BIOINFORMATICS

GROMACS 4.5: A high-throughput and highly parallel open source molecular simulation toolkit

Sander Pronk[1,2], Szilárd Páll[1,2], Roland Schulz[3,4], Per Larsson[5,6], Pär Bjelkmar[1,7], Rossen Apostolov[1,2], Michael R. Shirts[8], Jeremy C. Smith[3,4], Peter M. Kasson[5,6], David van der Spoel[1,9], Berk Hess[1,2], Erik Lindahl[1,2,7]

[1]Science for Life Laboratory, Stockholm & Uppsala, Sweden
[2]Theoretical & Computational Biophysics, KTH Royal Institute of Technology, Stockholm, Sweden
[3]UT/ORNL Center for Molecular Biophysics, Oak Ridge National Laboratory, Oak Ridge, USA
[4]Dept. Biochemistry and Cellular and Molecular Biology, University of Tennessee, Knoxville, USA
[5]Dept. of Molecular Physiology and Biological Physics, University of Virginia, Charlottesville, USA
[6]Dept. of Biomedical Engineering, University of Virginia, Charlottesville, USA
[7]Center for Biomembrane Research, Dept. Biochemistry & Biophysics, Stockholm University, Sweden
[8]Dept. of Chemical Engineering, University of Virginia, Charlottesville, USA
[9]Dept. of Cell & Molecular Biology, Uppsala University, Uppsala, Sweden

Associate Editor: Jonathan D. Wren

ABSTRACT

Motivation: Molecular simulation has historically been a low-throughput technique, but faster computers and increasing amounts of genomic and structural data are changing this by enabling large-scale automated simulation of, for instance, many conformers or mutants of biomolecules with or without a range of ligands. At the same time, advances in performance and scaling now make it possible to model complex biomolecular interaction and function in a manner directly testable by experiment. These applications share a need for fast and efficient software that can be deployed on massive scale in clusters, web servers, distributed computing, or cloud resources.

Results: Here, we present a range of new simulation algorithms and features developed over the last four years, leading up to the GROMACS 4.5 software package. The software now automatically handles wide classes of biomolecules such as proteins, nucleic acids, and lipids and comes with all commonly used force fields for these molecules built-in. GROMACS supports several implicit solvent models as well as new free energy algorithms, and the software now employs multithreading for efficient parallelization even on low-end systems, including windows-based workstations. Together with hand-tuned assembly kernels and state-of-the-art parallelization, this provides extremely high performance and cost efficiency for high-throughput as well as massively parallel simulations.

Availability: GROMACS is open source and free software, and available from `http://www.gromacs.org`.

Contact: erik.lindahl@scilifelab.se

1 INTRODUCTION

Although molecular dynamics simulation of biomolecules is frequently classified as computational chemistry, the scientific roots of the technique trace back to polymer chemistry and structural biology in the 1970s, where it was used to study the physics of local molecular properties–flexibility, distortion and stabilization–and relax early X-ray structures of proteins on short time scales (Lifson and Warshel, 1968; Levitt and Lifson, 1969; Berendsen, 1976; McCammon *et al.*, 1977). Molecular simulation in general was pioneered even earlier in physics and applied to simplified hard-sphere systems (Alder and Wainwright, 1957). The field of molecular simulation has developed tremendously since then, and simulations are now routinely performed on multi-microsecond scale where it is possible to repeatedly fold small proteins (van der Spoel and Seibert, 2006; Lindorff-larsen *et al.*, 2011; Bowman *et al.*, 2011), predict interactions between receptors and ligands (Chong *et al.*, 1999; Huang and Caflisch, 2011), predict functional properties of receptors and even capture intermediate states of complex transitions, e.g. in membrane proteins (Nury *et al.*, 2010). This classical type of single long simulation continues to be very important since it provides ways to directly monitor molecular processes not easily observed through other means. However, many current studies increasingly rely on large sets of simulations, enabled in part by the ever-increasing number of structural models made possible by sequencing and structural genomics as well as new techniques to estimate complex molecular properties using thousands of shorter simulations (Noé and Fischer, 2008; Pande *et al.*, 2010). Mutation studies can now easily build models and run short simulations for hundreds of mutants, model-building web servers frequently offer automated energy minimization and refinement (Zhang *et al.*, 2011), and free energy calculations are increasingly being used to provide better interaction energy estimates than what is possible with docking (Helms and Wade, 1998; Boyce *et al.*, 2009). In these scenarios, classical molecular dynamics simulations based on empirical models have a significant role to play since most properties of interest are defined by free

energies, which typically require extensive sampling that traditional quantum chemistry methods can not provide for large systems.

These developments would not have been possible without significant research efforts in simulation algorithms, optimization, parallelization, and not least ways to integrate simulations in modeling pipelines. The emergence of standardized packages for molecular modeling such as CHARMM (Brooks *et al.*, 2009), GROMOS (Christen *et al.*, 2005), Amber (Case *et al.*, 2005), NAMD (Phillips *et al.*, 2005), and GROMACS (Hess *et al.*, 2008) has been important since these have helped commoditize simulation and molecular modeling research and made the techniques available to life science application researchers who are not specialists in simulation development.

All these packages have complementary strengths and profiles; for the GROMACS molecular simulation toolkit, one of our primary long-term development goals has been to achieve the highest possible simulation efficiency for the small- to medium size clusters that were present in our own research labs. Since computational resources are typically limited in those settings, it is sometimes preferable to use throughput approaches with moderate parallelization that yield whole sets of simulations rather than maximizing performance in a single long simulation. However, in recent years, we have combined this with optimizing parallel scaling to enable long simulations when dedicated clusters or supercomputers are available for select critical problems.

Over the last four years since GROMACS 4, we have developed a number of new features and improvements that have led up to release 4.5 of the software and significantly improved both performance and efficiency for throughput as well as massively parallel applications. Many tasks that only a decade ago required exceptionally large dedicated supercomputing resources are now universally accessible, and sometimes they can even be run efficiently on a single workstation or laptop. However, contemporary "low-end" machines are now parallel computers ranging from 2-4 cores on a laptop and up to 16-32 cores on workstations and require parallel programs to utilize all resources in a single job. Here, we present the work and features that have gone into GROMACS 4.5, including development to make the code fully portable and multithreaded on a wide range of platforms, features to facilitate high-throughput simulation, and not least more efficient tools to help automate complex simulations such as free energy calculations, with another long-term goal of commoditizing affinity prediction as well. High-end performance in GROMACS has also been improved with new decomposition techniques in both direct and reciprocal space that push parallelization further and that have made microsecond simulation timescales reachable in a week or two even for large systems using only quite modest computational resources.

2 RESULTS

2.1 An open source molecular simulation framework

The development of GROMACS was originally largely driven by our own needs for efficient modeling. However, in hindsight the decision to release the package as both open source and free software was a significant advance for the project. The codebase has become a shared infrastructure with contributions from several research labs world-wide, where every single patch and all code review is public as soon as they are committed to the repository. We explicitly encourage extensions and re-use of the code; as examples, GROMACS is used as a module to perform energy minimization in other structural bioinformatics packages (including commercial ones), it is available as a component from many vendors that provide access to cloud computing resources, and some of the optimized mathematical functions (such as inverse square roots) have been reused in other codes. Many Linux distributions also provide provide precompiled or contributed binaries of the package. These features per se do not necessarily say anything about scientific qualities, but we believe this open development platform ensures *(i)* intensive code scrutiny, *(ii)* several state-of-the art implementations of algorithms, and *(iii)* immediate availability of research work to end users. Compared to only ten years ago, the project is now used everywhere from the smallest embedded processors to the largest supercomputers in the world, with applications ranging from genome-scale refinement of coarse-grained models to multi-microsecond simulations of membrane proteins or vesicle fusion.

2.2 Enabling efficient simulation on desktop resources

Supercomputers are still very important for the largest molecular simulations, but many users rely on very modest systems for their computational needs. For many applications, one can even argue this is the most important target: researchers often use interactive tools, companies are hesitant to invest in expensive computational infrastructure, and there is an increasing focus on high-throughput studies, where a single calculation cannot use 50% of a cluster. Historically, this low-end regime has been the primary goal for GROMACS (while e.g. NAMD (Phillips *et al.*, 2005) has focused on parallel scaling), and we have specifically focused efforts on achieving the highest possible efficiency on single nodes. GROMACS is designed for maximum portability, with external dependencies kept to a minimum and fall-back internal libraries provided whenever possible. It is possible to build GROMACS on almost any Unix-based system (including many embedded architectures). In GROMACS 4.5, we have extended this further, making Microsoft Windows a fully-supported platform. This is obviously relevant for many researchers' desktops, but it is also critical for distributed computing projects where the software runs on participant-controlled computers, e.g. in the Folding@Home project. One of the main challenges in the last few years has been the emergence of multi-core machines. While GROMACS runs in parallel, it was designed to use MPI communication libraries present on supercomputers rather than automatically using multiple cores. In release 4.5, we have solved this by designing a new internal "thread_MPI" interface layer that implements the MPI communication calls using multithreading and automatically uses every core available on a laptop or desktop for increased performance.

2.3 High-throughput simulation & modeling

As simulation software and computer performance has improved, biomolecular dynamics has increasingly been used for structure equilibration, sampling of models, or to test what effects mutations might have on structure and dynamics by introducing many different mutations and perform comparatively short simulations on multiple structures. While this type of short simulation might not look as technically impressive as long trajectories, we strongly feel it

is a much more powerful approach for many applications. Since simulations build on statistical mechanics, a result seen merely in one long trajectory might as well be a statistical fluctuation that would never be accepted as significant in an experimental setting. In contrast, by choosing to perform e.g. 50 100-ns simulations instead of a single $5\mu s$ one it is suddenly possible to provide standard error estimates and quantitative instead of qualitative results from simulations (Lange *et al.*, 2010). In addition, the same toolbox can be applied to liquid simulations, where the need for sampling is limited, but where one often needs to study a range of systems under different conditions (e.g. temperature) to extract data that can be compared to experiments. (Caleman *et al.*, 2012). As discussed above, GROMACS has always been optimized to achieve the best possible efficiency using scarce resources (which we believe is the norm for most users), and version 4.5 has introduced several additional features to aid high-throughput simulation. All GROMACS runs are now automatically checkpointed and can be interrupted and continued as frequently as required, and optional flags have been added to enable binary reproducibility of trajectories. Since GROMACS is successfully used in a number of distributed computing projects where both CPU and storage hardware might be less controlled, GROMACS's main simulation executable mdrun now flushes all pending buffers after each file-writing step and tries to flush file system cache when writing checkpoints. Since users often work with many data sets at once, we have implemented MD5 hashes on checkpoint continuation files to guarantee their integrity and to make sure the user does not append to the wrong file by mistake. These additional checks have allowed us to enable file appending on job continuation: repeated short jobs that continue from checkpoints will yield a single set of output just as from a single long job. Hundreds or even thousands of smaller simulations can be started with a single GROMACS execution command to optimize use on supercomputers that favor large jobs, and each of these can be parallel themselves if advantageous. GROMACS also supports simulations running in several modern cloud computing environments where virtual server instances can be started on demand. Since cloud computing usage is also billed by the hour, we believe the most instructive metric for performance and efficiency is to actually measure simulation performance in terms of the cost to complete a given simulation — for an example, see the performance section below.

2.4 Implicit solvent & knowledge-based modeling

In addition to the high-throughput execution model, there are a number of new code features developed to support modeling and rapid screening of structures. In previous versions GROMACS has not supported implicit solvent since it seemed of little utility when it was slower than explicit water. This has changed with version 4.5, and the code now comes with very efficient implementations of the Still (Qui *et al.*, 1997), HCT (Hawkins-Cramer-Truhlar) (Hawkins *et al.*, 1996) and OBC (Onufriev-Bashford-Case) (Onufriev *et al.*, 2004) models for Generalized Born interactions based on tabulated interaction rescaling (Larsson and Lindahl, 2010). Together with manually tuned assembly kernels, implicit solvent simulations can reach performance in excess of a microsecond per day for small proteins even on standard CPUs. The neighbor-searching module has been updated to support grid-based algorithms even in vacuo - including support for atoms diffusing away towards

infinity with maintained performance - and there are now also highly optimized kernels to compute all-vs-all interactions without cutoffs both for standard and Generalized Born interactions. The program now also supports arbitrary knowledge-based statistical interactions through atom-group specific tables both for bonded and nonbonded interactions. Constraints such as those used in refinement can be applied either to positions, atomic distances, or torsions, and there are several options for ensemble weighting of contributions from multiple constraints.

2.5 Strong scaling on massively parallel clusters

Despite the rapid emergence of high-throughput computing, the usage of massively parallel resources continues to be a cornerstone of high-end molecular simulation. Absolute performance is the goal for this usage too, but here it is typically limited by the scalability of the software. GROMACS 4.0 introduced a number of new features, including new neutral territory domain decomposition algorithm that is also used in the Desmond software (Bowers *et al.*, 2006), but the performance was still limited by the scaling of the particle-mesh Ewald (Essmann *et al.*, 1995) (PME) implementation, in particular the single-dimensional decomposition of the fast Fourier transform (FFT) grids. For GROMACS 4.5, this has been solved with a new implementation of a two-dimensional, or "pencil" decomposition of reciprocal space. A subset of nodes are dedicated to the PME calculation, and at the beginning of each step the direct-space nodes send coordinate and charge data to them. Since direct space can be composed in all three dimensions, a number of direct-space nodes (typically 3-4) map onto a single reciprocal-space node (Fig. 1). Limiting the computation of the 3D-FFT to a smaller number of nodes improves parallel scaling significantly (Hess *et al.*, 2008), and is now also used e.g. by NWChem (Valiev *et al.*, 2010). The new pencil decomposition makes it much easier to automatically determine both real- and reciprocal-space decompositions of arbitrary systems to fit a given number of nodes. The automatic load balancing step of the domain decomposition has also been improved, domain decomposition now works without periodic boundary conditions (important for implicit solvent) and GROMACS now includes tools to automatically tune the balance between direct and reciprocal-space work. In particular when running in parallel over large numbers of nodes it is advantageous to move more work to real space (which scales near-linearly) and decrease the reciprocal-space load to reduce the dimensions of the 3D-FFT grid (where the number of communication messages scales with the square of the number of nodes involved). The latest version of GROMACS also supports many types of multi-level parallelism; in addition to coding-level optimizations such as single-instruction multiple-data instructions and the multithreaded execution, GROMACS supports replica-exchange ensemble simulations where a single simulation can use hundreds of replicas that only communicate every couple of seconds, which makes it possible to scale even fairly small systems (e.g. a protein) to thousands of nodes for these kinds of calculations. Finally, for the very largest systems comprising hundreds of millions of particles we now achieve true linear weak scaling for reaction-field and other non-lattice-summation methods (Schulz *et al.*, 2009). A lot of recent work has been invested in reducing memory needs and enabling parallel IO, and the code has been shown to successfully scale to over 150,000 cores.

2.6 Automated topology generation for wide classes of molecules & force fields

It was clear that the automated tools to generate input files were somewhat limited in earlier releases of GROMACS; few molecules apart from single-chain proteins worked perfectly. For version 4.5, the `pdb2gmx` tool has been reworked and we now support automatic topology generation for proteins, DNA, RNA, and many small molecules. Any number of chains and different molecule classes can be mixed, and they are automatically detected. The program provides several different options for how to handle termini and `HETATM` records in structures, and residue names and numbering from the input files are now maintained throughout the main simulation and analysis tools. With the most recent version, the package now comes with standard support for virtually all major point-charge force fields: GROMOS43a1, GROMOS43a2, GROMOS45a3, GROMOS53a5, GROMOS53a6, Encad, OPLS, OPLS-AA/L, CHARMM19, CHARMM27, Amber94, Amber96, Amber99, Amber99SB, AmberGS, Amber03, and Amber99SB-ILDN. To the best of our knowledge, this range of forcefield support is currently unique among packages and makes it straightforward to systematically compare the influence of the parameter approximations in biomolecular modeling. The code also provides name translation files to support all the conventions used in the different force fields. In addition, a number of databases provide topologies for small molecules for use in GROMACS (Malde *et al.*, 2011; van der Spoel *et al.*, 2012) whereas small molecule topologies for the generalized Amber force field (Wang *et al.*, 2005), as well as topologies using the Charm general force field (Vanommeslaeghe *et al.*, 2010) can be readily converted to GROMACS format, further increasing the applicability of the software.

2.7 Accurate and flexible integration

GROMACS 4.5 also includes a number of additional integrators of the equations of motion. Originally the code only supported leapfrog verlet, which keeps track of the positions at the full step while the velocities are offset by half a time step. The velocity Verlet algorithm (Swope *et al.*, 1982) is now also fully supported. In velocity Verlet, positions and velocities are both at the same time point. For constant energy simulations, both algorithms give the same trajectories, but for constant temperature or constant pressure simulations, velocity Verlet integration provides many additional features. A number of pressure control or temperature control algorithms are only possible with velocity Verlet integrators, because these algorithms require both the pressure and temperature to be specified at the same time. Additionally, velocity Verlet is more common in other simulation packages (Amber, NAMD, Desmond), so it makes it easier to perform detailed comparisons between GROMACS and other molecular simulation engines. Good temperature algorithms already exist for leapfrog algorithm, however, so velocity Verlet is not generally necessary in such cases.

For pressure control, the existing algorithm by Parrinello and Rahman (Parrinello and Rahman, 1981; Nosé and Klein, 1983) is not quite correct using leapfrog, although it has been verified to give a correct distribution of volumes within statistical noise in many situations. Slight errors arise because of the time step mismatch between the components of the pressure calculation involving kinetic energy and potential energy. The introduction of velocity Verlet allows the use of additional, more rigorous pressure control algorithms, such as that of Martyna, Tuckerman, Tobias, and Klein (Martyna *et al.*, 1996), which can be useful e.g. for accurate lipid bilayer simulations in GROMACS 4.5.

The leapfrog Verlet and velocity Verlet are both implemented as specific instances of a method called Trotter factorization, a general technique for decomposition of the equations of motion which makes it possible to write out different symplectic integrators based on different ordering of the integration of different degrees of freedom. This Trotter factorization approach will make it possible to eventually support a large range of multistep and other accelerated sampling integrators in the future, and it is already used for more efficient temperature and pressure scaling. Historically, Gromacs has relied on virtual interaction sites and all-bond constraints to extend the shortest time step in integration (in contrast to e.g. NAMD that uses multiple time step integration), but this approach will make it possible to support both alternatives in future versions.

2.8 A state-of-the-art free energy calculation toolbox

Simulation-based free energy calculations provide a way to accurately include effects both of interactions and entropy, and accurately predict solvation and binding properties of molecules. It is one of the most direct ways that simulations can provide specific predictions of properties that can also be measured experimentally. GROMACS as well as other packages have long supported free energy perturbation and slow-growth methods to calculate free energy differences when gradually changing the properties of molecules. The present release of the code provides an extensive new free energy framework based on Bennett Acceptance Ratios (BAR). As a free energy perturbation technique, this allows the calculation of a free energy difference along an arbitrary coupling parameter λ, typically in multiple steps to provide sufficient phase space overlap between each step. The total energy (or actually, Hamiltonian) is then defined as $H(\lambda) = (1 - \lambda)H_0 + \lambda H_1$, where H_0 & H_1 are the Hamiltonians for the two end states. BAR uses differences in Hamiltonian as the basis for calculating the free energy difference, and has been shown to be both the most efficient free energy perturbation method for extracting free energy differences(Bennett, 1976), and a statistically unbiased estimator for this free energy difference(Shirts and Pande, 2005).

The Hamiltonian differences needed for BAR are now calculated automatically on the fly in simulations, rather than as a post-processing step using large full-precision trajectories, which makes it possible to use distributed computing or cloud resources where the available storage and bandwidth is limited. Rather than manually defining how to modify each molecule, the user can now simply specify that they want to calculate the free energy of decoupling a particular molecule or group of atoms from the system as a simulation parameter. Given the set of output files from such a project, the code also comes with a new tool `g_bar` that automatically calculates the free energy and its uncertainty at each step, and provides a finished estimate of the free energy required to change from one end state to the other, including estimates of the standard error and measures for the phase space overlap (Fig. 2).

2.9 Other features

In addition to the larger development concepts covered here, several additional parts of GROMACS have been improved and extended for version 4.5.

- We now support symplectic leapfrog and velocity Verlet integrators for fully reversible temperature and pressure coupling, with several new barostats and thermostats, including Nose-Hoover chains for ergodic temperature control and MTTK pressure control integrators. These are important for calculating accurate free energies, in particular for smaller systems or cases where pressure will affect the result.

- A new file-format plugin has been designed to allow GROMACS to directly read any trajectory or coordinate format supported by the optional VMD libraries.

- The previously labor-intensive task of embedding and equilibrating membrane proteins in lipid bilayers has been automated with the new tool `g_membed` (Wold *et al.*, 2010). Given a membrane protein structure and an arbitrary bilayer (including lipid mixtures or proteins), this tool virtually shrinks the membrane protein to a small axis and then gently "grows" it in over a few thousand steps. Lipids are removed based on overlap, and the tool has full support for asymmetrically-shaped proteins.

- Non-equilibrium simulation: it is now possible to pull any number of groups in arbitrary directions and to apply torques in addition to forces. The corresponding `g_wham` tool for analysis of non-equilibrium simulations has been updated extensively to allow robust error estimates using Bayesian bootstrapping (Hub *et al.*, 2010).

- The multi-scale modeling now includes a QM/MM interface to a number of common quantum chemistry programs and algorithms, coarse-grained (CG) modeling with force fields such as MARTINI (Marrink *et al.*, 2007), and a highly efficient parallel implicit solvent algorithm that can all be used in combination.

- Normal-mode analysis can now be performed for extremely large systems through a new sparse-matrix diagonalization engine that also works in parallel, and even for PME simulations it is possible to perform the traditional non-sparse (computationally costly) diagonalization in parallel.

3 PERFORMANCE

3.1 Scaling

For systems where absolute speed matters, the final simulation performance can be expressed as `speed_per_core * ncores * scaling_efficiency`. Rather than optimizing only for scaling efficiency, we have aimed to improve both absolute performance per core and the scaling efficiency at the same time. Recent enhancements in this respect include the better Particle Mesh Ewald (PME) parallel decomposition described above. The choice of method for calculating long-range electrostatics can greatly affect simulation performance, and rather than simply optimizing the method that scales best (reaction-field), we have worked to optimize the method that is currently viewed as best practice in the field (van der Spoel and van Maaren, 2006). By implementing a two-dimensional pencil node decomposition for PME and improving the dynamic load-balancing algorithms, we obtain close-to-linear scaling over large numbers of nodes for a set of benchmark systems that were selected as real world applications from our

and others recent work. Scaling results are plotted in Fig. 3 for a ligand-gated ion channel (Murail *et al.*, 2011) (a typical membrane protein in a bilayer), a massive vesicle fusion simulation (Kasson *et al.*, 2010), a virus capsid (Larsson *et al.*, 2012), and a large methanol-water mixture. To estimate real-world performance, we report scaling and performance results on two clusters: a Cray XE6 with a Gemini interconnect and a more commodity cluster with QDR Infiniband and significantly less than full bisectional bandwidth. Both machines were equipped with AMD Magny-cours processors clocked at 1.9-2.1GHz. For all simulations except the smaller ion channel (130,000 atoms), we obtain strong linear scaling well over 1000 cores; for the ion channel, the linear scaling regime still extends below 500 atoms/core. All of these benchmark simulations use PME long-range electrostatics where the lattice component is evaluated every single step; our tests with reaction field electrostatics show virtually perfect linear scaling for any number of cores as long as the system is large enough (> 250 atoms/core).

3.2 Single-node parallelization

GROMACS 4.5 implements parallelization at a low level through single instruction multiple data (SIMD) operations, and at a high level through the message-passing interface (MPI), and ongoing efforts add an intermediate level of OpenMP parallelization. This provides good scaling at high core counts but adds complexity to code deployment for small installations. We have therefore written a threads-only implementation of MPI calls that allows single-node parallelization of GROMACS using either POSIX or Windows threads without additional dependencies, using hardware-supported atomic and lock-free synchronization, with non-blocking communication when the MPI specification allows it. Scaling of the thread_MPI implementation is plotted in Fig. 4 for two different systems. The Villin headpiece is a small protein in water (7,300 atoms) simulated here using short 8 Å cut-offs, while the POPC bilayer is a membrane/water system with 17,400 atoms using Particle-Mesh-Ewald electrostatics and 10 Å cut-offs. The scaling behavior is near-identical to OpenMPI (an open source MPI library) on a single node, which is gratifying since OpenMPI is pretty much a state-of-the-art implementation. The advantage of the GROMACS thread_MPI implementation is that it is lightweight, reduces build complexity, and works on a wide variety of systems including Linux, OS/X, Windows, and most embedded systems. This development greatly facilitated large-scale deployment of parallel GROMACS simulations on architectures such as Folding@Home where a user cannot install an MPI library without administrator privileges.

3.3 Throughput simulations

As modern computers have increased in processing power, simulations that used to require supercomputers become tractable on small clusters and even single machines. This has two important consequences: moderate-size simulations become accessible to non-specialists without major allocations of supercomputing resources, and it becomes possible to run many simulations at once to perform moderate-throughput computation on different conditions, mutants of a protein, or small-molecule ligands. To illustrate both of these, and to show that it is not always necessary to invest in clusters to perform efficient simulations, we have benchmarked GROMACS

running on instances at one of the current major cloud providers. The cloud-computing market gives access to relatively capable machines and good burst capacity to thousands of cores or more. With the thread_MPI parallelization in GROMACS, simulation performance is quite good on single nodes, and the installation is trivial. To emphasize the general accessibility of performing these simulations, we have selected a number of small-to-medium size proteins and other common biological systems such as membranes, hydrocarbons and water, and illustrate throughput as the cost for completing a microsecond of simulation at a major cloud provider in October 2012 (Fig. 5). The competitive marked means pricing is relatively similar from all providers. As an example, the 1998 one-microsecond Villin simulation by Duan and Kollman was a landmark computational achievement at the time (Duan and Kollman, 1998). That simulation required months of supercomputer time using hundreds of nodes on one of the fastest machines in the US. Today, anybody can repeat the same simulation in under a week on a single cloud node at a cost of $11, bringing this well within range of a student project (and soon a class lab exercise). Equivalently, screening of hundreds of mutants becomes feasible even without large dedicated resources. The systems and settings have been selected to illustrate a range of different settings, which are described in more detail in the Supplementary Data.

4 CONCLUSIONS & OUTLOOK

Improvements in processor power, simulation algorithms, and new computing paradigms are opening a new frontier for molecular dynamics where many simulations of a moderate-sized system are now tractable. The amount of applicable methods and levels of accuracy available in molecular simulation packages is likewise expected to increase, with both more accurate polarizable models and less-detailed coarse-grained force fields gaining popularity (van der Spoel and Hess, 2011). This enables a fundamental change in the way we approach molecular simulation as a tool. The traditional use of molecular dynamics can be thought of as probing the physical consequences of a given starting protein sequence, ligand, and structure. Now, given an ensemble of 50 candidate models (as might be generated from NMR of a flexible or underdetermined complex), we can evaluate the relative probability of the models and define the accessible conformation space. We can also do mutant scans–the computational equivalent of combinatorial mutagenesis. Soon, approaches such as random walks in sequence space or ligand scaffold space will become routinely tractable. These new capabilities will demand a different approach to simulation–in addition to the underlying physics- and chemistry-based methodology, scientists will need to devote more attention to statistical sampling and leverage benefits of classical informatics techniques such as randomized search algorithms and network flow theory.

Over the next couple of years, we expect the high-throughput trend to become increasingly accentuated: Despite massively increased computational power, researchers have been reluctant to merely push longer simulations. Instead of extending membrane proteins simulations to a single 5-10 microsecond trajectory, most current publications rather use the same amount of total computing time for a whole set of shorter simulations to provide statistics. Fundamentally, we believe this is a scientifically sound

development, and one that is likely to move biomolecular simulation and modeling from compute-centric to data-centric approaches more similar to other methods used in bioinformatics.

REFERENCES

Alder, B. and Wainwright, T. (1957). Phase transition for a hard sphere system. *J. Chem. Phys.*, **27**(5), 1208–1209.

Bennett, C. H. (1976). Efficient estimation of free energy differences from monte carlo data. *J. Comp. Phys.*, **22**, 245.

Berendsen, H. (1976). Report of cecam workshop: models for protein dynamics. orsay may 24-july 17. Technical report, CECAM, Lyon.

Bowers, K. J., *et al.* (2006). Scalable algorithms for molecular dynamics simulations on commodity clusters. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC06)*, New York, NY, USA. ACM.

Bowman, G., *et al.* (2011). Atomistic folding simulations of the five helix bundle protein 6-85. *J. Am. Chem. Soc.*, **133**, 664–667.

Boyce, S., *et al.* (2009). Predicting ligand binding affinity with alchemical free energy methods in a polar model binding site. *J. Mol. Biol.*, **394**, 747–763.

Brooks, B., *et al.* (2009). Charmm: The biomolecular simulation program. *J. Comp. Chem.*, **30**(1545–1615).

Caleman, C., *et al.* (2012). Force field benchmark of organic liquids: Density, enthalpy of vaporization, heat capacities, surface tension, compressibility, expansion coefficient and dielectric constant. *J. Chem. Theor. Comput.*, **8**, 61–74.

Case, D., *et al.* (2005). The amer biomolecular simulation programs. *J. Comp. Chem.*, **26**, 1668–1688.

Chong, L., *et al.* (1999). Molecular dynamics and free-energy calculations applied to affinity maturation in antibody 48g7. *Proc. Natl. Acad. Sci. U.S.A.*, **96**(25), 14330–14335.

Christen, M., *et al.* (2005). The gromos software for biomolecular simulation: Gromos05. *J. Comp. Chem.*, **26**(16), 1719–1951.

Duan, Y. and Kollman, P. (1998). Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, **282**(5389), 740–744.

Essmann, U., *et al.* (1995). A smooth particle mesh ewald method. *J. Chem. Phys.*, **103**, 8577–8592.

Hawkins, D., *et al.* (1996). Parametrized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium. *J Phys Chem A*, **100**, 19824–19839.

Helms, V. and Wade, R. (1998). Computational alchemy to calculate absolute protein-ligand binding free energy. *J. Am. Chem. Soc.*, **120**, 2710–2713.

Hess, B., *et al.* (2008). Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory. Comput.*, **4**(2), 435–447.

Huang, D. and Caflisch, A. (2011). The free energy landscape of small molecule unbinding. *PLoS Comput. Biol.*, **7**(2), e1002002.

Hub, J. S., *et al.* (2010). g_wham-A Free Weighted Histogram Analysis Implementation Including Robust Error and Autocorrelation Estimates. *J. Chem. Theory Comput.*, **6**, 3713–3720.

Kasson, P. M., *et al.* (2010). Atomic-resolution simulations predict a transition state for vesicle fusion defined by contact of a few lipid tails. *PLoS Comput. Biol.*, **6**, e1000829.

Lange, O. F., *et al.* (2010). Scrutinizing Molecular Mechanics Force Fields on the Submicrosecond Timescale with NMR Data. *Biophys. J.*, **99**, 647–655.

Larsson, D., *et al.* (2012). Virus capsid dissolution studied by microsecond molecular dynamics simulations. *PLoS Comput. Biol.*, **8**(5), e1002502.

Larsson, P. and Lindahl, E. (2010). A high-performance parallel-generalized born implementation enabled by tabulated interaction rescaling. *J. Comp. Chem.*, **31**(14), 2593–2600.

Levitt, M. and Lifson, S. (1969). Refinement of protein conformations using a macromolecular energy minimization procedure. *J. Mol. Biol.*, **46**(269–279).

Lifson, S. and Warshel, A. (1968). A consistent force field for calculation on conformations, vibrational spectra and enthalpies of cycloalkanes and n-alkane molecules. *J. Phys. Chem*, **49**(11), 5116–5130.

Lindorff-larsen, K., *et al.* (2011). How fast-folding proteins fold. *Science*, **334**(6055), 517–520.

Malde, A. K., *et al.* (2011). An automated force field topology builder (ATB) and repository: Version 1.0. *J. Chem. Theor. Comput.*, **7**, 4026–4037.

Marrink, S. J., *et al.* (2007). The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B*, **111**, 7812–7824.

Martyna, G. J., *et al.* (1996). Explicit reversible integrators for extended systems dynamics. *Mol. Phys.*, **87**, 1117–1157.

McCammon, J., *et al.* (1977). Dynamics of folded proteins. *Nature*, **267**, 585–590.

Murail, S., *et al.* (2011). Microsecond simulations indicate that ethanol binds between subunits and could stabilize an open-state model of a glycine receptor. *Biophys J.*, **100**, 1642–1650.

Noé, F. and Fischer, S. (2008). Transition networks for modeling the kinetics of conformational change in macromolecules. *Curr. Opin. Struct. Biol.*, **18**, 154–162.

Nosé, S. and Klein, M. L. (1983). Constant pressure molecular dynamics for molecular systems. *Mol. Phys.*, **50**, 1055–1076.

Nury, H., *et al.* (2010). One-microsecond molecular dynamics simulation of channel gating in a nicotinic receptor homologue. *Proc. Natl. Acad. Sci. U.S.A.*, **107**, 6275–6280.

Onufriev, A., *et al.* (2004). Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins*, **55**(2), 383–394.

Pande, V., *et al.* (2010). Everything you wanted to know about markov state models but were afraid to ask. *Methods*, **52**(1), 99–105.

Parrinello, M. and Rahman, A. (1981). Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.*, **52**, 7182–7190.

Phillips, J., *et al.* (2005). Scalable molecular dynamics with namd. *J. Comp. Chem.*, **26**, 1781–1802.

Qui, D., *et al.* (1997). The GB/SA continuum model for solvation. A fast analytical method for the calculation of approximate born radii. *J Phys Chem A*, **101**, 3005–3014.

Schulz, R., *et al.* (2009). Scaling of Multimillion-Atom Biological Molecular Dynamics Simulation on a Petascale Supercomputer. *J. Chem. Theory Comput.*, **5**, 2798–2808.

Shirts, M. R. and Pande, V. S. (2005). Comparison of efficiency and bias of free energies computed by exponential averaging, the bennett acceptance ratio, and thermodynamic integration. *J. Chem. Phys.*, **122**, 144107.

Swope, W. C., *et al.* (1982). A computer-simulation method for the calculation of equilibrium-constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, **76**, 637–649.

Valiev, M., *et al.* (2010). NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications*, **181**, 1477–1489.

van der Spoel, D. and Hess, B. (2011). Gromacs - the road ahead. *WIREs Comput. Mol. Sci.*, **1**, 710–715.

van der Spoel, D. and Seibert, M. (2006). Protein folding kinetics and thermodynamics from atomistic simulations. *Phys. Rev. Lett*, **96**, 238102–238106.

van der Spoel, D. and van Maaren, P. J. (2006). The origin of layer structure artifacts in simulations of liquid water. *J. Chem. Theor. Comput.*, **2**, 1–11.

van der Spoel, D., *et al.* (2012). GROMACS molecule & liquid database. *Bioinformatics*, **28**, 752–753.

Vanommeslaeghe, K., *et al.* (2010). Charmm general force field: A force field for drug-like molecules compatible with the charmm all-atom additive biological force fields. *J. Comput. Chem.*, **31**, 671–690.

Wang, J., *et al.* (2005). Antechamber, an accessory software package for molecular mechanical calculations. *J. Comput. Chem.*, **25**, 1157–1174.

Wold, M., *et al.* (2010). g_membed: Efficient insertion of a membrane protein into an equilibrated lipid bilayer with minimal perturbation. *J. Comput. Chem.*, **31**(11), 2169–2174.

Zhang, J., *et al.* (2011). Atomic-level protein structure refinement using fragment-guided molecular dynamics conformation sampling. *Structure*, **19**(12), 1784–1795.
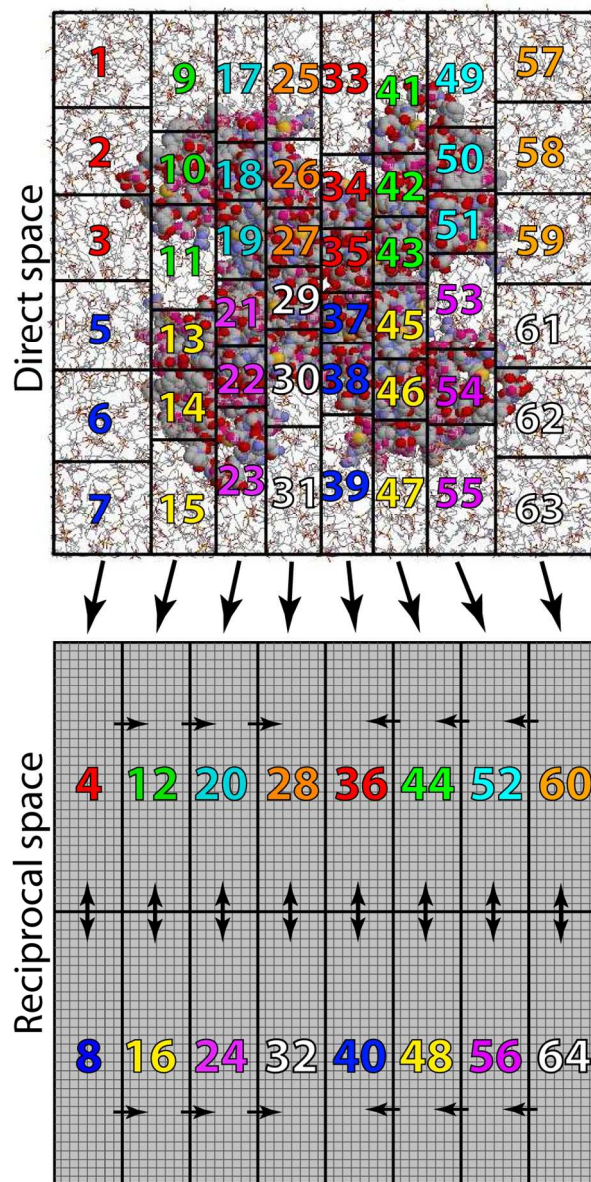
Figure 1

3D Domain decomposition in real space combined with 2D pencil domain decomposition in reciprocal space. The scaling in previous versions of GROMACS was limited by the reciprocal space PME setup, and in particular the 1D decomposition of FFTs along the x-axis. The pencil grid decomposition improves reciprocal space scaling considerably, and makes it easier to use arbitrary numbers of nodes. Colors in the plot refer to a hypothetical system with 4 cores per node, where three are used for direct-space and 1 for reciprocal-space calculations.

82x160mm (300 x 300 DPI)
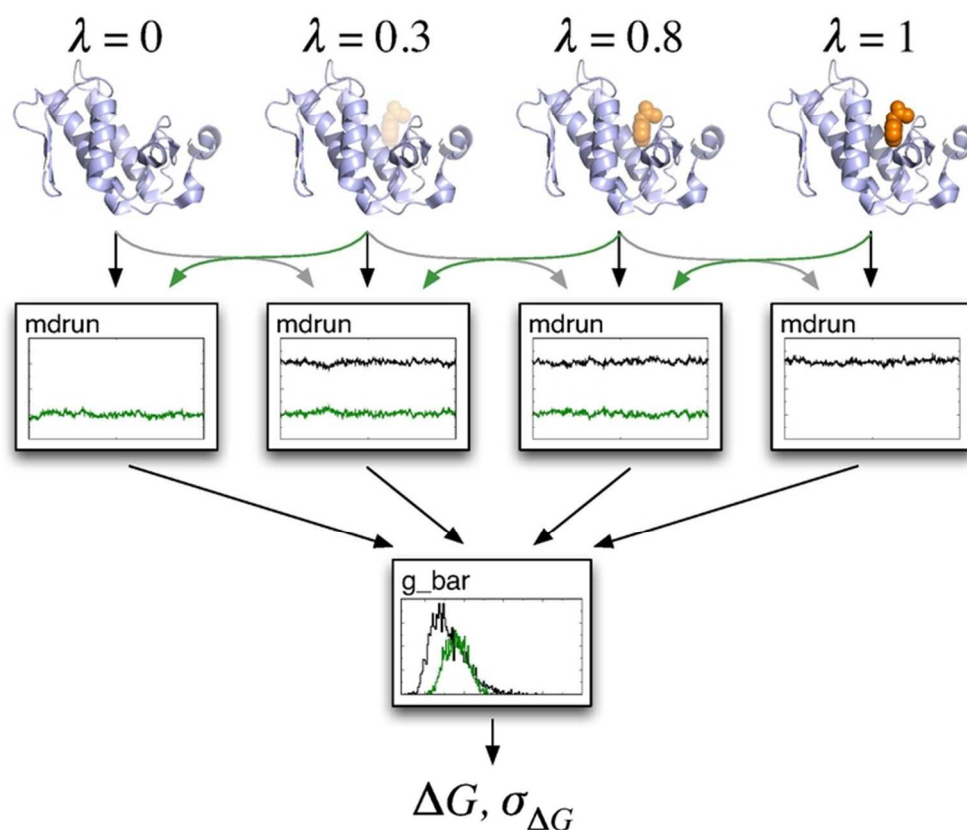
Free energy calculations using Bennett Acceptance Ratios. GROMACS 4.5 provides significantly enhanced tools
to automatically create topologies describing decoupling of molecules from the system to calculate binding or hydration free
energies. The Hamiltonian of the system is
defined as $H(\lambda)=(1-\lambda)H_0 + \lambda H_1$, where
H_0 & H_1 are
the Hamiltonians for the two end states. The user specifies a sequence of lambda points and runs simulations where the phase space overlaps and
Hamiltonian differences are calculated on the fly. Finally, all these files are provided to the new g_bar tool that
automatically analyses the results and provides free energies as well as standard error estimates for the system change.
68x56mm (300 x 300 DPI)

a. Infiniband cluster
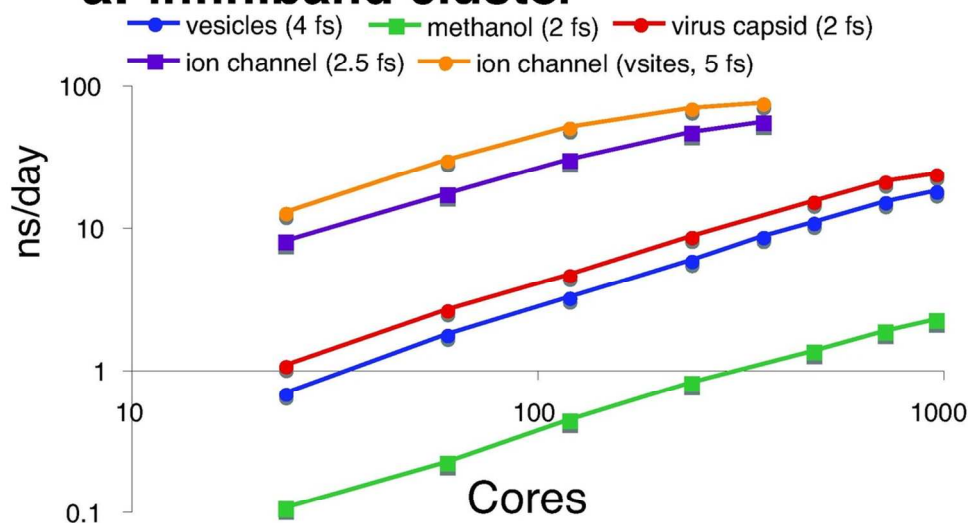
b. Cray XE6
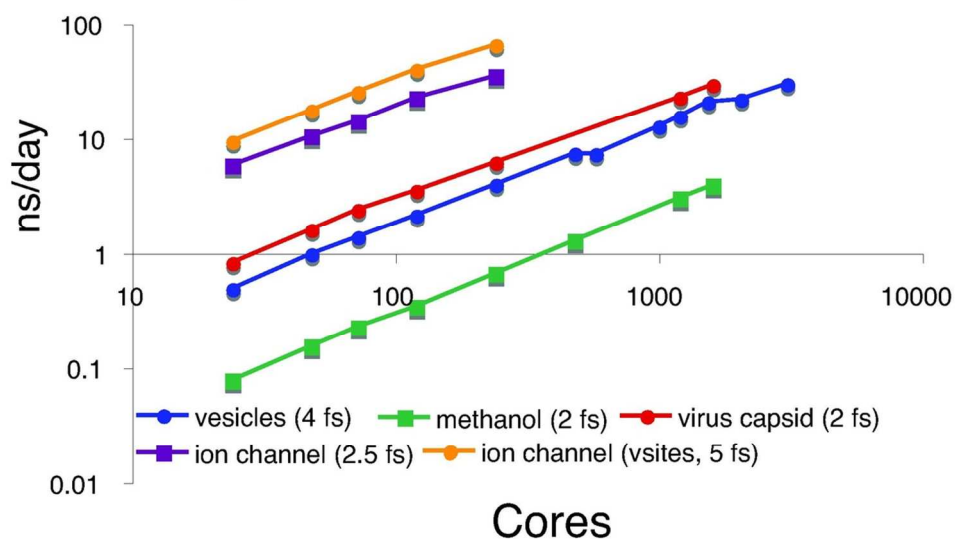
Strong scaling of medium-to-large systems. Simulation performance is plotted as a function of number of cores for a series of simulation systems.
Performance data were obtained on two clusters: one that is thinly connected using QDR Infiniband but not full bisectional bandwidth and a more expensive Cray XE6 with
a Gemini interconnect. In increasing order of molecular size: the ion channel with virtual sites had 129,692 atoms, the ion channel without virtual sites had 141,677 atoms,
this virus capsid had 1,091,164 atoms, the vesicle fusion system had 2,511,403 atoms, and the methanol system had 7,680,000 atoms. See Supplementary Data for details.

103x128mm (300 x 300 DPI)
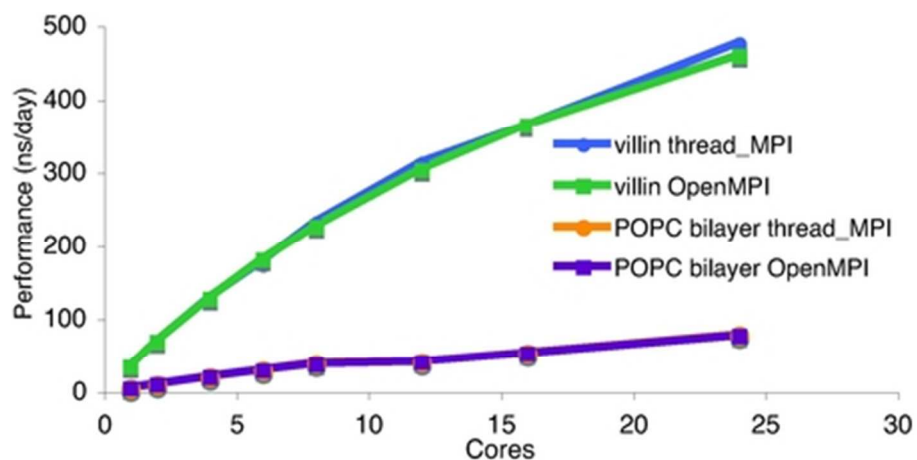
Efficient and portable parallel execution using POSIX or Windows threads. Performance is plotted as a function of number of cores using the thread_MPI library and compared to using OpenMPI. Simulations were run on a single node with 24 AMD 8425HE cores running at 2.66 GHz. Performance is nearly identical between the two parallel implementations. Data are plotted for the Villin and POPC bilayer benchmark systems.

39x19mm (300 x 300 DPI)
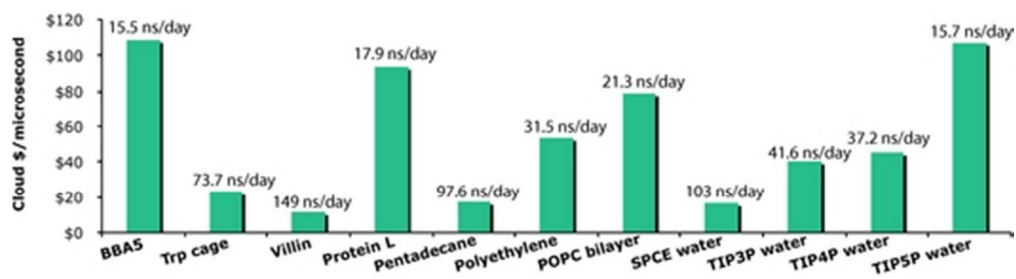
Cost efficiency of Gromacs on small systems. Cost per microsecond is plotted for a series of small systems running on a single 8-core node at a major cloud provider. Simulation
details are available in the Supplementary Data. The label on each bar indicates the performance (inversely proportional to cost). The ready availability of cloud compute instances enables extremely cost-efficient high-throughput simulation using individual nodes.

44x12mm (300 x 300 DPI)