A Rapid Solvent Accessible Surface Area Estimator for Coarse Grained Molecular Simulations

Shuai Wei,[a] Charles L. Brooks, III,*[b] and Aaron T. Frank*[b]

The rapid and accurate calculation of solvent accessible surface area (SASA) is extremely useful in the energetic analysis of biomolecules. For example, SASA models can be used to estimate the transfer free energy associated with biophysical processes, and when combined with coarse-grained simulations, can be particularly useful for accounting for solvation effects within the framework of implicit solvent models. In such cases, a fast and accurate, residue-wise SASA predictor is highly desirable. Here, we develop a predictive model that estimates SASAs based on C_α-only protein structures. Through an extensive comparison between this method and a comparable method, POPS-R, we demonstrate that our new method, Protein-C_α Solvent Accessibilities or PCASA, shows better performance, especially for unfolded conformations of proteins. We anticipate that this model will be quite useful in the efficient inclusion of SASA-based solvent free energy estimations in coarse-grained protein folding simulations. PCASA is made freely available to the academic community at https://github.com/atfrank/PCASA. © 2017 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.24709

Introduction

Compared to simulations in which solvent molecules are explicitly represented, simulations that use implicit solvent models are significantly more efficient. Typically, this improved efficiency is exploited to either carry out longer simulations or to simulate larger and more complex molecular systems. One group of implicit solvent models use solvent accessible surface area (SASA) as an indicator of solute-solvent contacts and the corresponding solvation forces and energies, for a given atom in a molecular system, can be estimated as a function of its corresponding atom type and its SASA.[1–3] These methods are usually estimate the solvent transfer free energy (TFE) using the famous model of Tanford.[2,4,5] SASA-based implicit solvent models have been used to study changes in biomolecular stability in different solvent environments by calculating the TFEs between different solvent conditions.[6–8] They have also been used in simulations in which the biomolecule is coarse-grained (i.e., modeled using a reduced representation). For instance, O'Brien et al. developed the molecular transfer model to study protein folding in different osmolyte solvents by incorporating the SASA-based TFE estimation with their two-bead per residue Gō model.[9,10]

SASA can be rigorously calculated by rolling a spherical probe along the surface of a biomolecule.[7,11] Analytical formula for SASA calculation have been derived by Connolly,[12] Richmond,[13] followed by Buša et al.,[14] and Klenin et al.[15] These calculations are computationally expensive and so it is not efficient to directly implement SASA-based TFE calculations on-the-fly during simulations of biomolecules. Therefore, it is desirable to have a fast and accurate numerical estimator for SASA; several approaches that estimate SASAs from all-atom models have been developed.[16–18] Most of these methods, such as the method implemented by Hasel et al.,[18] LCPO,[19] and POPS,[20,21] are constructed in the spirit of the fundamental framework by Wodak and Janin,[22] which accurately estimates SASAs using a probability estimation as a function of relative distances of each atom with its neighbors. The most recent and popular variant of this approach, POPS,[20,21] reported very accurate estimation of SASA values for all-atom structures and the method was further extended for C_α based coarse-grained protein structures (POPS-R).[21] Unfortunately, because POPS-R was parameterized using only native protein structures, it exhibits reduced performance when applied to unfolded conformations and is thus of diminished utility in protein folding studies.

In this work, we develop a fast and accurate predictive model that estimates SASA from C_α coordinates only, and importantly, one that exhibits reduced bias for folded structures. More specifically, we implemented a simple linear model that, for a given residue, depends only on the geometric distance between the corresponding C_α of that residue and C_α of the residues within some cutoff distance. Unlike other methods, the training set used for parameterizing our method contained both folded and unfolded protein conformations. Compared to the current-state-of-the-art, POPS-R, our method, **P**rotein-**C**$_\alpha$ **S**olvent **A**ccessibilities (which will be referred to as PCASA), exhibited better overall accuracy over an independent testing set that contained both folded and unfolded protein conformations. The consistency of the performance for both folded and unfolded conformations indicates an unbiased

[a] S. Wei
Department of Chemistry, University of Michigan, Ann Arbor, MI 48109
[b] C. L. Brooks, III, A. T. Frank
Departments of Biophysics and Chemistry, University of Michigan, Ann Arbor, MI 48109
E-mail: brookscl@umich.edu, E-mail: afrankz@umich.edu
© 2017 Wiley Periodicals, Inc.

Table 1. Bayesian-derived estimates of the expected errors (σ) of SASA predictors parameterized using different combinations of r_{cut} and γ.

	$\gamma=-5$	$\gamma=-3$	$\gamma=-1$
r_{cut} = 5.0 Å	47.5 Å^2	47.5 Å^2	45.6 Å^2
r_{cut} = 10.0 Å	47.5 Å^2	47.3 Å^2	28.0 Å^2
r_{cut} = 15.0 Å	47.5 Å^2	47.1 Å^2	32.1 Å^2
r_{cut} = 20.0 Å	47.5 Å^2	47.1 Å^2	34.9 Å^2

Table 2. Accuracy statistics for total SASAs estimated using PCASA and POPS-R.

	R	τ	ρ	RMSE (Å^2)	MAE (Å^2)
POPS-R	0.998	0.962	0.998	2332.1	1994.3
PCASA	0.999	0.968	0.998	627.5	448.3

parameterization, which is a merit of its implementation, and makes it particularly well-suited for SASA-based TFE calculations during coarse-grained protein folding simulations.

Methodology

Model formula and parameterization

The SASA of a given residue, i, is calculated as

$$\mathrm{SASA}_i = \alpha_i - \sum_{j\in(r_{ij}<r_{cut})} \beta_{ij} r_{ij}^{\gamma}, \quad (1)$$

where the index j runs over all residues within r_{cut} of residue i, α_i is the reference SASA corresponding to a fully exposed residue of type i, β_{ij} is a scaling parameter that depends on both i and j, and γ is a common parameter. Separate predictive models were parameterized with r_{cut} and γ set to either 5.0, 10.0, 15.0, and 20.0 Å and -5, -3, and -1, respectively. For each combination of r_{cut} and γ, a predictive model was parameterized using a linear Bayesian regression model via the MCMCregress function implemented in the R package, MCMCpack.[23] In addition to providing parameters for the models (α_is and β_{ij}s), MCMCregress also returns an estimate of the accuracy of the predictive models (σ). The σ values determined for each model were used to determine the best combination of r_{cut} and γ, and thus final predictive model implemented in PCASA.

The simplicity of the model we use to predict SASA results in a more straightforward expression of SASA of some residue i with respect to r_{ij} than current physics-based models. Specifically, the derivative of SASA with respect r_{ij} as required for calculation of SASA-based solvation forces, is simply expressed as

$$\frac{d}{dr_{ij}}\mathrm{SASA}_i = -\gamma \sum_{j\in(r_{ij}<r_{cut})} \beta_{ij} r_{ij}^{\gamma-1}. \quad (2)$$

Protein structure dataset for model parameterization

To parameterize PCASA, we used a large protein structure dataset containing 527 proteins that were randomly chosen from the protein data bank (PDB). To reduce performance bias toward folded, native-like protein conformations, for each protein in the dataset, 10 folded and 10 unfolded conformations were generated by performing MD simulations. Simulations were performed with the KB Gō-like model[24] at 600K with the step size of 22 fs. Structures were saved every 200 steps. For the entire dataset, a total of 10,520 structures were therefore obtained.

To calculate the reference SASAs (the target used to train PCASA), all-atom models were first reconstructed from the C_α KB Gō-like models using MMTSB.[25] For each all-atom model, the reference SASA for each residue was calculated using CHARMM.[26] To parameterize PCASA, the 527 proteins was first divided into a training set (80%; Supporting Information Table S1) and an independent testing set (20%; Supporting Information Table S2). PCASA was then trained using the training set and the performance evaluated using the testing set.

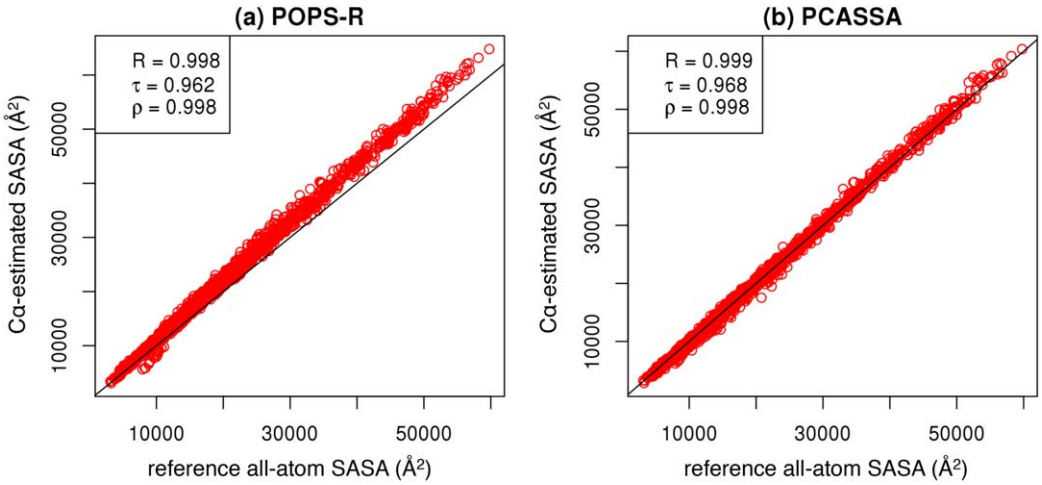

Figure 1. Total SASAs for proteins in the testing set. Shown are correlations plots comparing the total reference SASAs and a) POPS-R- and b) PCASA-predicted total SASAs for each protein in the testing set. [Color figure can be viewed at wileyonlinelibrary.com]

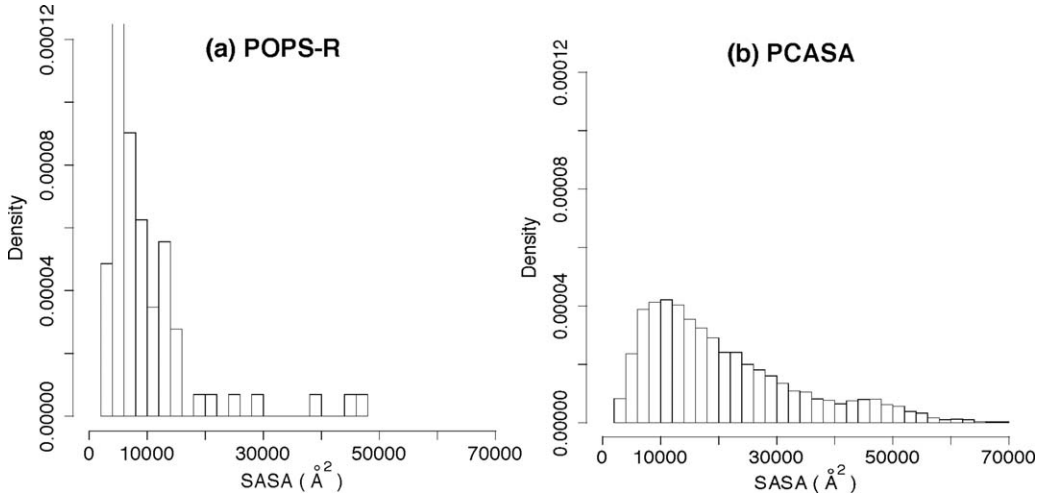

Figure 2. Training set distributions. Shown are the histograms of the total SASAs in a) the POPS-R and b) the PCASA training sets.

Assessing the performance of PCASA

To assess the performance of PCASA, we compared the PCASA-calculated SASAs to the reference SASAs by computing the Pearson correlation coefficient (R); the Spearman rank correlation coefficient (ρ); the Kendall correlation coefficient (τ); the mean-absolute-error (MAE); and the root-mean-squared-error (RMSE). Using these "metrics," PCASA was evaluated on an independent testing set that included both folded and unfolded conformations, and then separately on subsets that included only folded (native-like) conformations and only unfolded (denatured) conformations. For comparison, the same analyses were also performed when predicting SASAs using POPS-R.

Results and Discussion

The predictive model with r_{cut} and γ set to 10 Å and −1, respectively, exhibited the lowest estimated errors

In this work, we parameterized simple predictive models that enabled residue-wise SASAs to be estimated based solely on C_α coordinates. As is evident from eq. (1), in addition to the model parameters α and β, the model also depends on the choice of r_{cut} and γ. In this work, we explored predictive models in which r_{cut} and γ were set to 5.0, 10.0, 15.0, and 20.0 Å and −5, −3, and −1, respectively. Shown in Table 1 are the expected errors of the various models explored in this work. These estimates were obtained as output of the linear Bayesian regression that we carried out using MCMCregress function in MCMCpack R package.[23] As can be observed in Table 1, the model with r_{cut} and γ set to 10 Å and −1, respectively, exhibited the lowest expected error (28.0 Å). As such, we chose this model to be the model implemented in PCASA (Supporting Information Table S3-S5).

Table 3. Accuracy statistics for residue-wise SASAs estimated using PCASA and POPS-R.

	R	τ	ρ	RMSE (Å^2)	MAE (Å^2)
POPS-R	0.799	0.612	0.807	31.5	24.4
PCASA	0.843	0.654	0.843	26.9	20.8

PCASA predicts total SASA and residue-wise SASA with high accuracy

We first assessed the performance of PCASA by computing the total SASAs (in Å^2) for all the proteins in the testing set (Supporting Information Table S2). As can be seen in Figure 1, PCASA-predicted SASAs agree well with the reference SASAs, as evidenced by the high correlation between the two. For example, the Pearson (R), Kendall (τ), and Spearman (ρ) correlations were 0.999, 0.968, and 0.998, respectively (Table 2). By comparison, the R, τ, and ρ values for POPS-R were 0.998, 0.962, and 0.998, respectively (Table 2). Together, these results indicate that both PCASA and POPS-R were able to predict SASAs with good accuracy. For this testing set, however, PCASA was able to achieve a higher degree of accuracy than POPS-R. For example, the MAE and the RMSE between PCASA-predicted and reference SASA were 448.3 and 627.5 Å^2, compared with 1994.3 and 2332.1 Å^2, respectively, for POPS-R. It is interesting to note that this discrepancy appears to be more pronounced for proteins with larger total SASAs (Fig. 1).

To test whether the difference in performance of PCASA and POPS-R for proteins with larger total SASAs may be due to differences in the statistically "coverage" in the database used to train PCASA and POPS-R, respectively, we compared the distribution of total SASAs in both training sets. We found that the distribution of total SASAs in the PCASA training set was more continuous than POPS-R, and spanned a larger range, (2329 to 69,850 Å^2), compared to (2009, 46,789 Å^2) for POPS-R; generally, there were more examples of proteins with larger total SASAs in the PCASA training set than POPS-R. These differences in the training set distributions provide the most likely explanation as to why PCASA exhibits greater accuracy for proteins with larger to total SASAs (Fig. 2).

To further assess the performance limits of PCASA, we repeated the above analysis, but instead of comparing the total SASA, we compared the residue-wise SASA predictions.

Table 4. Accuracy statistics for residue-wise SASAs estimated for folded and unfolded structures using PCASA and POPS-R.

	Folded					Unfolded				
	R	τ	ρ	RMSE (Å^2)	MAE (Å^2)	R	τ	ρ	RMSE (Å^2)	MAE (Å^2)
POPS-R	0.808	0.615	0.811	30.0	23.4	0.751	0.559	0.755	33.1	25.7
PCASA	0.838	0.647	0.838	26.3	20.3	0.819	0.621	0.814	27.2	21.1

Mirroring the results above, the R, τ, and ρ values for PCASA were 0.843, 0.654, and 0.843, respectively (Table 3), compared with 0.799, 0.612, and 0.807, respectively, for POPS-R (Table 3). Likewise, the MAE and RMSE between PCASA-predicted and reference SASA were 26.9 and 20.8 Å^2, compared with 31.5 and 24.4 Å^2, respectively, for POPS-R, confirming that not only was PCASA better able to recapitulate the total reference SASAs for proteins, it was also able to better recapitulate the residue-wise reference SASAs.

PCASA predicts SASAs for folded and unfolded conformations with similar accuracy

As mentioned above, we suspected that the performance of POPS-R for the unfolded structures would not be as good as for folded structures due to their bias for the native structures in the training set used to parameterize the model. To test this point, we performed statistical analysis by dividing our dataset into two groups (folded and unfolded) based on the protein radius of gyration (R_g). As we recorded structures from simulations approximately with equal sampling of folded and unfolded states, half of the structures with low R_g for each protein are grouped as the folded structures and the rest as the unfolded structure. In Table 4, we present the statistics for PCASA and POPS-R SASA estimation for folded and unfolded conformations.

As assumed, the POPS-R performance is the much better for the folded protein structures with a much higher R value of 0.808 than for the unfolded structures with an R value of 0.751. While the model developed in this work shows consistent performance for both folded and unfolded structures with very similar R values of 0.838 and 0.819. The same picture emerges from the comparison of the MAE and RMSE.

To examine this further, six proteins were randomly picked from our testing set and representative folded, molten globule-like, and unfolded structures were selected for each. For each protein and each representative conformation, the total PCASA- and POPS-R-predicted SASAs were compared directly to their corresponding reference values. Consistently, for the folded, molten globule-like, and unfolded conformations, PCASA predictions mirrored closely the reference values. In contrast, POPS-R exhibited increasing discrepancies going from folded to unfolded conformations (Fig. 3). The difference

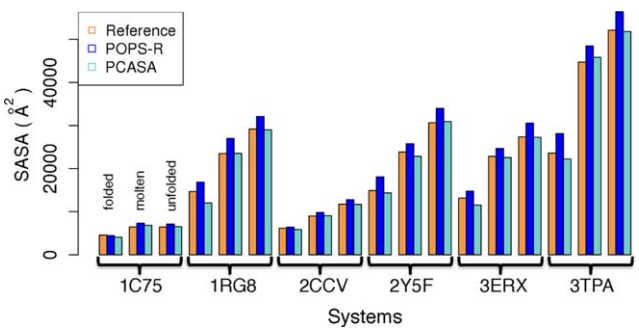

Figure 3. Comparing total SASAs for folded, molten globule-like, and unfolded conformations. The bar plot shows the comparison between the reference SASA (orange), POPS-R- (blue), and PCASA-predicted (cyan) SASAs for six randomly chosen proteins in the training set. [Color figure can be viewed at wileyonlinelibrary.com]

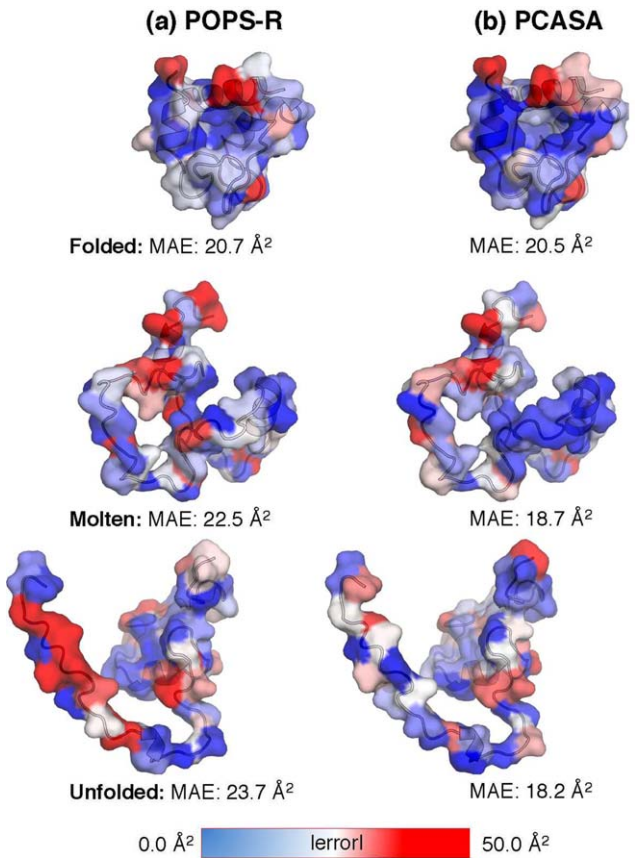

Figure 4. Reside-wise difference between POPS-R- and PCASA-predicted SASA and the reference SASA. In this figure, the absolute differences between reference, a) POPS-R-predicted, and b) PCASA-predicted SASA values are projected on the surface of representative folded (top), molten globule-like, (middle) and unfolded (bottom) conformations of the protein corresponding to PDBID: 1C75. Also indicated for each is the residue-wise MAE. [Color figure can be viewed at wileyonlinelibrary.com]

is especially apparent when the difference between the reference and predicted SASA values are projected onto the individual structures of the proteins. As exemplified for the protein shown in Figure 4, the residue-wise discrepancy between predicted and reference SASA is more pronounced for POPS-R than PCASA, and in the case of POPS-R, the discrepancy increases as one goes from the folded to unfolded conformations.

To summarize the results presented above, our findings highlight the ability of PCASA to accurately recapitulate the reference SASAs, regardless of whether the conformations are native-like (folded) or non-native-like (unfolded). This consistency is highly desired as an important potential application of PCASA is to study protein folding and unfolding via coarse grained simulations coupled with a SASA-based implicit solvent model. In such simulations, folded and unfolded conformations may dynamically interconvert, and as such, accurately calculating the solvation energy differences, and in turn, the SASAs, will be crucial.

Conclusions

In this work, we developed a fast and straightforward method that predicts residue-wise SASAs from C_α coordinates of protein structures. This was accomplished using a large dataset of folded and unfolded structures and an equation that has a simple dependence on distances between C_α atoms. Our method, PCASA, was parameterized using a Bayesian linear regression model. We demonstrated PCASA is able to accurately predict residue-wise, as well, total SASAs for conformations in an independent testing set. Importantly, PCASA was also shown not to be biased toward either folded or unfolded conformations. Among other applications, we envision that PCASA should find utility in accounting for solvation effects in coarse-grained protein folding simulations via SASA-based estimation of the transfer free energies. Currently, however, PCASA can only estimate SASA for protein only systems. In future work, we will develop methods that enable PCASA to estimate protein SASAs for proteins in the presence of other molecules, like, nucleic acids, lipids, and small molecules. To facilitate its widespread use, PCASA is made freely available to the academic community at https://github.com/atfrank/PCASA.

Acknowledgments

The authors would like to acknowledge the valuable scientific contribution of Sean M. Law, who wrote the C++ library used to implement PCASA.

Keywords: residue-wise SASA · go-model · coarse-grained protein structure · Bayesian linear regression

How to cite this article: S. Wei, C. L., Brooks III, A. T., Frank. *J. Comput. Chem.* **2017**, *38*, 1270–1274. DOI: 10.1002/jcc.24709

Additional Supporting Information may be found in the online version of this article.

[1] M. Feig, C. L. Brooks, III, *Curr. Opin. Struct. Biol.* **2004**, *14*, 217.
[2] G. Ziv, G. Haran, *J. Am. Chem. Soc.* **2009**, *131*, 2942.
[3] A. Momen-Roknabadi, M. Sadeghi, H. Pezeshk, S. A. Marashi, *BMC Bioinformatics* **2008**, *9*, 1.
[4] C. Tanford, *Adv. Protein Chem.* **1968**, *23*, 121.
[5] C. Tanford, *Adv. Protein Chem.* **1970**, *24*, 1.
[6] S. L. Lin, R. Nussinov, D. Fischer, H. J. Wolfson, *Proteins* **1994**, *18*, 94.
[7] A. Shrake, J. Rupley, *J. Mol. Biol.* **1973**, *79*, 351.
[8] F. M. Richards, *Annu. Rev. Biophys. Bioeng.* **1977**, *6*, 151.
[9] E. P. O'Brien, B. R. Brooks, D. Thirumalai, *Biochemistry* **2009**, *48*, 3743.
[10] E. P. O'Brien, G. Ziv, G. Haran, B. R. Brooks, D. Thirumalai, *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 13403.
[11] B. Lee, F. M. Richards, *J. Mol. Biol.* **1971**, *55*, 379.
[12] M. L. Connolly, *J. Appl. Crystallogr.* **1983**, *16*, 548.
[13] T. J. Richmond, *J. Mol. Biol.* **1984**, *178*, 63.
[14] J. Buša, J. Džurina, E. Hayryan, S. Hayryan, C. K. Hu, J. Plávka, I. Pokorný, J. Skřivánek, M. C. Wu, *Comput. Phys. Commun.* **2005**, *165*, 59.
[15] K. V. Klenin, F. Tristram, T. Strunk, W. Wenzel, *J. Comput. Chem.* **2011**, *32*, 2647.
[16] M. Petitjean, *J. Comput. Chem.* **1994**, *15*, 507.
[17] M. L. Connolly, *J. Am. Chem. Soc.* **1985**, *107*, 1118.
[18] W. Hasel, T. F. Hendrickson, W. C. Still, *Tetrahedron Comput. Methodol.* **1988**, *1*, 103.
[19] J. Weiser, P. S. Shenkin, W. C. Still, *J. Comput. Chem.* **1999**, *20*, 217.
[20] F. Fraternali, L. Cavallo, *Nucleic Acids Res.* **2002**, *30*, 2950.
[21] L. Cavallo, J. Kleinjung, F. Fraternali, *Nucleic Acids Res.* **2003**, *31*, 3364.
[22] S. J. Wodak, J. Janin, *Proc. Natl. Acad. Sci. USA* **1980**, *77*, 1736.
[23] A. D. Martin, K. M. Quinn, J. H. Park, et al. *J. Stat. Softw.* **2011**, *42*, 1.
[24] J. Karanicolas, C. L. Brooks, III, *J. Mol. Biol.* **2003**, *334*, 309.
[25] M. Feig, J. Karanicolas, C. L. Brooks, III, *J. Mol. Graph. Model.* **2004**, *22*, 377.
[26] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus, *J. Comput. Chem.* **1983**, *4*, 187.

Received: 10 August 2016
Revised: 8 November 2016
Accepted: 17 November 2016
Published online on 16 April 2017