

SIMONA 1.0: An Efficient and Versatile Framework for Stochastic Simulations of Molecular and Nanoscale Systems

T. Strunk,^{*,[a]} M. Wolf,^{*,[a]} M. Brieg,^[b] K. Klenin,^[b] A. Biewer,^[a] F. Tristram,^[a] M. Ernst,^[a]
P. J. Kleine,^[a] N. Heilmann,^[a] I. Kondov,^[b] and W. Wenzel^{†[a]}

Molecular simulation methods have increasingly contributed to our understanding of molecular and nanoscale systems. However, the family of Monte Carlo techniques has taken a backseat to molecular dynamics based methods, which is also reflected in the number of available simulation packages. Here, we report the development of a generic, versatile simulation package for stochastic simulations and demonstrate its application to protein conformational change, protein-protein association, small-molecule protein docking, and simulation of the growth of nanoscale clusters of organic molecules. Simulation of molecular and nanoscale systems

(SIMONA) is easy to use for standard simulations via a graphical user interface and highly parallel both via MPI and the use of graphical processors. It is also extendable to many additional simulations types. Being freely available to academic users, we hope it will enable a large community of researchers in the life- and materials-sciences to use and extend SIMONA in the future. SIMONA is available for download under <http://int.kit.edu/nanosim/simona>.

Introduction

Driven by ever-increasing computational resources molecular simulation methods have become increasingly important to understand (bio)molecular function in the last decades.^[1–4] Although experimental techniques have been developed that probe smaller and smaller scales,^[5,6] our ability to model the behavior of molecules, clusters, and their aggregates has steadily increased. During these developments, the family of Monte Carlo methods,^[7] which was very popular to elucidate the thermodynamic properties in condensed matter physics,^[8] has increasingly taken the backseat in the number of applications compared to the family of molecular dynamics methods. The latter, for which a number of well-supported simulation packages are available, now permits simulation of moderate size systems for hundreds of microseconds (using specialized hardware)^[11] or even very large systems^[9] for shorter time scales. Molecular dynamics simulations, which solve Newton's equation of motion on a finely discretized time scale, allow us to follow the time evolution of the system in a virtual experiment. Monte Carlo methods, in comparison, generate a thermodynamic ensemble of states and thus offer complementary information, but few generically applicable program packages are available today.

The great advantage of the molecular dynamics approach, to observe physical processes and directly extract kinetic information, is limited by the fact that the individual time step of the method has hovered around the femtosecond range, seriously hampering the observation of long timescale processes and the generation of sufficient statistics. This has led to the development of techniques, such as the replica exchange methods^[10,11] (called parallel tempering^[12–15] in the context of Monte Carlo), in which a continuous trajectory is broken into replica simulations at dif-

ferent temperatures, which enhances the sampling of the conformational space, while obviously losing kinetic information.

Many simulation and modeling programs of nanoscale systems have been developed and perfected in the recent years.^[16–21] For molecular dynamics of protein systems GRO-MACS,^[16] CHARMM,^[20] and AMBER,^[17] to name just a few, provide many different forcefields including the popular Amber99SB and optimized potentials for liquid simulations (OPLS) implementations.^[22–26] However, there are many processes where thermodynamic information is sufficient to describe or even guide experiment. Examples of such processes are, for biomolecular systems, the relative stability of proteins, investigations of protein-protein, and protein-ligand interactions. In the material sciences, aggregation, nanoparticle growth, and the properties of thin films can often be described on the basis of thermodynamic observations.

In contrast to molecular dynamics methods, there are a few off-the-shelf simulation packages for Monte Carlo simulations.

[a] T. Strunk, M. Wolf, A. Biewer, F. Tristram, M. Ernst, P. J. Kleine, N. Heilmann, W. Wenzel Institute of Nanotechnology, Karlsruhe Institute of Technology, PO Box 3640, 76021 Karlsruhe, Germany
E mail: wolfgang.wenzel@kit.edu

[b] M. Brieg, K. Klenin, I. Kondov
Steinbuch Center for Computing, Karlsruhe Institute of Technology, PO Box 3640, 76021 Karlsruhe, Germany

*These authors contributed equally to this work.

Contract/grant sponsor: FP7 MINOTOR; contract/grant number: GA 228424

Contract/grant sponsor: FP7 MMM@HPC; contract/grant number: GA 261594

Contract/grant sponsor: Baden Württemberg Stiftung (HPC 5)

Contract/grant sponsor: Carl Zeiss Stiftung (T.S.)

Contract/grant sponsor: Landesgraduiertenförderung Baden Württemberg (M.B.)

Simulations of protein systems were for example carried out using Rosetta,^[27] Profasi,^[21] simple molecular mechanics for proteins (SMMP),^[28] and POEM.^[29] Although some molecular dynamics software packages can perform Monte Carlo sampling, such implementations are typically not numerically optimal as they cannot exploit specific features of Monte Carlo simulations (such as modifications of only a small part of the system at a time). There are a number of specialized Monte Carlo simulation packages for protein simulations and other applications. Rosetta^[27] ranks well in the biannual critical assessment of techniques for protein structure prediction (CASP) competition using a custom fragment assembly and Monte Carlo refinement protocol,^[30–32] whereas Profasi was able to fold several proteins from a coiled or extended conformation.^[21] Customizing these simulations for other material classes like crystals is often tedious due to the high degree of specificity toward simulations of proteins. Packages like OPTIM^[33] were developed to optimize clusters of atoms and/or molecules and adapted to biomolecules to carry out discrete path sampling or to calculate pathways and rates in energy landscapes of biomolecules^[33–35] using the PATHSAMPLE driver of OPTIM.

In this article, we describe SIMONA (Simulation of MOlecular and NANoscale systems), a generic monte-carlo (MC)-based simulation package, which enables rapid prototyping of forcefields and simulation methods for efficient Monte Carlo simulations. The design goals of SIMONA were to provide the community with an efficient and adaptable tool for complex molecular simulations at three levels: at the “application” level, a graphical user interface eases the setup, execution, and monitoring of simulations for standard simulations. At the “new system” level, a python script-based preprocessor simplifies parameterization of novel components, for example, small molecules, polymers, or metal-organic frameworks and implementation of new forcefield parameterizations. At the “new forcefield level,” modifications of the object-oriented C++ backend simplify rapid prototyping of novel forcefield terms and simulation protocols, for example, evolutionary algorithms. Many forcefield components are already implemented for graphical processors in the OpenCL standard.

After describing the implementation, we demonstrate the application of SIMONA for three interesting application fields: we modeled cluster formation for materials relevant for organic light emitting diodes (OLED),^[36–39] simulate protein folding, protein–protein interactions, and conduct protein–ligand docking studies. However, these applications merely serve as examples: SIMONA will be freely available (in source code) to academic users and we hope that the work reported here serves as a useful toolbox for the molecular simulation community to perform efficient Monte Carlo-based simulations for many other systems and to implement, in a modern and efficient platform, many of the existing protocols^[40–42] that are already available but are difficult to use for general problems at present. SIMONA is available for download under <http://int.kit.edu/nanosim/simona>.

Methods

The applicability of molecular simulation methods requires both an efficient implementation and an adaptable user interface. SIMONA targets problems concerning the sampling of

thermodynamic ensembles of molecules that interact by various classical interactions. To implement this package, we have thus followed the path taken by various MD packages to generate a “topology file” by a preprocessor that specifies the system and interactions and a computational engine that reads this information, performs the simulation, and returns a trajectory along with other output. The simulation engine can be made most efficient if it abstracts from the details of the specific system and represents the simulation as a movement of point-like objects (usually atoms) in space according to rules that vary from application to application. This approach also paves the way for coarse-graining simulations. The computational engine, written in C++, can be adapted to different computational platforms.

To make this simulation package applicable to a wide range of problems, one must provide a means to encode, via a preprocessor, the input data (typically information regarding the coordinates, chemical properties, interactions, and simulation method) into the standardized format readable by the simulation (XML in SIMONA). This preprocessing step typically requires very little computational effort in comparison with the subsequent simulation, yet varies significantly from one application to the next. Using a hierarchical scheme, such as XML, for this purpose, encapsulates information for specific classes of the program and makes the input files more transparent and adaptable.

Although standard forcefields exist for most biomolecules,^[43] encoding polymers or small molecules, transition metals or coarse-grained objects require quite different input specifications. For this reason, we have implemented the pre- and post processor in Python, a script-based language that can be adapted on-the-fly to many changing requirements. The basic preprocessing modules thus provide (a) Python classes that read certain widely used input formats, (b) a hierarchy of Python classes that correspond to the XML elements recognized by the computational engine, and (c) the “glue” that transforms the data generated by the input into a tree of classes corresponding to the XML output. The subsequent execution of the algorithm by the computational engine is then automatic; the user thus interfaces with the program only via the Python interface for pre- and post processing.

The generation of the input XML can be controlled by the user at various levels, in the following listed in the order of the complexity of the required changes of the program. (1) For standardized applications, the user can set up the simulation with the graphical interface and there are several tutorials for selected applications. (2) Nonstandard interactions or parameter values can be defined by changing assignments in the preprocessors as discussed below. (3) Complex algorithms, which are key to many specialized MC methods, can be encoded using a XML-based “programming language,” and (4) class derivation on the python and C++ side will permit the expert user to implement novel methods, while inheriting all existing features. Documentation and tutorials are provided on the download webpage.

System definition

The encoding of the simulated system (i.e., biomolecules, polymers, and nanoparticles) is split into four distinct sections by the preprocessor: Configuration, Moves, Forcefield, and Algorithm as illustrated in Figure 1. When reading an input file, the

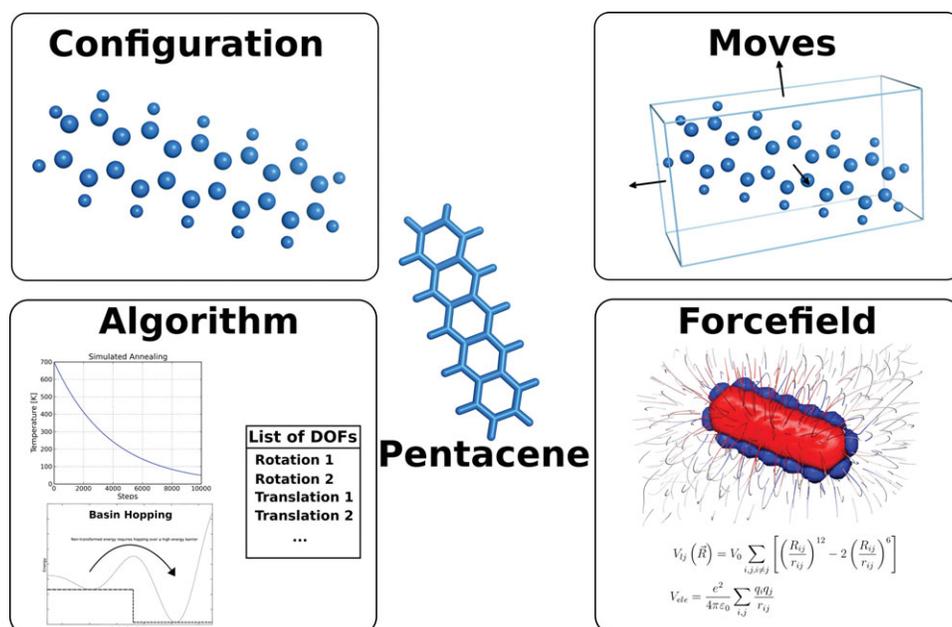


Figure 1. Systems definition in a SIMONA simulation. The simulation is encoded by a Configuration section (top left panel: position of atoms, here illustrated for a pentacene molecule), a Moves section (here specifying only rigid body rotation and translation for each molecule), a Forcefield section (implementing all interactions between atoms in the energy model), and an Algorithm section, encoding the protocol for the modification of configuration under the rules specified in the moves section. This hierarchical section permits implementation of very complex algorithms at the XML level without the need to modify the C++ backend. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

preprocessor splits the input file's information into abstract coordinates (the configuration object), assigns forcefield information (radii, forcefield parameters) and stores them in a Forcefield object, detects degrees of freedom (moves object), and implements a temporal simulation hierarchy for the simulations (algorithm).

In the following, we briefly discuss these sections of the input. Most researchers trying to set up a nonstandard system comprising elements not envisioned at the time the program was conceived are aware of the difficulties to implement simulations for such systems. SIMONA reduces all systems to assemblies of coordinates that are later manipulated by the simulation backend, such that all of these complexities are addressed in the generation of the input file. SIMONA presently reads protein data bank (PDB), protein coordinates, charge [Q] and radius (PQR), and MOL2 files and compressed versions thereof, and exploits a hierarchical atom-recognition scheme to assign atom types in a flexible, user-modifiable fashion, that later can be used to implement moves and interactions. The configuration section thus contains either one or several conformations of the system (e.g., a population as needed for simulation methods such as parallel tempering^[12,44] and evolutionary algorithms).

The moves section has its correspondence with constraint sections in molecular dynamics programs. In molecular dynamics, typically the entire system is flexible and the topology of the system is constrained by the interactions. In many Monte Carlo simulations, however, only specific degrees of freedom are being manipulated, such as dihedral angles in biomolecular systems for receptor-ligand docking simulations. The move section thus contains a list of elementary transformations to change the conformation of the system.

Interactions

Standard Interactions and Forcefields. Interactions in SIMONA are implemented in a hierarchical basis: in addition to the standard bonded potentials (bond-length, angles, dihedral angles), SIMONA currently implements the forcefield terms found in Table 1. SIMONA provides many of the standard non-bonded potentials used in molecular simulations, such as Lennard-Jones, 10–12-potentials, dihedral and angle dependent potentials, and so forth. In addition, Coulomb electrostatics and various generalized Born (GB) models, including a very efficient solvent accessible surface solver,^[45] have been implemented. Because there are only a few demonstrated examples of Monte Carlo simulations with explicit solvent models, incorporation of efficient implementations for implicit solvent models are important for Monte Carlo-based simulation suites. These physics-based

potential terms are complemented by a variety of constraint potentials, which are needed in specialized simulations, for example, by confining a ligand to the vicinity of a binding pocket. Some specialized potential terms, such as those required for the protein forcefields PFF01/PFF02, have also been implemented.^[29,46]

These individual potential terms can be combined into forcefields, some of which can already be selected in the graphical user interface (GUI), like protein forcefields PFF01/02^[46] and the potential used in FlexScreen for receptor-ligand docking.^[47] The parameterizations of some of these forcefields exist for proteins, DNA and can be adapted using the atom-type based assignment of parameters to small molecules. For parameterization of nonstandard systems, a parameter generator is supplied in the GUI, which can import partial charges from density functional theory (DFT) calculations and supply defaults for Van-der-Waals radii (which can be modified by the user). The FlexScreen potential includes the popular Autodock scoring function^[48] as a subset. The forcefields do not contain explicit symmetrization as reported in Mafolepsza et al.^[49] as all the currently implemented moves only perturb proper dihedral angles keeping improper angles intact.

To open SIMONA to a wide range of simulations, we have started to implement an import mechanism for GROMACS top files, such that systems parameterized by the GROMACS preprocessor can be imported into SIMONA. This makes all potentials currently accessible by GROMACS and structure based potentials by SMOG^[50] accessible in SIMONA. However, at present, parsing of the bonded terms (angles and bond distances) is not complete.

Implementing Complex Novel Potentials. Researchers wishing to implement and test new forcefields can exploit three unique

Potential	Functional form	References
Lennard Jones	$\sum_{i=0}^N \sum_{j/i}^N \varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$	Lennard Jones ⁷⁸
Coulomb electrostatics	$\frac{1}{4\pi\epsilon_0\epsilon_P} \sum_{i=0}^N \sum_{j/i}^N \left(\frac{q_i q_j}{r_{ij}} \right)$	
Implicit solvent models	$\sum_{i=0}^N \sigma_i A_i$	Eisenberg and McLachlan ⁷⁹
Generalized born electrostatics	$\frac{1}{8\pi\epsilon_0} \left(\frac{1}{\epsilon_W} - \frac{1}{\epsilon_P} \right) \sum_{i,j}^N \frac{q_i q_j}{\sqrt{r_{ij}^2 + R_i R_j} \exp(-r_{ij}^2 / 4R_i R_j)}$	Brieg et al ⁵⁹ and Still et al ⁶⁰
Constraint potentials	General distance dependent $f(r_i, r_j)$, Disulfide Potentials, Morse Potential, and so forth	Kondov et al. ⁸⁰
Compound protein forcefield PFF02	$E_{\text{PFF02}} = E_{\text{LJ}} + E_{\text{ELE}} + E_{\text{SASA}} + E_{\text{HB}} + E_{\text{Torsion}}$ $E_{\text{SMOG}} = E_{\text{Bonds}} + E_{\text{Angles}} + E_{\text{Dihedrals}}$	Herges et al, Verma et al. ^{39,40}
Go models	$+ \sum_{\text{Contacts}}^N \varepsilon_C \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_{\text{Non contacts}}^N \varepsilon_{\text{NC}} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12}$	Noel et al ⁴³

Parameters for these forcefield terms are assigned in the preprocessor. Although each of the potentials has a default parameter set, the parameters can be changed on a per atom basis.

mechanisms: first, SIMONA uses a generic function parser, implemented by the FunctionParser library that will convert functions written in plain text into the XML input file, for example, "sin(sqrt(x*x+y*y))" into arbitrary novel pair potentials in SIMONA, that can be applied to arbitrary subsets of atoms in the system. A second, more complex, variant of this scheme implements general (3,4,5,...N)-body interactions, which may be used for complex new potentials, such as coordination chemistry for transition metals or reactive forcefields. It is important to note that these novel potentials can be implemented without modifying or recompiling the C++ backend. Once a novel potential has been validated, it can be implemented into the core of SIMONA by using class derivations both on the Python and C++ sides of the code to implement the new potential efficiently.

OpenCL Support for General N-Body Forcefields. SIMONA contains OpenCL support for generalized N-Body forcefields. In extensive tests, we observed speedups of 130 compared to a single CPU for systems with more than 15,000 atoms.^[51] The SIMONA OpenCL code was tested on AMD and nvidia GPU and Intel and AMD CPU architectures. A brief illustration of the SIMONA parallelization strategy on GPU is given in Figure 2.

Algorithms

Standard Algorithms. The Algorithm section is the most complex section of the SIMONA input. In contrast to molecular dynamics simulations, where typically most of the system is fully flexible, many Monte Carlo schemes use complex nested algorithms to modify the conformation. For standard applications, the user can just use the algorithm prototypes implemented in the GUI, including nested Monte Carlo simulations, simulated annealing, basin hopping,^[52-54] and evolutionary algorithms described below.^[55] In addition, we established a system for threading or running on clusters via message parsing interface (MPI). In the following, we discuss preparation of standard simulations via the GUI frontend and of threaded evolutionary algorithms as an example of a more complex simulation.

The XML file generated by the python preprocessor implements a programming language that permits implementation of novel simulation protocols with relative ease. This language features elements for loop building, nesting of subtasks, and conditional execution, making it possible to implement very complex simulations flows, without recoding the C++ kernel. A standard Metropolis Monte Carlo simulation in this language is illustrated in Figure 3.

Designing and Using Complex Simulation Workflows. As a more complex example, we illustrate an evolutionary strategy that

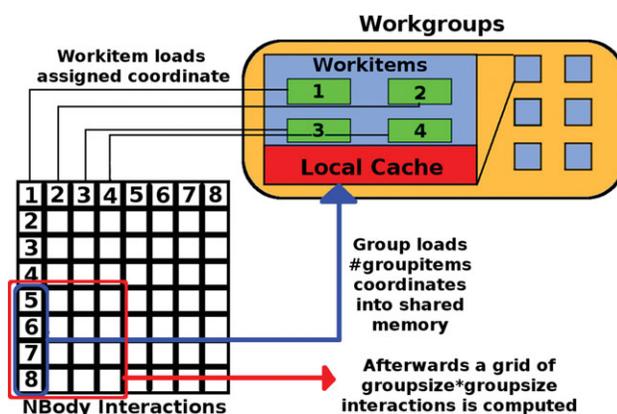


Figure 2. Scheme of the OpenCL NBody parallelization. Step 1: an atom index is assigned to each OpenCL workitem (i.e., shader processor). Each workitem loads its specific parameters/coordinate. Step 2: the number of atoms is partitioned into intervals of length group size (typically 2^N with $N > 4$, set by the OpenCL library). Shaders of each workgroup load one interval of coordinates into shared local memory. Step 3: the workitems iterate synchronously over the just loaded coordinate segment, calculate the interaction energy, and sum it into a local register. Step 4: on finishing the loop, the algorithm synchronizes all workitems and loads the next interval until all intervals have been loaded. Step 5: the sums are written in a result vector and transferred to the host. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

```

<Algorithm>
  <RepeatedMove>
    <repeats>12000</repeats>
    <tstart>450.0</tstart> <tend>150.0</tend>
    <tscaling>geometric</tscaling>
    <TransformationSequence>
      <ConditionalTransformation>
        <TransformationChoice>
          <SetTranslationRandom wt="1.0">
            <SetRotationRandom wt="1.0">
              <TransformationChoice wt="20.0">
                <SetDihedralRelativeRandom wt="1.0">
                  <SetDihedralRelativeRandom wt="1.0">
                    <SetDihedralRelativeRandom wt="1.0">
                      ....
                    </TransformationChoice>
                  </TransformationChoice>
                </MetropolisAcceptanceCriterion>
                <energymodel_nr>0</energymodel_nr>
                <kB>0.0019858775</kB>
              </MetropolisAcceptanceCriterion>
            </ConditionalTransformation>
          <TransformationSequence>
            <EnergyOutput>
            <ConfigurationOutput>
          </TransformationSequence>
        </TransformationSequence>
      </RepeatedMove>
    </Algorithm>

```

for i in 0..12000 do

```

#geometrical temperature scaling
temperature = temperature_before*temp_factor
Do all transformations in a row
Create a copy of the current configuration
Apply following transformations to copy
Choose one transformation of
  A random translation (with a weight of 1)
  A random rotation (with a weight of 1)
  Choose one (dihedral perturbation) (weight 20)
    A dihedral perturbation (weight 1)
    A dihedral perturbation (weight 1)
    A dihedral perturbation (weight 1)
    ... (complete List of dihedral angles)

Evaluate energy difference between copy and original
Accept or reject based on Metropolis criterion

Do all transformations in a row
Output the energy
Output the snapshot for a trajectory

```

Figure 3. Algorithm section of a Metropolis Monte Carlo implementation translated into pseudocode. In every repeat of the RepeatedMove, a Transformation is carried out (in this case a displacement, rotation or dihedral rotation). The resulting state is accepted or rejected by the Metropolis acceptance criterion. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

we have used to *de novo* fold proteins with up to 60 amino acids.^[55] In this method, we explore the free-energy surface in many parallel Monte Carlo simulations using a population selection scheme that balances energy improvement and population diversity (see Fig. 4). Because the standard evolutionary algorithm was found to tend to freeze the population after some time, we have implemented a multitemperature general-

ization, which evolves several populations at different (constant) temperature. The XML code for this algorithm, which will automatically parallelize via MPI on as many nodes as are available, is illustrated in Supporting Information, Figure S1. Comparison with Figure 3 for the standard Monte Carlo algorithm illustrates that the much more complex algorithm can be implemented very easily.

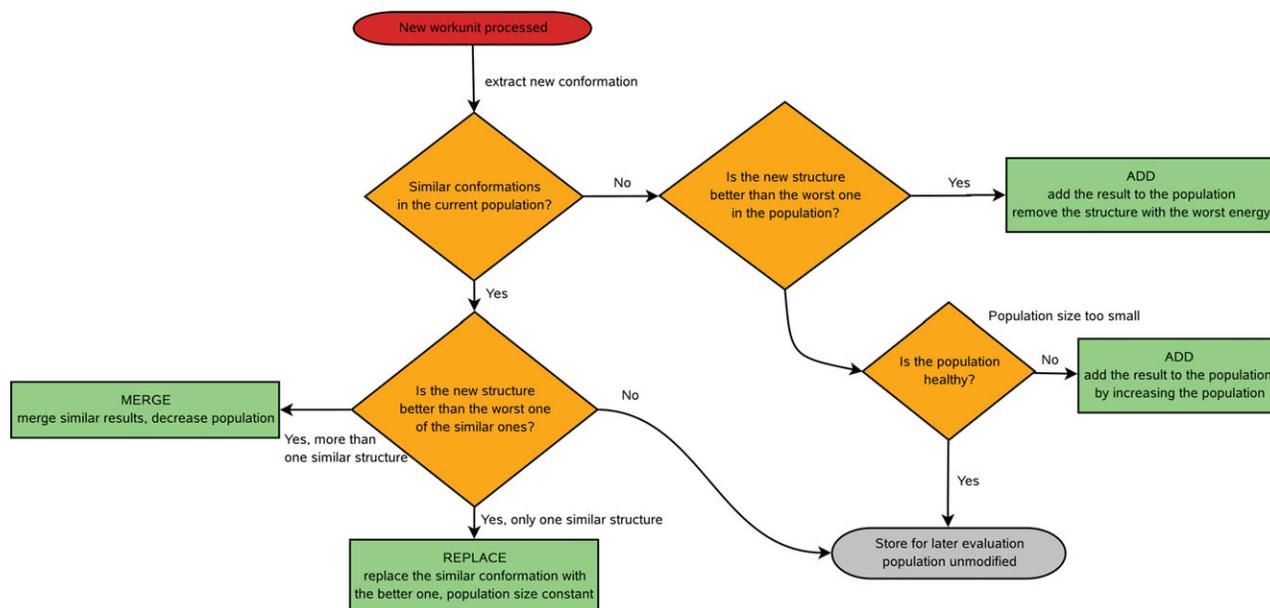


Figure 4. Flowchart of the evolutionary algorithm. The Evolutionary Algorithm focuses on evolving structures in a population by balancing structural diversity and energy optimization. After randomly annealing a single structure from the population, the conformation is extracted and compared to the population. In case of similar conformations worse of energy, these are discarded from the population. If the new conformation is energetically viable and structurally dissimilar to the existing ones, it is included in the population. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Selected Applications

In the following, we report several applications of SIMONA that were chosen to illustrate the versatility of SIMONA. SIMONA implements both the protein-simulation algorithms^[56–58] and force fields^[29,46] that were used to fold over 30 small proteins from unfolded conformations to the native state, and most components of the FlexScreen receptor-ligand docking program, which has been used in a variety of applications.^[47,59,60] In a materials-science application, SIMONA was already used to explain selective separation of carbon nanotubes with designed polymers.^[61]

Here, we illustrate applications of SIMONA that go beyond our previous investigations: in the first application, we observe the repeated folding and unfolding of the trp-cage protein^[62–66] in unbiased Monte Carlo simulations using an all atom force field with a GB model.^[67,68] In the second protein-related application, we use the BOINC^[69] server POEM@HOME^[70] to perform 200,000 protein–protein docking simulations on 35,000 computers concurrently.

Next, we turn to modeling receptor ligand interactions: computer-based drug development improves the time and cost efficiency of a modern drug design process. Molecular docking simulations are often used in high-throughput screening to preselect a promising subset of possible active ingredients, but the correlation between experimental affinities and scoring functions is typically weak. We hope that SIMONA, which can implement both types of scoring functions, can contribute to close the gap between high-throughput and accuracy demands in receptor-ligand binding simulations. Here, we report results of a single-stage massively parallel docking simulation for the biotin-streptavidin complex, and a multi-stage flexible receptor-ligand docking protocol^[47,59,60] for five pharmaceutically relevant receptors which converge to a mean deviation of 0.98 Å (range: 0.38–1.45 Å) for the lowest energy pose found in the simulation in comparison to the experimental structures.

A final application example deals with the simulation of growth processes in an organic material that is used in OLED. Prior work with SIMONA protocols included applications in nanoparticle growth,^[71] sorting nanotubes,^[61] OLED/organic photovoltaics,^[38] and many other fields. Growth and structure formation processes, in all aforementioned application areas, occur on timescales beyond a millisecond in rather large systems and are therefore challenging to all atom molecular dynamics simulations. In many cases, kinetic information is not crucial to interpret experiments, for example, in assessing electron/hole mobilities for thin amorphous films, such that MC-based methods may be the method of choice.

Protein folding

The 20 amino acid trp-cage miniprotein has long been used to demonstrate the functionality of protein-folding protocols.^[62–66] Here, we do not use PFF01/02,^[46] but a forcefield comprising of the Amber99^[22,23,25] Lennard-Jones and Coulomb terms. Electrostatic interactions are described in an implicit solvent model

comprising a GB term using Powerborn^[67] for born radii computation and Stills GB formula^[68] with an exponential factor of 4. Nonpolar solvent interactions were modeled using a uniform solvent accessible surface area term^[45] with a tension coefficient of 5.42 cal/Å². Backbone and sidechain dihedral angles are automatically selected by the preprocessor as degrees of freedom, the Amber99SB^[22] dihedral potential for these dihedrals was also added.

We first unfolded the protein in a high-temperature simulation and then performed 10 independent Monte Carlo simulations comprising 10⁷ steps each at T = 300, 350, 400, 450, 500K. For T = 450 K, we observe repeated folding and unfolding events between an unfolded ensemble with a mean root mean square deviation (RMSD) of 4.9 Å to the native conformation (see Fig. 5, conformations with RMSDs below 3 Å were classified to belong to the native ensemble). In these 10 independent simulations, we observe folding-unfolding transitions in either direction approximately every 250,000 MC steps, which correspond to 90 min CPU on a standard single-core PC. At lower/higher temperatures, the occupancy of the folded/unfolded ensembles increases/decreases (see Supporting Information, Fig. S2). These simulations demonstrated that the conformational landscape of small peptides can be efficiently sampled using standard protocols in a very short amount of time.

Protein-protein docking

Protein–protein interactions mediate many important signaling processes^[72] in the cell and are therefore studied widely with many methods, including simulations. The CAPRI exercises regularly assess the state-of-the-art in protein complex prediction.^[73] Here, we report a simple simulation, exploiting the massively parallel POEM@HOME network,^[70] for contact prediction. Simulations were performed with the PFF02, which was successfully used previously for alanine screening,^[74] suggesting it may also be reliable for the prediction of the conformation of protein complexes.

We studied the dimerization of the fire ant venom allergen (pdb code: 2YGU).^[75] The protein occurs as a homodimer in its native state and comprises 125 amino acids per chain. The crystal structure was resolved to 2.6 Å, and the complex is stabilized via a disulfide bond that links the CYS21 of both chains. In forming such a complex, the question arises whether the native conformation of the complex is a unique free energy minimum in the absence of the disulfide bridge, or selected via conformational selection from a multitude of competing structures.^[76,77]

Starting from the crystal structure, we generated 200,000 starting structures by offsetting the two structures by 15 Å in a random direction and rotating them about an arbitrary axis by a random angle. Using POEM@HOME, these starting structures were then annealed from 700 to 50 K by using the PFF02 potential in simulated annealing simulations with a geometric cooling schedule running for 50,000 Monte Carlo steps each. Moves incorporate rigid body center of mass translations uniformly distributed between 0 and 1.4 Å in a random direction and random rotations by up to 10° around an arbitrary

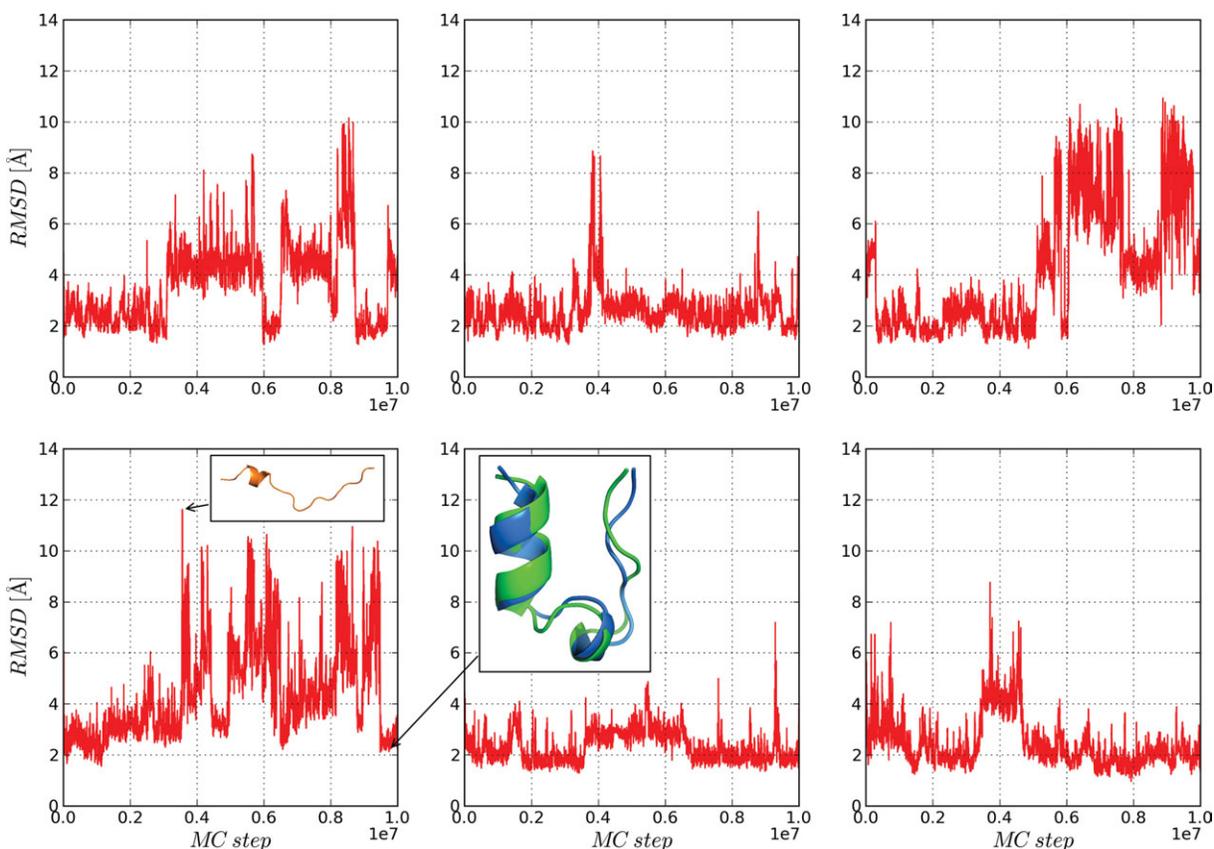


Figure 5. Six independent runs of the folding of protein 1L2Y at a temperature of 450 K. Multiple folding/unfolding events can be observed in the runs. The lower left panel shows both a snapshot of a completely unfolded chain and a structure folded to experimental resolution at 2.2 Å RMSD.

axis. If the two distinct chains move away further than 15 Å, a biased move toward the center is applied. These moves attempt a uniformly distributed random step between 0.9 and 2.0 Å toward the center of mass of the other chain. Resulting structures were found in a wide energy range, replicating the experience of other docking protocols that short simulations

started far from the experimental conformation cannot find the correct binding pose.

The lowest energy configuration features a RMSD of 1.0 Å to the experimental structure and is offset by an energy gap of 5.6 kcal/mol from the next lowest energy conformation, which has a RMSD of 15.5 Å to the experimental conformation.

The interchain disulfide bridge (not included in the forcefield) can only form correctly in the cluster of lowest energy structures shown in the left panel of Figure 6. The forcefield PFF02 favored complex conformations which envelop the docking interface near the actual docking site of 2YGU, but overestimates energy differences (because it was developed for protein structure prediction).

The lowest energy structure selects the native conformation in the absence of a potential modeling the disulfide bridge, because most of the surface of the docking interface is covered. It is interesting to note that there are few competing low-energy

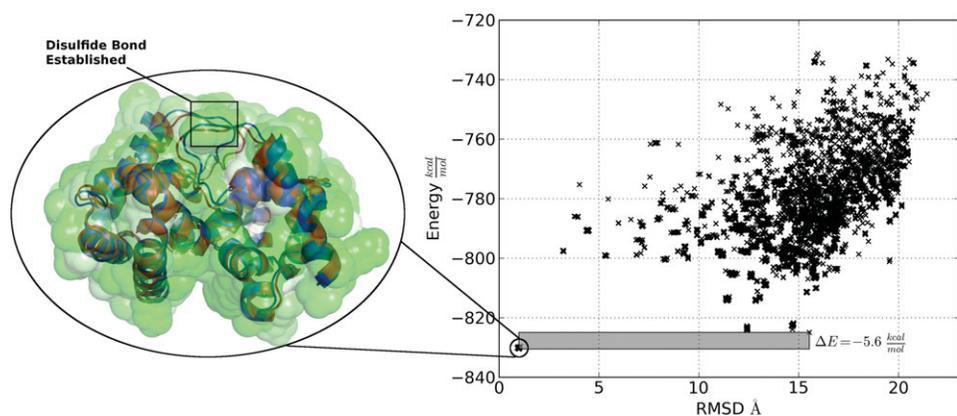


Figure 6. Results of the rigid docking prediction of protein 2YGU. Left: overlay of the predicted (blue) and the native (red) structure of protein 2YGU. The lowest energy structure features a backbone RMSD of 1.0 Å to the experimental conformation. The rectangle indicates the position of the disulfide bridge in the experimental structure, but there are no corresponding potentials in the simulation. The surface shading green to white indicates an increasing hydrophobicity of the protein surface. Light surfaces are hydrophobic, green surfaces hydrophilic. Right: distribution of the final structures below an energy of -720 kcal/mol. The lowest energy conformation has a gap by 5.6 kcal/mol to the next lowest energy structure of different docking topology.

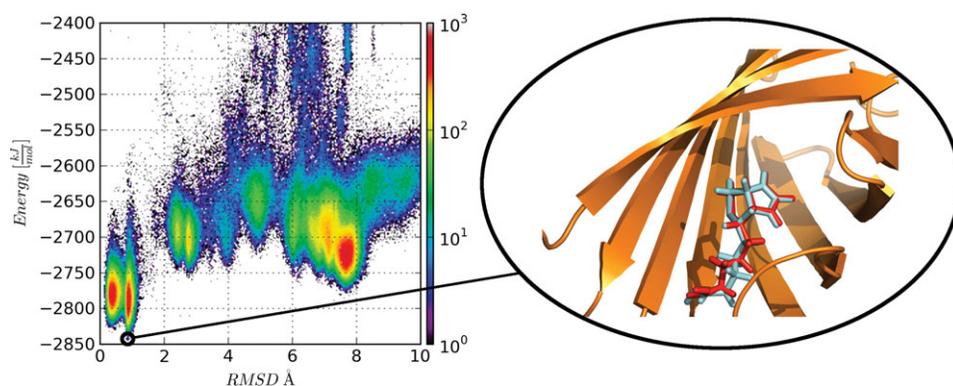


Figure 7. Results of the streptavidin biotin docking simulation. Left: density plot of the RMSD versus Energy map for 660.720 structures (limited by a max. RMSD of 10 Å and a max. energy of 2400 kJ/mol). The RMSD values were calculated including all ligand atom coordinates without a prior alignment. Right: overlay between the energetically best structure (blue) and the native (red) structure of the protein ligand complex 1STP. The predicted complex has a RMSD distance of 0.88 Å to the native one. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

structures featuring a low RMSD around the lowest energy structure. Supporting Information, Figure S3 shows RMSD/energy snapshots during the simulations. Near the correct docking pose (RMSD < 4 Å) a sharp funnel toward the correct docking pose is visible explaining this isolation of states with small RMSD.

Receptor ligand docking

Prediction of receptor-ligand conformations and computation of affinity estimates is an important present-day application of molecular simulation methods in drug discovery. Although most presently methods to estimate the relative or absolute free energy of binding are mostly based on molecular-dynamics sampling, most docking methods rely on stochastic optimization, often using physics-inspired sampling techniques.^[78–80] In the following, we present two applications of SIMONA for receptor-ligand docking using two distinct approaches: a whole-surface approach that samples the entire protein surface without any prior knowledge about the docking pocket and a cascaded docking strategy similar to that previously used in FlexScreen that samples only the vicinity of a known binding pocket.

Whole Surface Sampling. The streptavidin-biotin complex (PDB-ID: 1STP) is one of the most common examples for protein-ligand docking, as its binding energy is one of the highest measured for noncovalent binding systems,^[81] making it a good benchmark system. The SIMONA GUI contains already a specialized menu to set up docking simulations. Here, we used our BOINC volunteer network POEM@HOME to generate an ensemble of 883.244 structures of the streptavidin-biotin complex.^[70] Every single simulation relaxed a randomly placed ligand conformation at constant temperature $T = 300\text{K}$ performing 300.000 Monte Carlo steps. Although the ligand was completely flexible in dihedral space, we kept the receptor backbone dihedrals fixed and allowed the rotation of 29 side-chain dihedrals around the binding pocket. All energies were evaluated in the FlexScreen forcefield.^[47] The low-energy subset of the results of this brute-force sampling approach,

including all 660.720 structures with an energy < 2400 kJ/mol (absolute energy) and all atom RMSD < 10 Å is shown in Figure 7. The energetically lowest structure has a RMSD value of 0.88 Å to the native structure.

Cascaded Docking Approach. The whole-surface docking approach may be used *ad hoc* to sample the protein surface, but is not viable for large-scale screening of ligand databases for drug development. Following the FlexScreen protocol, we applied a four-stage, cascaded docking approach for four pharmaceutically relevant receptor-ligand complexes. Initially, the ligand is

placed into the docking pocket, displaced randomly by maximum displacement of 2 Å and randomly perturbed 10,000 times by rotations around single bonds (which are identified by an automated procedure). These structures enter stage 0 of the cascade as starting conformations. The idea of the cascades (see Table 2 for details) is to perform many short simulations on a large possible set of ligand poses, then to select the lowest energy poses at the end of the stage. These poses are passed to the next stage and subjected to somewhat longer simulations until only a few conformations are subjected to long simulations in the last stage. As in the standard FlexScreen protocol, individual simulations used the stochastic tunneling method.

Using this protocol, we investigated ribonuclease-guanylic acid complex 1RNT, spleen tyrosine kinase complexed with staurosporin 1XBC, antibody 21D8 complexed with hapten 1C5C, an L-arabinose-receptor complex 1ABP, a cytidin-ribonuclease complex 1ROB, and a streptavidin-biotin complex 1STP. All binding poses were predicted to experimental resolution, as shown in Figure 8. The RMSDs of the predicted structures ranged from 0.31 to 1.45 Å, and the lowest energy conformations agree well with the experimental structures.

Cluster growth of organic materials

Not only biomolecular systems, but also systems from the material sciences require efficient simulation methods.^[82] In

Table 2. Parameters used in the cascaded docking approach.

Stage	No. of poses	No. of steps	Forcefield E_i
0	96	1000	E_{LJ}
1	96	5000	$E_{LJ} + E_C + E_{HB} + E_{ISE}$
2	16	30,000	$E_{LJ} + E_C + E_{HB} + E_{ISE}$
3	8	75,000	$E_{LJ} + E_C + E_{HB} + E_{ISE}$

Although the first stage only used a Lennard Jones (E_{LJ}) potential to remove steric clashes, simulations of further stages used a full physical potential comprising also an Electrostatics (E_C), Hydrogen Bond (E_{HB}) and implicit solvent (E_{ISE}) potential.

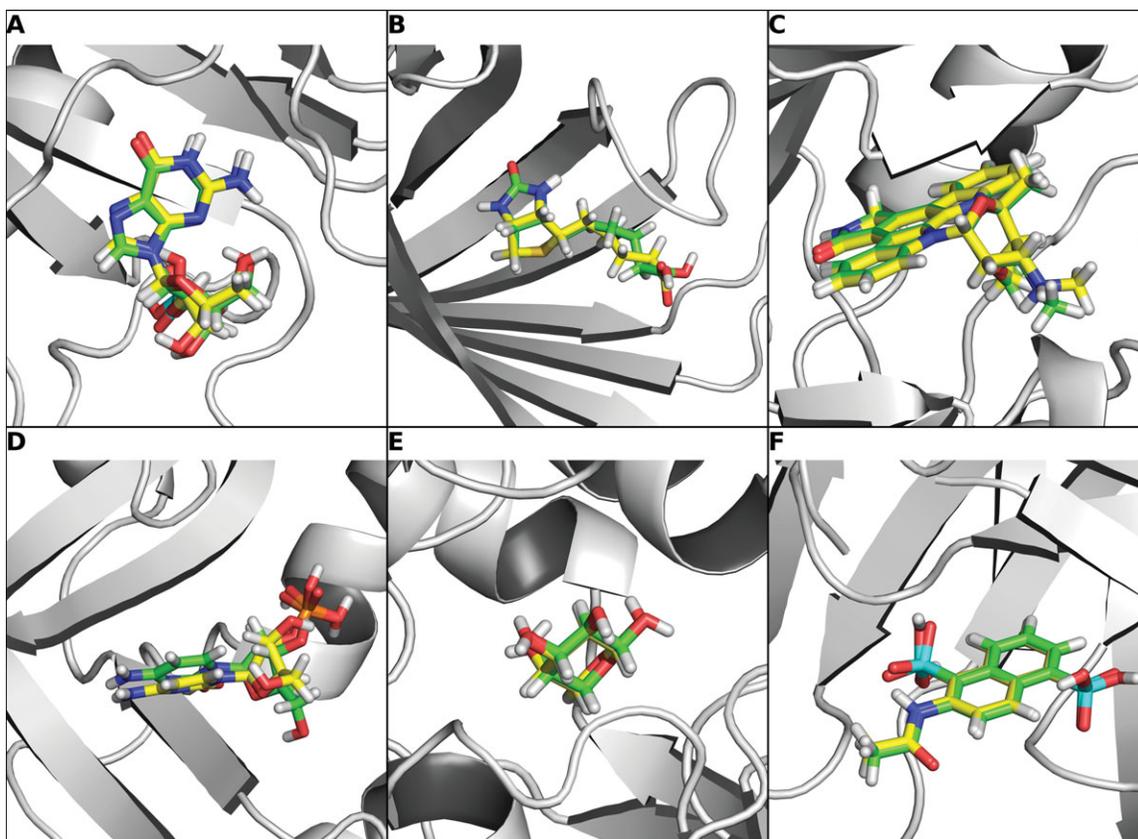


Figure 8. Docking poses of lowest energy after applying the cascaded docking pipeline. The images show the experimental reference structure in green and the prediction in yellow. Following are the respective PDB IDs of the benchmark complexes and the RMSDs toward the respective experimental structure: A 1RNT, 1.45 Å B: 1STP, 0.95 Å C: 1XBC, 0.38 Å D: 1ROB, 1.45 Å E: 1ABE, 0.31 Å F: 1C5C, 1.34 Å. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

contrast to biomolecular systems, where simulations are presently dominated by biopolymers with largely standardized components (DNA, RNA, and proteins), the materials sciences offer a vast variety of components and simulation problems.

To focus on one specific example, we investigate a cluster formation in pentacene, a material used in OLED. OLED are made from multiple layers of materials, comprised of metals, oxides, polymers, and/or small organic molecules, which emit photons when subjected to an external field. The first step in modeling the complex processes in OLED at the molecular scale is the determination of the morphology of the layers and their interfaces. Here, we look at cluster growth of pentacene, $C_{22}H_{14}$, one material presently used in the electroluminescent layer of small-molecule OLED.^[83] Interest in pentacene grew especially after the discovery that bulk and thin-film pentacene is a p-type organic semiconductor. Pentacene based materials feature mobilities similar to that of amorphous silicon, with mobilities of $\sim 5 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$ for polycrystalline pentacene films.

We have performed cluster growth simulations on pentacene systems starting from a random assembly of molecules. As input, we took the pentacene morphology from the Cambridge Crystallographic Data Centre (CIF ID 2012157)^[84] and generated partial charges of pentacene using DFT.^[85] We then prepared a SIMONA Parameter File (SPF) comprising Lennard-Jones parameters and the partial charges from the DFT simulation. The simulation is

prepared using the Monte Carlo preprocessor of the GUI by loading both spf and the mol2 file and setting the environmental parameters (temperature, forcefields, rigid-body moves). We then use the structure cloner (found under Expert Settings) and generate 15,000 pentacene replicas with this parameterization and enable the OpenCL implementations of the forcefields.

Starting from this setup, we ran 400,000 simulations with 500,000 steps each with a constant temperature of 300.0 K on the distributed POEM@HOME architecture.^[70] In the following, we analyze the cluster topologies that emerged from the simulations: Most clusters feature layers of face-on-edge pentacene molecule stacked with other layers in a direct or interleaved fashion as seen in Figure 9. To quantify these results, we looked at two criteria: cluster size and emergence of order.

After clustering all samples using a modified version of the power diagram described in Klenin et. al.,^[45] we determine the distribution of cluster sizes (Fig. 10). The figure demonstrates a near-exponential distribution of the cluster size over five orders of magnitude. This data demonstrate that clustering occurs but gives no information about the order of the molecules within a cluster. The latter was quantified by using the relative alignment of stacks of neighboring molecules, as defined by the relative orientation of their inertia tensor, and counting the number of molecules within a stack. The distributions of these stack trails the cluster distribution but

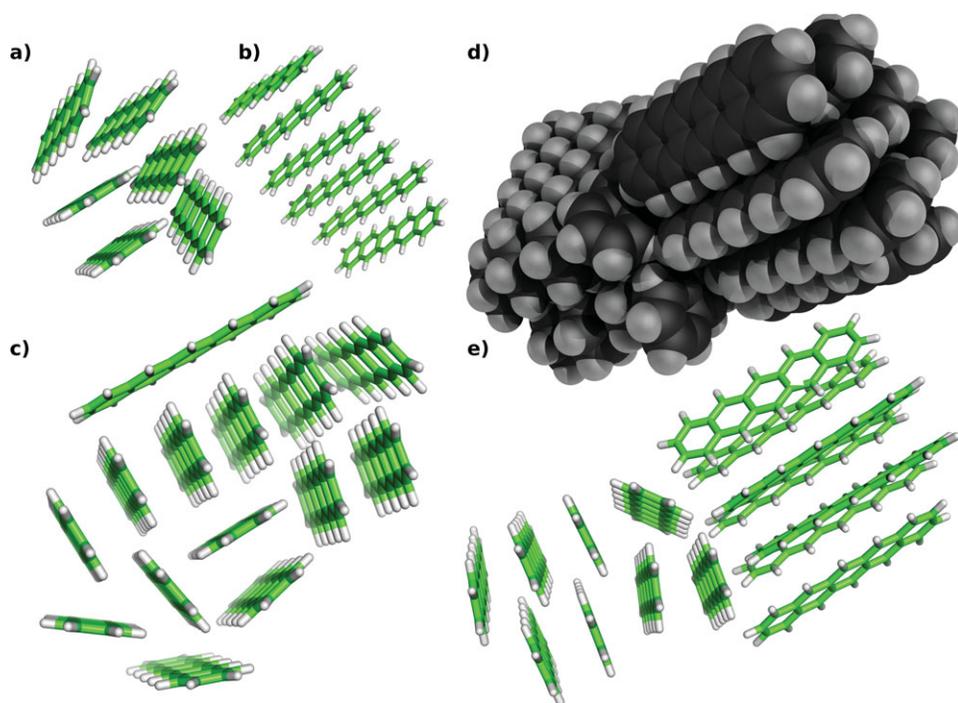


Figure 9. Various examples characteristic for the cluster topologies found in the simulations of pentacene clustering. a) Initial stage of building a herringbone conformation. b) Direct Face on Face pi stacking conformation. c) Unordered pentacene bundle with pentacene ordered in z direction. d) and e) Two pentacene cluster directions merged into a single cluster, space fill, and licorice visualizations. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

demonstrates that a significant fraction of the molecules within the cluster has attained local order.

Discussion

Driven by the growth of the available computational resources, molecular simulation methods have become increasingly important in complementing experiment and theory in understanding and predicting the properties of molecular and nanoscale systems in the life- and materials sciences. Applications of Monte Carlo-based methods are presently reported less frequently than molecular dynamics-based investigations, in part because few adaptable, general-purpose Monte Carlo simulation packages are available. Because move generation in Monte Carlo methods is fundamentally different than in molecular dynamics methods, standalone Monte Carlo implementations have an efficiency advantage over the adaptation of molecular dynamics packages for Monte Carlo simulations. The lack of general-purpose Monte Carlo packages has hampered the development of efficient novel Monte Carlo algorithms in comparison to the vast array of tools available to extract thermodynamic information, for example, calculation of free-energy differences, from molecular dynamics based methods. This is unfortunate, because in many of these applications, molecular dynamics simulations, which all struggle with difficulties to sample sufficiently long timescales, serve merely as a workhorse to explore the conformational space.

For this reason, we have implemented SIMONA, a modern adaptable general-purpose Monte Carlo simulation package that

allows users with different requirements to easily conduct Monte Carlo simulations for molecular and nanoscale systems or to develop novel Monte Carlo-based protocols. To facilitate simulations of complex nonstandard systems, we have implemented the package with a script-based preprocessor (Python) which generates a topology file (XML) for subsequent simulation with an efficient C++-based backend, which uses architecture-specific optimization for most low-level operations (Eigen/OpenCL). Parallelization is supported via MPI and some, though not all force-field components can exploit GPUs. Simulations using massively distributed grid-type computational environments are also possible.

Researchers who want to use the protocols already implemented in the package can prepare their simulation using a graphical user interface and import mechanisms for topology

files of widely used programs, such as GROMACS, are already partially implemented. To facilitate force fields development, SIMONA supports rapid prototyping of novel potentials at the preprocessor level. Expert users can use class derivation both at the preprocessor and backend side to implement new algorithms and forcefields.

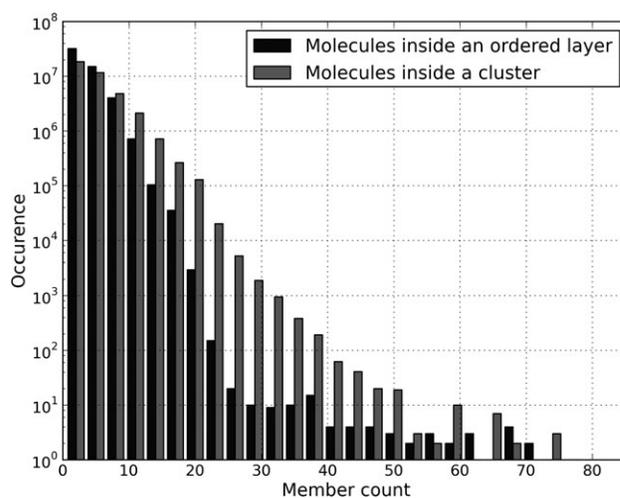


Figure 10. Histogram of cluster sizes and number of pentacene molecules in ordered layers. For small counts, both distributions decrease exponentially. Although pentacene forms a crystal in its native conformation, the Monte Carlo simulation freezes and crystals stop growing. As only rigid body moves of single pentacene molecules are allowed, it gets exponentially harder to move a complete cluster.

SIMONA implements the forcefields and algorithms developed in our group in the last decade for protein folding (PFF01/PFF02) and receptor-ligand docking (FlexScreen), and makes these methods available in an easy-to-use package. Standard simulations using these methods for protein conformational sampling and receptor-ligand docking can be set up easily using the GUI. Here, we have reported illustrative applications of novel components: Using a GB implicit solvent model, we demonstrated reproducible folding events of the 1L2Y peptide in 50 independent simulations, each of which requires only a few days on a single core. We also reported simulations on protein-protein docking comprising 200,000 distributed simulations using PFF02, which selected the experimental binding pose by energy to good accuracy. Similarly, good accuracy was reported in sample simulations for receptor-ligand docking for several pharmaceutically relevant receptor-ligand complexes. In a final application, we investigated the growth of pentacene clusters as an example for an application from the material sciences.

We have chosen these applications to demonstrate the versatility of the package and hope that they are the starting point for many more applications and a fruitful development of novel Monte Carlo-based methods. Given the wide range of available methods and applications, no single group can hope to develop and maintain a software package covering even a fraction of the possible applications. We therefore hope that this first version of SIMONA, which is distributed in source code for free academic use, can serve as the nucleus around which a community of researchers can implement existing algorithms and develop new methods in an adaptable framework without having to start from scratch for new applications. SIMONA is available for download under <http://int.kit.edu/nanosim/simona>.

Acknowledgments

We thank Felix Ehrler, Julia Setzler, Yvonne Klapper, Carolin Seith, Louisa Scholz, Tobias Neumann, and Abhinav Verma for testing, and Felix Ehrler for his work on the online documentation. We thank the participants of POEM@HOME for their continued support.

Keywords: Monte Carlo simulation · protein folding · receptor ligand docking · protein docking · cluster growth

- [1] D. E. Shaw, P. Maragakis, K. Lindorff Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, W. Wriggers, *Science* **2010**, 330, 341. Available at: <http://www.sciencemag.org/content/330/6002/341.abstract>.
- [2] N. P. King, W. Sheffler, M. R. Sawaya, B. S. Vollmar, J. P. Sumida, I. André, T. Gonen, T. O. Yeates, D. Baker, *Science* **2012**, 336, 1171. Available at: <http://www.sciencemag.org/content/1336/6085/1171.abstract>.
- [3] R. O. Dror, R. M. Dirks, J. P. Grossman, H. Xu, D. E. Shaw, *Annu. Rev. Bio phys.* **2012**, 41, 429. Available at: <http://dx.doi.org/410.1146/annurev-biophys-042910-155245>.
- [4] G. G. Dodson, D. P. Lane, C. S. Verma, *EMBO Rep.* **2008**, 9, 144. Available at: <http://dx.doi.org/110.1038/sj.embor.7401160>.
- [5] C. I. Blaga, J. Xu, A. D. DiChiara, E. Sistrunk, K. Zhang, P. Agostini, T. Miller, L. F. DiMauro, C. D. Lin, *Nature* **2012**, 483, 194. Available at: <http://dx.doi.org/110.1038/nature10820>.
- [6] J. A. Katan, C. Dekker, *Cell* **2011**, 147, 979. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0092867411013559>.
- [7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *J. Chem. Phys.* **1953**, 21, 1087. Available at: <http://dx.doi.org/10.1063/1081.1699114>.
- [8] K. Binder, In *Mathematical Tools for Physicists*; Wiley VCH Verlag GmbH & Co. KGaA; Germany, **2006**; pp. 249-280.
- [9] V. Knecht, S. J. Marrink, *Biophys. J.* **2007**, 92, 4254. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0006349507712274>.
- [10] Y. Sugita, Y. Okamoto, *Chem. Phys. Lett.* **1999**, 314, 141. Available at: <http://www.sciencedirect.com/science/article/pii/S0009261499011239>.
- [11] W. Nadler, U. H. E. Hansmann, *Phys. Rev. E* **2007**, 76, 057102. Available at: <http://link.aps.org/doi/10.1103/PhysRevE.76.057102>.
- [12] R. H. Swendsen, J. S. Wang, *Phys. Rev. Lett.* **1986**, 57, 2607. Available at: <http://link.aps.org/doi/10.1103/PhysRevLett.57.2607>.
- [13] C. Geyer, *Stat. Sci.* **1992**, 7, 473. Available at: <http://dx.doi.org/410.2307/2246094>.
- [14] K. Hukushima, K. Nemoto, *J. Phys. Soc. Jpn.* **1996**, 65, 1604.
- [15] U. H. E. Hansmann, *Chem. Phys. Lett.* **1997**, 281, 140. Available at: <http://www.sciencedirect.com/science/article/pii/S0009261497011986>.
- [16] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory. Comput.* **2008**, 4, 435. Available at: <http://dx.doi.org/410.1021/ct700301q>.
- [17] 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. Available at: <http://dx.doi.org/1610.1002/jcc.20290>.
- [18] 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. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popovic, 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. Available at: <http://www.hubmed.org/display.cgi?uids=21187238>.
- [19] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kalé, K. Schulten, *J. Comput. Chem.* **2005**, 26, 1781. Available at: <http://dx.doi.org/1710.1002/jcc.20289>.
- [20] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caffisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, M. Karplus, *J. Comput. Chem.* **2009**, 30, 1545. Available at: <http://dx.doi.org/1510.1002/jcc.21287>.
- [21] A. Irbäck, S. Mohanty, *J. Comput. Chem.* **2006**, 27, 1548. Available at: <http://dx.doi.org/1510.1002/jcc.20452>.
- [22] V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, C. Simmerling, *Proteins: Struct. Funct. Bioinformatics* **2006**, 65, 712. Available at: <http://dx.doi.org/710.1002/prot.21123>.
- [23] T. E. Cheatham, P. Cieplak, P. A. Kollman, *J. Biomol. Struct. Dyn.* **1999**, 16, 845. Available at: <http://dx.doi.org/810.1080/07391102.07391999.10508297>.
- [24] K. Lindorff Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, D. E. Shaw, *Proteins: Struct. Funct. Bioinformatics* **2010**, 78, 1950. Available at: <http://dx.doi.org/1910.1002/prot.22711>.
- [25] J. Wang, P. Cieplak, P. A. Kollman, *J. Comput. Chem.* **2000**, 21, 1049. Available at: [http://dx.doi.org/1010.1002/1096-1987X\(200009\)200021:200012<201049::AID-JCC200003>200003.200000.CO;200002.F](http://dx.doi.org/1010.1002/1096-1987X(200009)200021:200012<201049::AID-JCC200003>200003.200000.CO;200002.F).
- [26] W. L. Jorgensen, D. S. Maxwell, J. Tirado Rives, *J. Am. Chem. Soc.* **1996**, 118, 11225. Available at: <http://dx.doi.org/11210.11021/ja9621760>.
- [27] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, D. Baker, *Methods Enzymol.* **2004**, 383, 66. Available at: <http://www.hubmed.org/display.cgi?uids=15063647>.
- [28] F. Eisenmenger, U. H. E. Hansmann, S. Hayryan, C. K. Hu, *Comput. Phys. Commun.* **2001**, 138, 192.

- [29] T. Herges, W. Wenzel, *Biophys. J.* **2004**, *87*, 3100.
- [30] K. T. Simons, R. Bonneau, I. Ruczinski, D. Baker, *Proteins* **1999**, *37* (Suppl 3), 171. Available at: <http://www.pubmed.org/display.cgi?uids=10526365>.
- [31] P. Bradley, D. Chivian, J. Meiler, K. M. S. Misura, C. A. Rohl, W. R. Schief, W. J. Wedemeyer, O. Schueler Furman, P. Murphy, J. Schonbrun, C. E. M. Strauss, D. Baker, *Proteins* **2003**, *53*(Suppl 6), 457. Available at: <http://www.pubmed.org/display.cgi?uids=14579334>.
- [32] D. Chivian, D. E. Kim, L. Malmstrom, J. Schonbrun, C. A. Rohl, D. Baker, *Proteins* **2005**, *61*(Suppl 7), 157. Available at: <http://www.pubmed.org/display.cgi?uids=16187358>.
- [33] D. J. Wales, *Int. Rev. Phys. Chem.* **2006**, *25*, 237. Available at: <http://dx.doi.org/210.1080/01442350600676921>.
- [34] D. A. Evans, D. J. Wales, *J. Chem. Phys.* **2003**, *119*, 9947. Available at: <http://dx.doi.org/9910.1063/9941.1616515>.
- [35] J. M. Carr, D. J. Wales, *J. Chem. Phys.* **2005**, *123*, 234901. Available at: <http://dx.doi.org/234910.231063/234901.2135783>.
- [36] W. Brütting, S. Berleb, A. G. Mückl, *Org. Electron.* **2001**, *2*, 1.
- [37] J. J. Kwiatkowski, J. Nelson, H. Li, J. L. Bredas, W. Wenzel, C. Lennartz, *Phys. Chem. Chem. Phys.* **2008**, *10*, 1852.
- [38] J. L. Brédas, J. E. Norton, J. Cornil, V. Coropceanu, *Acc. Chem. Res.* **2009**, *42*, 1691.
- [39] A. Fuchs, T. Steinbrecher, M. S. Mommer, Y. Nagata, M. Elstner, C. Lennartz, *Phys. Chem. Chem. Phys.* **2012**, *14*, 4259.
- [40] U. H. E. Hansmann, A. Koliński, *Multiscale Approaches to Protein Modeling*, Springer: New York, **2011**; pp. 209-230 Available at: <http://dx.doi.org/4410.1007/978-4411-4419-6889-4410-4419>.
- [41] Y. Okamoto, *J. Mol. Graphics Modell.* **2004**, *22*, 425. Available at: <http://www.sciencedirect.com/science/article/pii/S1093326303001980>.
- [42] B. A. Berg, *Nature* **1993**, *361*, 708. Available at: <http://dx.doi.org/710.1038/361708a361700>.
- [43] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, D. A. Case, *J. Comput. Chem.* **2004**, *25*, 1157. Available at: <http://www.pubmed.org/display.cgi?uids=15116359>.
- [44] D. J. Earl, M. W. Deem, *Phys. Chem. Chem. Phys.* **2005**, *7*, 3910. Available at: <http://dx.doi.org/3910.1039/B509983H>.
- [45] K. V. Klenin, F. Tristram, T. Strunk, W. Wenzel, *J. Comput. Chem.* **2011**, *32*, 2647. Available at: <http://dx.doi.org/2610.1002/jcc.21844>.
- [46] A. Verma, W. Wenzel, *Biophys. J.* **2009**, *96*, 3483. Available at: <http://www.sciencedirect.com/science/article/pii/S0006349509003877>.
- [47] B. Fischer, K. Fukuzawa, W. Wenzel, *Proteins: Struct. Funct. Bioinformatics* **2008**, *70*, 1264. Available at: <http://dx.doi.org/1210.1002/prot.21607>.
- [48] R. Huey, G. M. Morris, A. J. Olson, D. S. Goodsell, *J. Comput. Chem.* **2007**, *28*, 1145. Available at: <http://dx.doi.org/1110.1002/jcc.20634>.
- [49] E. Malolepsza, B. Strodel, M. Khalili, S. Trygubenko, S. N. Fejer, D. J. Wales, *J. Comput. Chem.* **2010**, *31*, 1402. Available at: <http://dx.doi.org/1410.1002/jcc.21425>.
- [50] J. K. Noel, P. C. Whitford, K. Y. Sanbonmatsu, J. N. Onuchic, *Nucleic Acids Res.* **2010**, *38*(suppl 2), W657. Available at: http://nar.oxfordjournals.org/content/38/suppl_2/W657.abstract.
- [51] T. Strunk, M. Wolf, W. Wenzel, In *Computational Methods in Science and Engineering: Proceedings of the Workshop SimLabs@KIT*, November 29-30, 2010, Karlsruhe, Germany; I. Kondov, G. Poghosyan, O. Kirner, O. Schneider, F. Schmitz, Eds.; KIT Scientific Publishing, Karlsruhe: Karlsruhe, Germany, **2011**; pp. 35-46 Available at: <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000023323>.
- [52] D. J. Wales, J. P. K. Doye, *J. Phys. Chem. A* **1997**, *101*, 5111. Available at: <http://dx.doi.org/5110.1021/jp970984n>.
- [53] A. Verma, A. Schug, K. H. Lee, W. Wenzel, *J. Chem. Phys.* **2006**, *124*, 044515. Available at: <http://dx.doi.org/044510.041063/044511.2138030>.
- [54] M. C. Prentiss, D. J. Wales, P. G. Wolynes, *J. Chem. Phys.* **2008**, *128*, 225106. Available at: <http://dx.doi.org/225110.221063/225101.2929833>.
- [55] A. Schug, W. Wenzel, *Biophys. J.* **2006**, *90*, 4273.
- [56] S. M. Gopal, K. Klenin, W. Wenzel, *Proteins: Struct. Funct. Bioinformatics* **2009**, *77*, 330. Available at: <http://dx.doi.org/310.1002/prot.22438>.
- [57] I. Kondov, A. Verma, W. Wenzel, *Biochemistry* **2009**, *48*, 8195. Available at: <http://dx.doi.org/8110.1021/bi900702m>.
- [58] T. Strunk, K. Hamacher, F. Hoffgaard, H. Engelhardt, M. D. Zillig, K. Faist, W. Wenzel, F. Pfeifer, *Mol. Microbiol.* **2011**, *81*, 56. Available at: <http://>
- [59] D. B. Kokh, R. C. Wade, W. Wenzel, *WIREs: Comput. Mol. Sci.* **2011**, *1*, 298. Available at: <http://dx.doi.org/210.1002/wcms.1029>.
- [60] D. B. Kokh, W. Wenzel, *J. Med. Chem.* **2008**, *51*, 5919. Available at: <http://dx.doi.org/5910.1021/jm800217k>.
- [61] F. A. Lemasson, T. Strunk, P. Gerstel, F. Hennrich, S. Lebedkin, C. Barner-Kowollik, W. Wenzel, M. M. Kappes, M. Mayor, *J. Am. Chem. Soc.* **2010**, *133*, 652. Available at: <http://dx.doi.org/610.1021/ja105722u>.
- [62] J. W. Neidigh, R. M. Fesinmeyer, N. H. Andersen, *Nat. Struct. Mol. Biol.* **2002**, *9*, 425. Available at: <http://dx.doi.org/410.1038/nsb1798>.
- [63] C. Simmerling, B. Strockbine, A. E. Roitberg, *J. Am. Chem. Soc.* **2002**, *124*, 11258. Available at: <http://dx.doi.org/11210.11021/ja0273851>.
- [64] C. D. Snow, B. Zagrovic, V. S. Pande, *J. Am. Chem. Soc.* **2002**, *124*, 14548. Available at: <http://dx.doi.org/14510.11021/ja0286041>.
- [65] S. Chowdhury, M. C. Lee, G. Xiong, Y. Duan, *J. Mol. Biol.* **2003**, *327*, 711. Available at: <http://www.sciencedirect.com/science/article/pii/S0022283603001773>.
- [66] J. W. Pitera, W. Swope, *Proc. Natl. Acad. Sci.* **2003**, *100*, 7587. Available at: <http://www.pnas.org/content/7100/7513/7587.abstract>.
- [67] M. Brieg, J. Setzler, W. Wenzel, **2012**.
- [68] W. C. Still, A. Tempczyk, R. C. Hawley, T. Hendrickson, *J. Am. Chem. Soc.* **1990**, *112*, 6127. Available at: <http://dx.doi.org/6110.1021/ja00172a00038>.
- [69] D. P. Anderson, In *5th IEEE/ACM International Workshop on Grid Computing*, **2004**; Pittsburgh, PA, USA, pp. 4-10.
- [70] T. Strunk, P. Anand, M. Brieg, M. Wolf, K. Klenin, I. Meliciani, F. Tristram, I. Kondov, W. Wenzel, *Proceedings of the 3rd International Workshop on Science Gateways for Life Sciences*, University of Westminster, Westminster, London, UK, **2011**.
- [71] S. Behrens, W. Habicht, W. Wenzel, K. J. Böhm, *J. Nanosci. Nanotechnol.* **2009**, *9*, 6858. Available at: <http://www.ingentaconnect.com/content/asp/jnn/2009/00000009/00000012/art00000010>.
- [72] J. A. Wells, C. L. McClendon, *Nature* **2007**, *450*, 1001.
- [73] J. Janin, *Proteins: Struct. Funct. Bioinformatics* **2002**, *47*, 257. Available at: <http://dx.doi.org/210.1002/prot.10111>.
- [74] I. Meliciani, K. Klenin, T. Strunk, K. Schmitz, W. Wenzel, *J. Chem. Phys.* **2009**, *131*, 034114. Available at: <http://dx.doi.org/034110.031063/034111.3177008>.
- [75] A. S. Borer, P. Wassmann, M. Schmidt, D. R. Hoffman, J. J. Zhou, C. Wright, T. Schirmer, Z. Marković Housley, *J. Mol. Biol.* **2012**, *415*, 635. Available at: <http://www.sciencedirect.com/science/article/pii/S0022283611011338>.
- [76] O. F. Lange, N. A. Lakomek, C. Farès, G. F. Schröder, K. F. A. Walter, S. Becker, J. Meiler, H. Grubmüller, C. Griesinger, B. L. de Groot, *Science* **2008**, *320*, 1471. Available at: <http://www.sciencemag.org/content/1320/5882/1471.abstract>.
- [77] T. Wlodarski, B. Zagrovic, *Proc. Natl. Acad. Sci.* **2009**, *106*, 19346. Available at: <http://www.pnas.org/content/19106/19346/19346.abstract>.
- [78] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, A. J. Olson, *J. Comput. Chem.* **2009**, *30*, 2785.
- [79] B. Fischer, H. Merlitz, W. Wenzel, *Comput. Life Sci., Proc.* **2005**, *3695*, 186.
- [80] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, P. S. Shenkin, *J. Med. Chem.* **2004**, *47*, 1739.
- [81] P. C. Weber, D. H. Ohlendorf, J. J. Wendoloski, F. R. Salemme, *Science* **1989**, *243*, 85. Available at: <http://www.pubmed.org/display.cgi?uids=2911722>.
- [82] D. J. Wales, H. A. Scheraga, *Science* **1999**, *285*, 1368.
- [83] M. A. Wolak, B. Jang, L. C. Palilis, Z. H. Kafafi, *J. Phys. Chem. B* **2004**, *108*, 5492.
- [84] R. B. Campbell, J. M. Robertson, J. Trotter, *Acta Crystallogr.* **1961**, *14*, 705. Available at: <http://dx.doi.org/710.1107/S0365110X61002163>.
- [85] J. J. P. Stewart, MOPAC for Solid State Physics. *Quant. Chem. Prog. Exch.* **1985**, *5*, 62.