi}_{\text{pre}}) \propto \exp\left(-\frac{1}{2}\delta\bar{\chi}_{\text{pre}}^{T}\lambda(\mathbf{C}_{n-6}+\mathbf{S}^{T}\mathbf{C}_{6}\mathbf{S})\delta\bar{\chi}_{\text{pre}}\right). \quad (1)$$

Here, $\mathbf{C}$ is an inverse diagonal covariance matrix specifying the desired fluctuations for the individual angular degrees of freedom, $\mathbf{S}$ is a linear transformation mapping the prerotational degrees of freedom to the corresponding postrotational values, and λ is a scaling parameter determining the size of the move. In effect, to first order, the method samples from a distribution of closed chain structures, ensuring high-quality local structure in all samples. We have recently shown that this has a dramatic impact on simulation performance, in particular for dense molecular systems.[4]

Low acceptance rates are also sometimes observed in side-chain moves. When using a fine-grained force-field such as OPLS-AA/L, we have experienced that the standard resampling of side-chains can be overly intrusive. Particularly in the case of side-chains involved in several simultaneous hydrogen bonds, traditional moves tend to break all hydrogen bonds at once, typically leading to the rejection of such updates. To avoid this problem, PHAISTOS includes a novel move (`side-chain-local`) that, for a given side-chain, randomly selects an atom that potentially participates in hydrogen bonds, and constrains its position using a technique similar to that of the semilocal BGS backbone move.[4,9]

Energies

Two established force-fields are currently implemented within the framework: the PROFASI force-field[17] and the OPLS-AA/L[18] force-field in combination with the generalized Born surface area (GB/SA) implicit solvent model.[19] These represent two extremes in the range of force-fields available in the literature: an ultrafast force-field modeling effective interactions in the presence of a solvent and a classic fine-grained molecular mechanics force-field combined with a more accurate implicit solvent model. The two force-fields were selected to provide support for a broad range of simulation tasks. The efficiency of the PROFASI force-field makes it possible readily to conduct reversible folding simulations of peptides and small proteins.[17] The OPLS-AA/L force-field in combination with the GB/SA solvent model is more accurate, but also significantly slower, and is typically used for exploring the details of native ensembles. It can also be used for structure refinement, for instance of structures obtained in a reversible folding simulation using PROFASI. For increased efficiency, all nonbonded force-field terms in both force-fields have been implemented using the chaintree data structure,[20] which avoids recalculation of energy contributions that are not modified in a given iteration of the simulation. Together with effective local moves, this can result in a considerable computational speed-up.

PROFASI. The PROFASI force-field consists of four terms[17]

$$E = E_{\text{ev}} + E_{\text{hb}} + E_{\text{sc}} + E_{\text{loc}} \quad (2)$$

where E_{ev} captures excluded volume effects, E_{hb} is a hydrogen bond term, E_{sc} is a side-chain interaction term, and E_{loc} concerns the local interactions along the chain. The excluded volume potential is a simple r^{-12} interaction between all atom pairs, where r denotes the distance between the atoms. The strength of a hydrogen bond in PROFASI depends on the detailed geometry of the bond, parameterized through the N—H—O and H—O—C angles. The side-chain potential consists of a charge–charge and a hydrophobicity contribution. For each residue pair, these consist of a product between a conformation-dependent contact strength and an energy that depends on the specific amino acid types involved in the bond. Finally, the local energy term captures interactions between partial charges in neighboring peptide units along the chain, with a correction term for improved consistency with the Ramachandran plot, and a side-chain torsion potential. As bond angles and bond lengths are assumed fixed during PROFASI simulations, no further local interactions are included.

A distinguishing feature of the PROFASI force-field is the presence of a global interaction cutoff of 4.5 Å. Although this necessarily excludes various long-range interactions, it is also one of the main reasons behind the efficiency of the force-field. Despite this restriction, the force-field has been demonstrated to successfully fold a range of peptides and small proteins,[17] while still being fast enough for many-body aggregation simulations.[21,22]

OPLS-AA/L. In contrast to PROFASI, the OPLS-AA/L force-field includes local terms for bond angles, bond lengths, and torsions. The bond angle and bond length potentials are simple harmonic terms, whereas the torsion term has the form[18]

$$E_{\text{torsion}} = \sum_i \sum_{j=1}^{3} w_j \left(1 + (-1)^{j+1} \cos(j\theta_i)\right) \quad (3)$$

where the outer sum iterates over all dihedrals θ_i. The non-bonded interactions include standard Lennard-Jones and Coulomb potentials

$$E_{\text{nb}} = \sum_{i>j} w_{ij} \left(4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right) + \frac{1}{4\pi\varepsilon_0} \frac{q_i q_j}{r_{ij}}\right) \quad (4)$$

where r_{ij} is the distance between atoms i and j, q_i and q_j are the corresponding partial charges, ε_0 is the vacuum permittivity, and σ_{ij} and ϵ_{ij} are calculated using the combination rules $\sigma_{ij} = \sqrt{\sigma_i \sigma_j}$ and $\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}$, respectively. Finally, w_{ij} works to exclude interactions between atoms that are separated by only a few covalent bonds. Thus, $w_{ij} = 0.0$ for direct neighbors (1,2) and pairs separated by a single other atom (1,3), $w_{ij} = 0.5$ for pairs separated by two atoms (1,4), and $w_{ij} = 1.0$ for all others.

Our implementation of OPLS-AA/L follows that of the Tinker simulation package.[23] We ensured that energies produced by our program match those obtained when running Tinker.

GB/SA. The PROFASI force-field is parameterized to capture effective interactions in the presence of a solvent. In contrast, OPLS-AA/L should be combined with a suitable solvent model to reproduce physiological conditions. To model the effect of the solvent on hydrophobic interactions and electrostatics, we use the OPLS-AA/L force-field in combination with the GB/SA implicit solvent model.[19]

Many implicit solvent models express the solvation free energy G_{solv} as a sum of nonpolar and electrostatic contributions

$$G_{\text{solv}} = G_{\text{npol}} + G_{\text{pol}} \quad (5)$$

Here, G_{npol} is the free energy of solvating the molecule with all the partial charges set to zero, and G_{pol} is the reversible work required to increase the charges from zero to their full values.[24] In GB/SA, the nonpolar contribution G_{npol} is assumed to be proportional to the solvent accessible surface area, while the generalized Born approximation is used to calculate the electrostatic solvation energy using the pairwise summation[25]

$$G_{\text{pol}} = -\frac{1}{8\pi\varepsilon_0}\left(1 - \frac{1}{\varepsilon}\right) \sum_{i,j}^{n} \frac{q_i q_j}{f_{\text{GB}}} \quad (6)$$

where ε is the dielectric constant of the solvent, and q_i is the partial charge of atom i. $f_{\text{GB}} = \sqrt{r_{ij}^2 + \alpha_i \alpha_j \exp\left(-r_{ij}^2 / 4\alpha_i \alpha_j\right)}$ is a function of the distance r_{ij} and of the so-called Born radii α, which reflects the average distance of the charged atom to the dielectric medium. For our implementation, the Born radii are calculated using an analytical expression proposed by Still and coworkers.[19]

Incorporating Probabilistic Models in the Energy Function. When using moves that are based on probabilistic models such as TORUSDBN and BASILISK, it gives rise to a bias in the simulation, which can be regarded as an implicit energy term. In PHAISTOS, the energy contributions of these probabilistic models can also be evaluated explicitly, by adding them as a term to the energy function. This makes it possible to use the probabilistic models as energies in a simulation with a standard set of unbiased moves, or to compensate for the bias of a move by adding the corresponding energy term with negative weight. When used as energies, the values are reported in minus log-probabilities. To facilitate the combination of classic energy terms with probabilistic terms, the energies of physical force-fields such as PROFASI and OPLS-AA/L are likewise reported as minus log-probabilities: they are multiplied by $-1/kT$, where T is the simulation temperature and k is the Boltzmann constant.

As an example, for the TORUSDBN-like model in Figure 1, the log-likelihood for a given state is

$$\begin{aligned} LL(\bar{X}, \bar{A}) &= \ln \sum_{\bar{H}} P(\bar{X}, \bar{A}, \bar{H}) \\ &= \ln \sum_{\bar{H}} P(X_1|H_1) P(A_1|H_1) P(H_1) \prod_{i=2}^{N} P(X_i|H_i) P(A_i|H_i) P(H_i|H_{i-1}) \end{aligned} \quad (7)$$

where $\bar{X}$, $\bar{A}$, and $\bar{H}$ are the sequences of angle pairs, amino acid labels, and hidden node labels, respectively, i is the residue index, and N is the sequence length. Each hidden node label is the index of a component of the emission distributions of the model. For instance, the Ramachandran distribution is modeled as a weighted sum of bivariate von Mises distribution components.[14] The hidden nodes are "nuisance" parameters and are therefore summed out in the evaluation of the likelihood. Note that the sum runs over all possible hidden node *sequences*, a calculation that can be done efficiently using dynamic programming.[14]

Observables

Observables in PHAISTOS allow a user to extract information about the current state of a simulation. Examples of observables include root-mean-square-deviation (RMSD) (`observable-rmsd`) and radius of gyration (`observable-rg`). In addition, all energy terms are also available as observables. A user can specify a selection of observables from the command line or settings file, choosing how frequently they should be registered and in which format. While most observables will return a single value, others have more elaborate outputs, such as dumping of complete structural states to PDB files (`observable-pdb`) or to a molecular trajectory file in the Gromacs XTC format (`observable-xtc-trajectory`).[26] Finally, observables can be dumped to the header or as *b*-factors in outputted PDB-files. The latter makes it possible to annotate structures with residue-specific information, such as the number of contacts, degree of burial, and more sophisticated evaluations of the environment of each residue.[27]

Monte Carlo

Although PHAISTOS can be used for Monte Carlo minimization, the primary focus of the framework is MCMC simulation,

where the goal is to produce samples from the Boltzmann distribution corresponding to a given force-field at a specified temperature. All moves in PHAISTOS are, therefore, designed to be compatible with the property of detailed balance, in the sense that their proposal probabilities can be evaluated. That is, for a move from state x to state x'

$$\pi(x)P(x \to x') = \pi(x')P(x' \to x) \tag{8}$$

where $\pi(x)$ is the stationary distribution, and $P(x \to x')$ is the probability of moving from state x to x' using a given move. Factoring $P(x \to x')$ into a "selection" probability P_s and an "acceptance" probability P_α, we have

$$\frac{P_\alpha(x \to x')}{P_\alpha(x' \to x)} = \frac{\pi(x')P_s(x' \to x)}{\pi(x)P_s(x \to x')} \tag{9}$$

Most of the moves are symmetric, in the sense that $P_s(x' \to x)/P_s(x \to x') = 1$. However, for moves such as the local and semilocal moves, this is not the case, and it is important that this bias be correctly compensated for. Implementation-wise, each Move object is responsible for calculating the bias that it introduces, and the Monte Carlo class will then compensate for it when necessary.

Metropolis-Hastings. The most common way to ensure that eq. (9) is fulfilled is to use the Metropolis-Hastings (MH) acceptance criterion

$$P_\alpha(x \to x') = \min\left(1, \frac{\pi(x')P_s(x' \to x)}{\pi(x)P_s(x \to x')}\right) \tag{10}$$

This is the default simulation method used in PHAISTOS (`monte-carlo-metropolis-hastings`). It is useful for exploring near-native ensembles and can be efficient when simulating at the critical temperature of a system. However, for more complicated systems, MH simulations tend to spend excessive periods of time exploring local minima, leading to poor mixing, and, therefore, slow convergence.

Generalized Ensembles. To avoid the mixing problems associated with standard MH simulations, PHAISTOS includes support for conducting simulations in generalized ensembles.[28] Rather than sampling directly from the Boltzmann distribution, the central idea is to generate samples from a modified distribution, and subsequently reweight the obtained statistics to the Boltzmann distribution at a desired temperature. The acceptance criterion becomes

$$P_\alpha(x \to x') = \min\left(1, \frac{w(x')P_s(x' \to x)}{w(x)P_s(x \to x')}\right) \tag{11}$$

for a given weight function $w(x)$. The typical choice is the multicanonical ensemble,[29] corresponding to a flat distribution over energies. That is, $w(x) = 1/g(E(x))$, where $E(x)$ is the energy associated with conformational state x, and g is the number of states associated with a given energy (density of states). Another example is the $1/k$ ensemble, which attempts to provide ergodic sampling while maintaining primary focus on the low energy states.[30] In this case, the weight function is $w(x) = 1/k(E(x))$, where $k(E(x)) = \int_{-\infty}^{E(x)} g(\hat{E})\,d\hat{E}$. It can be shown that this is approximately equivalent to a flat histogram over $\ln(E(x))$.[30]

We have recently developed an automated method, MUNINN, for estimating the weights w in generalized ensemble simulations (http://muninn.sourceforge.net/). It employs the generalized multihistogram equations,[31] and uses a non-uniform adaptive binning of the energy space, ensuring efficient scaling to large systems. In addition, MUNINN allows weights to be restricted to cover a limited temperature range of interest. The MUNINN functionality is seamlessly integrated into PHAISTOS and can be activated by selecting the corresponding Monte Carlo engine (`monte-carlo-muninn`).

Monte Carlo Minimization. PHAISTOS contains a few simulation algorithms that are directed at optimization, rather than sampling. These include a simulated annealing class (`monte-carlo-simulated-annealing`) and a greedy Monte Carlo optimization class (`monte-carlo-greedy-optimization`), which are useful in cases where the user is interested in a single low-energy structure, rather than a full structural ensemble.

Program design

The framework is designed to be modular, both in software design and in the choices exposed to the user from the command line or settings file. As illustrated in the UML diagram in Figure 2, all energy terms are derived from the same base class and implement the same interface. Energy functions can easily be constructed from the command line or settings file by including the energy terms of interest. Moves and Monte Carlo simulation algorithms are structured in a similar way. This design makes it straightforward to implement new energy terms, moves or simulation algorithms with little knowledge of the overall code. Iterators are provided for easy iteration over atoms or residues in a molecule. In addition, caching and rapid determination of interacting atom pairs is made possible by an implementation of the chaintree algorithm.[20]

Finally, through a modular build-system, developers can readily write their own modules utilizing the library. Modules are separate code entities that are autodetected by the build system when present and can be enabled and disabled at compile time, making it easy to share code among collaborators.

Example

We include a step-by-step walk-through of the PHAISTOS simulation process. The goal is to conduct a reversible folding simulation of the 20-residue beta3s peptide,[32] demonstrating several of the features described earlier.

The user interface of PHAISTOS is designed to make it as easy as possible to set up simulations. Almost all options have default values, and it is, therefore, usually sufficient to supply only a few input options to the program. The program behavior can then gradually be fine-tuned using additional options in the configuration file later on. For this particular example, we use the following command from the command line

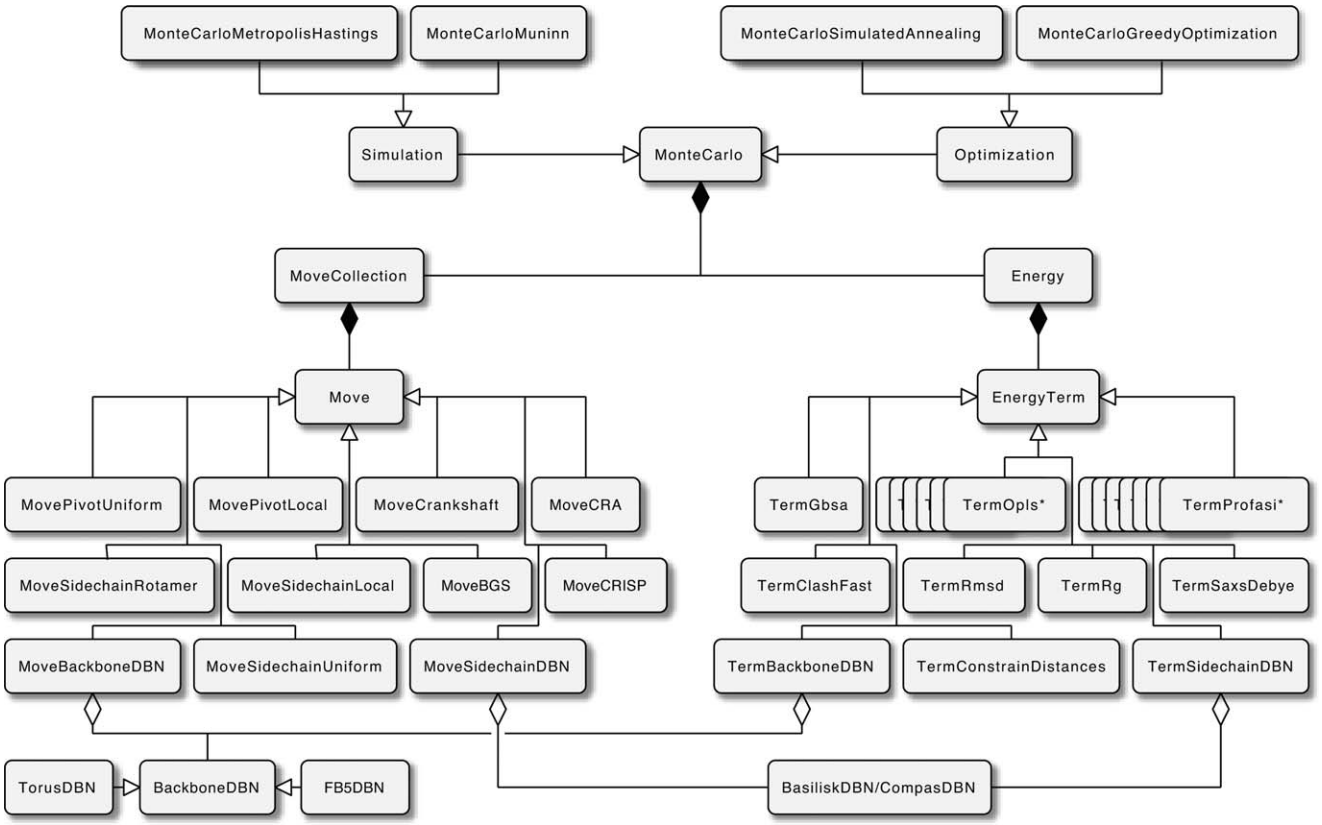

Figure 2. A UML-diagram of the major classes in the PHAISTOS library (black diamond: composition, white diamond: aggregation, arrow: inheritance). A Monte Carlo simulation object contains a MoveCollection object, which consists of a selection of moves, and an Energy object, which comprises a number of energy terms (TermOpls* and TermProfasi* denote the entire set of OPLS and PROFASI energy terms, respectively). Note that the probabilistic models (BackboneDBN/BasiliskDBN/CompasDBN) are available both as energy terms and as moves. A detailed description of all classes can be found in the Doxygen documentation on the PHAISTOS web site.

```
$ ./phaistos --aa-file beta3s.aa \
             --energy profasi-cached backbone-dbn [weight:-1] \
             --move backbone-dbn sidechain-uniform semilocal-dbn-eh \
             --threads 8 --temperature 283 \
             --monte-carlo muninn[min-beta:0.6,max-beta:1.1] \
             --observable backbone-dbn rmsd[reference-pdb-file:beta3s. pdb] \
             --observable xtc-trajectory
```

The aa-file argument specifies that we are reading an amino acid sequence from a file. The energy and move options select the relevant energy terms and moves, respectively. In this case, we use a cached version of the PROFASI force-field, and sample using TORUSDBN moves, uniformly distributed side-chain moves, and BGS moves using the TORUSDBN as a prior. We specify that the simulation should be conducted in eight parallel threads and set the temperature to 283 K. MUNINN is chosen to be the Monte Carlo engine, using a β (inverse temperature) range of $[0.6; 1.1]$. These β factors are unit-less, specified relative to the inverse temperature, and thus correspond to a temperature range of $\left[(1.1 \cdot 1/283\text{K})^{-1}; (0.6 \cdot 1/283\text{K})^{-1}\right] = [257\text{K}; 472\text{K}]$. Finally, we specify that we wish to record observables about the backbone-dbn energy, the RMSD to the native state, and dump structures to an XTC trajectory file. Apart from the RMSD observable, we do not provide the program with any information about the structure of the protein, and the simulation will, therefore, start in a random extended state.

To illustrate the framework's support for various types of biased sampling, this example uses a variant where the TORUSDBN bias is included in the sampling, and explicitly subtracted in the energy (i.e., the weight: -1 option of backbone-dbn). This means that the bias cancels out when extracting statistics at $\beta = 1$, thus producing unbiased estimates at 283K. The simulation will produce a flat histogram over the expected energy range corresponding to the specified temperature range ($[\langle E \rangle_{257\text{K}}; \langle E \rangle_{472\text{K}}]$).

We ran PHAISTOS with the settings above on an eight-core 3.4 GHz Intel Xeon processor for 1 week. Figure 3 gives an overview of the results. The free energy plot in Figure 3a shows the distribution of energy versus RMSD extracted directly from the samples dumped during the simulation. As the samples were generated using a generalized ensemble

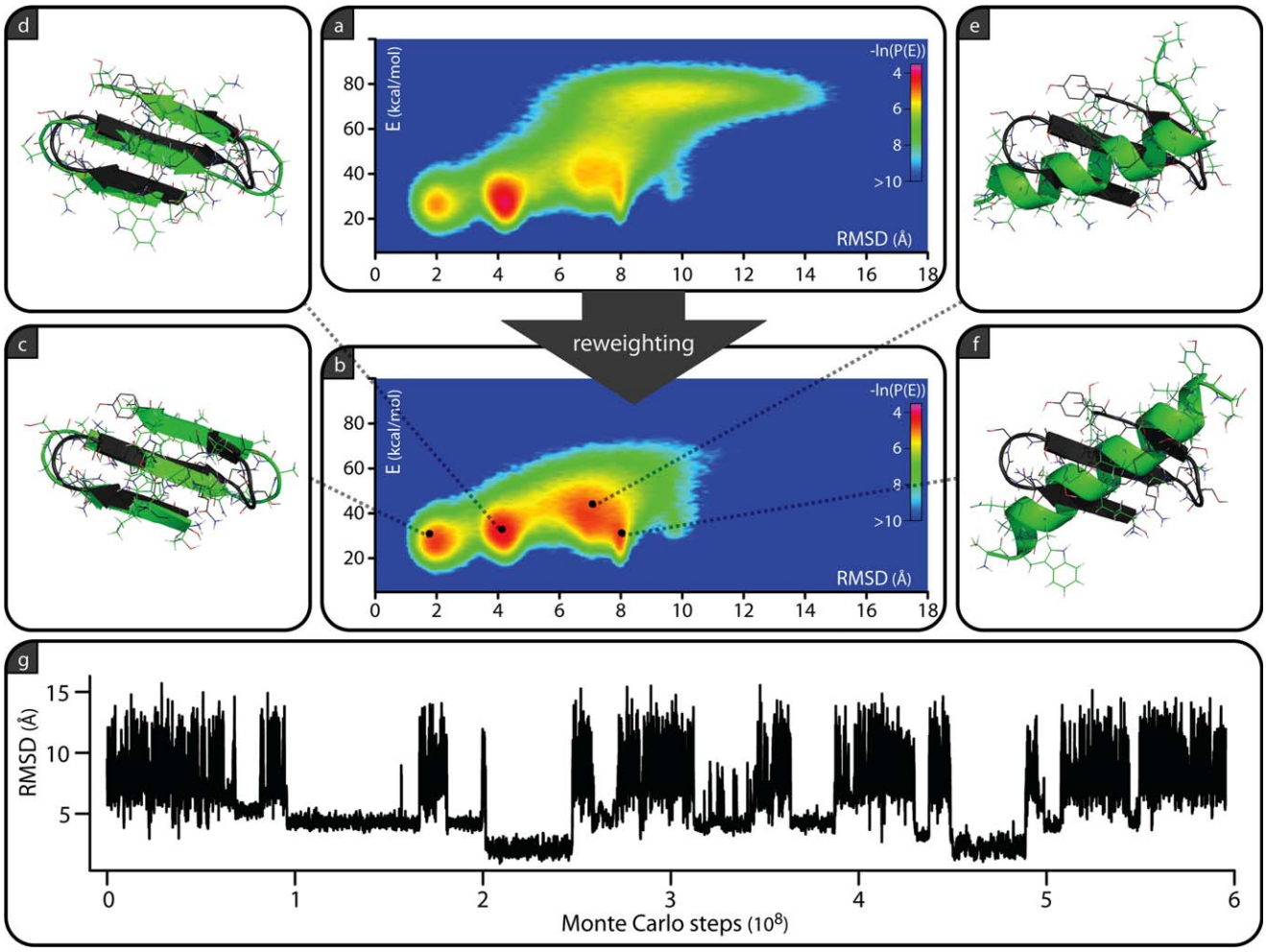

Figure 3. Illustration of a reversible folding simulation of the beta3s peptide in PHAISTOS. The simulation was conducted with the PROFASI force-field, using the MUNINN multihistogram method and a set of moves including TORUSDBN as a dihedral proposal distribution. The bias introduced by TORUSDBN is compensated for to ensure correctly distributed samples. a) Free energy plot as a function of energy and RMSD in the multicanonical (flat histogram) ensemble. b) Free energy plot as a function of energy and RMSD, reweighted to the canonical ensemble at 283 K. c–f) Representative cluster medoids found with reweighted clustering using the PLEIADES module, compared to the native structure[32] (shown in black). Figures created using Pymol.[33] g) RMSD versus time of one of the eight threads in the simulation.

technique, they must be reweighted to retrieve the statistics according to the Boltzmann distribution at the specified temperature. This is done using a script included in the MUNINN module, resulting in the plot in Figure 3b.

To find representative structures in the ensemble, we use the PLEIADES clustering module,[34] included in the framework. Again, it is important to remember that the raw data are produced in a generalize ensemble setting and must be reweighted. We ran the RMSD-based weighted *k*-means method implemented in the PLEIADES module to select the highlighted structures in Figure 3.

From the analysis above, we conclude that at 283 K, the protein is marginally stable in the PROFASI force-field, with native-like populations at 2 and 4 Å RMSD, but also a significant population of unstructured or helical conformations. These results are in approximate agreement with experiments, which suggest a folded population of between 13 and 31%.[32] This result is compatible with a previously published simulation of the same protein using an unbiased simulation technique.[17]

Results

To illustrate the versatility of PHAISTOS, we highlight several recently published applications of the framework.

Structure prediction and inference

An example of the applicability of PHAISTOS in the context of protein structure prediction is found in a recent study on potentials of mean force.[35] The study demonstrates how probabilistic models of local protein structure such as TORUSDBN and BASILISK can be combined with probabilistic models of nonlocal features, such as hydrogen bonding and compactness, using a simple probabilistic technique.

The framework has also been applied for inference of protein structure from small-angle X-ray scattering (SAXS) and nuclear magnetic resonance (NMR) experimental data. SAXS data contain low-resolution information on the overall shape of a protein, which can be useful for determining the relative domain positions and orientations in multidomain proteins or complexes.

This can for instance be used to infer structural models of multi-domain proteins connected by flexible linkers, given the atomic structures of the individual domains. Such calculations require efficient back-calculation of SAXS curves, which is made possible through a coarse-grained Debye method.[36,37]

NMR experimental data can provide high-resolution structural information that can improve the accuracy of a simulation. PHAISTOS contains preliminary support for sampling conditional on chemical shift data, which is known to contain substantial information on the local structure of a protein.[38,39] Furthermore, the framework was recently used for inferential structure determination using pair-wise distances obtained from NOE experiments, with TORUSDBN and BASILISK as prior distribution for the protein's backbone and side chains.[40]

Efficient clustering

Efficiently clustering a large number of protein structures is an important task in protein structure prediction and analysis. Typically, clustering programs require costly RMSD calculations for many pairs in the set of structures. PHAISTOS contains a clustering module called PLEIADES that uses a *k*-means clustering approach[41] to reduce the number of pair-wise RMSD distance calculations. Furthermore, PLEIADES includes support for replacing the RMSD distance computations with distances between vectors of Gauss integrals,[42] which provides dramatic computational speedups.[34]

Native ensembles

The energy landscape around the native state tends to be rugged, making it challenging to sample such states efficiently.[1] For these tasks, the CRISP backbone move is particularly well suited, given its ability to propose subtle, nondisruptive updates to the protein backbone. Monte Carlo simulations using this move were recently shown to explore conformational space with an efficiency on par with MD, outperforming the current state-of-the-art in local Monte Carlo move methods.[4]

The TYPHON module[43] rapidly explores near-native ensembles using the CRISP move in combination with a user-defined set of nonlocal restraints. Local structure is under the control of probabilistic models of the backbone (TORUSDBN) and side chains (BASILISK), while nonlocal interactions such as hydrogen bonds and disulfide bridges are heuristically imposed as Gaussian restraints. TYPHON can be seen as a "null model" of conformational fluctuations in proteins: it rapidly explores the conformational space accessible to a protein given a set of specified restraints.

Discussion

The relevance of a new software package should be assessed relative to already existing packages in the literature. We acknowledge that in our case, there are a number of such alternatives already available. We describe the most important ones here, focusing on the differences to the framework presented in this article.

Of the available Monte Carlo software packages, the ROSETTA package[11] is perhaps the most widely used and has an impressive track record for protein structure prediction and design.[44] The package focuses primarily on structure/sequence prediction (optimization) rather than simulation, and consequently, many of the moves in ROSETTA are not compatible with the property of detailed balance.

PHAISTOS also has some overlap with the PROFASI simulation package,[45] in the sense that both implement the BGS move[9] and the PROFASI energy function.[17] The PROFASI simulation program was designed as a tool for studying protein aggregation and is thus highly optimized for many-chain simulation using their lightweight force-field and under the assumption of fixed bond angles. PHAISTOS aims to provide a greater flexibility in the choice of energies and a wider selection of moves and is not limited to a fixed bond-angle representation.

The closest alternatives to PHAISTOS are perhaps the CAMPARI software package[1] and the Monte Carlo package in CHARMM,[2] which both provide functionality for conducting MCMC simulations using various force-fields and moves. Compared with PHAISTOS, the selection of force-fields and moves differ, and the focus is different. For instance, PHAISTOS has a strong focus on sampling using probabilistic models of local structure, which is not supported by either of the two alternatives.

The current version of the PHAISTOS framework has several limitations. To a user familiar with MD software, the primary limitation will presumably be the lack of explicit solvent models in the framework. The large conformational moves that provide the sampling advantage of Monte Carlo simulations are difficult to combine with an explicit solvent representation. In line with other Monte Carlo simulation packages, PHAISTOS is, therefore, currently limited to implicit solvent simulations. Another limitation is that PHAISTOS can currently only simulate a single polypeptide at a time. This restriction will be removed in the next release of the software, which will also include implementations of several new force-fields.

As the list of applications demonstrates, even in its current form, the framework provides the necessary tools for conducting relevant MCMC simulations of protein systems. The framework incorporates generalized ensembles and novel Monte Carlo moves, including moves that incorporate structural priors as proposal distributions. These features are unique to this framework and have been shown to increase sampling efficiency considerably.

The software is freely available under the GNU General Public License v3.0. All source code is fully documented using the Doxygen system (http://www.doxygen.org), and a user manual is available for detailed descriptions on how to set up simulations. Both sources of information are accessible via the PHAISTOS web site, http://phaistos.sourceforge.net.

Keywords: Markov chain Monte Carlo simulation · protein structure · probabilistic models · local moves · conformational sampling

How to cite this article: W. Boomsma, J. Frellsen, T. Harder, S. Bottaro, K. E. Johansson, P. Tian, K. Stovgaard, C. Andreetta, S. Olsson, J. B. Valentin, L. D. Antonov, A. S. Christensen, M. Borg, J. H. Jensen, K. Lindorff-Larsen, J. Ferkinghoff-Borg, T. Hamelryck, *J. Comput. Chem.* **2013**, *34*, 1697–1705. DOI: 10.1002/jcc.23292

[1] A. Vitalis, R. V. Pappu, *Annu. Rep. Comput. Chem.* **2009**, *5*, 49.
[2] J. Hu, A. Ma, A. R. Dinner, *J. Comput. Chem.* **2006**, *27*, 203.
[3] J. P. Ulmschneider, M. B. Ulmschneider, A. Di Nola, *J. Phys. Chem. B* **2006**, *110*, 16733.
[4] S. Bottaro, W. Boomsma, K. E. Johansson, C. Andreetta, T. Hamelryck, J. Ferkinghoff-Borg, *J. Chem. Theory Comput.* **2012**, *8*, 695.
[5] M. Habeck, M. Nilges, W. Rieping, *Phys. Rev. Lett.* **2005**, *94*, 18105.
[6] M. R. Betancourt, *J. Chem. Phys.* **2005**, *123*, 174905.
[7] C. Smith, T. Kortemme, *J. Mol. Biol.* **2008**, *380*, 742.
[8] J. P. Ulmschneider, W. L. Jorgensen, *J. Chem. Phys.* **2003**, *118*, 4261.
[9] G. Favrin, A. Irbäck, F. Sjunnesson, *J. Chem. Phys.* **2001**, *114*, 8154.
[10] R. L. Dunbrack, F. E. Cohen, *Protein Sci.* **1997**, *6*, 1661.
[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. Popovíc, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker and P. Bradley, *Methods Enzymol.* **2011**, *487*, 545.
[12] T. Przytycka, *Proteins* **2004**, *57*, 338.
[13] T. Hamelryck, J. T. Kent, A. Krogh, *Plos Comput. Biol.* **2006**, *2*, e131.
[14] W. Boomsma, K. V. Mardia, C. C. Taylor, J. Ferkinghoff-Borg, A. Krogh, T. Hamelryck, *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105*, 8932.
[15] T. Harder, W. Boomsma, M. Paluszewski, J. Frellsen, K. E. Johansson, T. Hamelryck, *BMC Bioinform.* **2010**, *11*, 306.
[16] N. Gō, H. A. Scheraga, *Macromolecules* **1970**, *3*, 178.
[17] A. Irbäck, S. Mitternacht, S. Mohanty, *PMC Biophys.* **2009**, *2*, 2.
[18] G. A. Kaminski, R. A. Friesner, J. Tirado-Rives, W. L. Jorgensen, *J. Phys. Chem. B* **2001**, *105*, 6474.
[19] D. Qiu, P. S. Shenkin, F. P. Hollinger, W. C. Still, *J. Phys. Chem. A* **1997**, *101*, 3005.
[20] I. Lotan, F. Schwarzer, D. Halperin, J.C. Latombe, *J. Comput. Biol.* **2004**, *11*, 902.
[21] A. Irbäck, S. Mitternacht, *Proteins* **2007**, *71*, 207.
[22] D. W. Li, S. Mohanty, A. Irbäck, S. Huo, *PLoS Comput. Biol.* **2008**, *4*, e1000238.
[23] J. W. Ponder, F. M. Richards, *J. Am. Chem. Soc.* **1987**, *8*, 1016.
[24] B. Roux, T. Simonson, *Biophys. Chem.* **1999**, *78*, 1.
[25] W. C. Still, A. Tempczyk, R. C. Hawley, T. Hendrickson, *J. Am. Chem. Soc.* **1990**, *112*, 6127.
[26] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435.
[27] K. E. Johansson, T. Hamelryck, *Proteins*, doi: 10.1002/prot.24277.
[28] U. H. E. Hansmann, Y. Okamoto, *J. Comput. Chem.* **1997**, *18*, 920.
[29] B. A. Berg, T. Neuhaus, *Phys. Rev. Lett.* **1992**, *68*, 9.
[30] B. Hesselbo, R. B. Stinchcombe, *Phys. Rev. Lett.* **1995**, *74*, 2151.
[31] J. Ferkinghoff-Borg, *J. Eur. Phys. J. B* **2002**, *29*, 481.
[32] E. De Alba, J. Santoro, M. Rico, M. Jimenez, *Protein Sci.* **1999**, *8*, 854.
[33] Schrödinger, LLC., The PyMOL Molecular Graphics System, Version 0.99rc6.
[34] T. Harder, M. Borg, W. Boomsma, P. Røgen, T. Hamelryck, *Bioinformatics* **2012**, *28*, 510.
[35] T. Hamelryck, M. Borg, M. Paluszewski, J. Paulsen, J. Frellsen, C. Andreetta, W. Boomsma, S. Bottaro, J. Ferkinghoff-Borg, *PLoS ONE* **2010**, *5*, e13714.
[36] K. Stovgaard, C. Andreetta, J. Ferkinghoff-Borg, T. Hamelryck, *BMC Bioinform.* **2010**, *11*, 429.
[37] N. G. Sgourakis, O. F. Lange, F. DiMaio, I. André, N. C. Fitzkee, P. Rossi, G. T. Montelione, A. Bax, D. Baker, *J. Am. Chem. Soc.* **2011**, *133*, 6288.
[38] A. Cavalli, X. Salvatella, C. M. Dobson, M. Vendruscolo, *Proc. Natl. Acad. Sci. U.S.A.* **2007**, *104*, 9615.
[39] Y. Shen, O. Lange, F. Delaglio, P. Rossi, J. M. Aramini, G. Liu, A. Eletsky, Y. Wu, K. K. Singarapu, A. Lemak, A. Ignatchenko, C. H. Arrowsmith, T. Szyperski, G. T. Montelione, D. Baker, A. Bax, *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 4685.
[40] S. Olsson, W. Boomsma, J. Frellsen, S. Bottaro, T. Harder, J. Ferkinghoff-Borg, T. Hamelryck, *J. Magn. Reson.* **2011**, *213*, 182.
[41] S. Lloyd, *IEEE Trans. Inf. Theory* **1982**, *28*, 129.
[42] P. Røgen, B. Fain, *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 119.
[43] T. Harder, M. Borg, S. Bottaro, W. Boomsma, S. Olsson, J. Ferkinghoff-Borg, T. Hamelryck, *Structure* **2012**, *20*, 1028.
[44] R. Das, D. Baker, *Annu. Rev. Biochem.* **2008**, *77*, 363.
[45] A. Irbäck, S. Mohanty, *J. Comput. Chem.* **2006**, *27*, 1548.

Received: 6 December 2012
Revised: 14 March 2013
Accepted: 20 March 2013
Published online on 26 April 2013