H2020 EINFRA-5-2015

Center of Excellence for Computational Biomolecular Research

www.bioexcel.eu

Project Number 675728

D1.4 – Long-term hardware software assessment for pilot applications and general community

WP1: Software Scalability & Usability

The opinions of the authors expressed in this document do not necessarily reflect the official opinion of the BioExcel partners nor of the European Commission.

Document Information

Deliverable Number	D1.4
Deliverable Name	Long-term hardware software assessment for pilot applications and general community
Due Date	2018-12-31 (PM38)
Deliverable Lead	KTH
Authors	Mark Abraham (KTH), Paul Bauer (KTH), Christian Blau (KTH), Erik Lindahl (KTH), Gilbert Netzer (KTH), Erwin Laure (KTH), Mikael Trellet (UU), Adrien Melquiond (UU), Emiliano Ippoiliti (JUELICH), Vytautas Gapsys (MPG)
Keywords	Hardware software assessment
WP	WP1
Nature	Report
Dissemination Level	Public
Final Version Date	2018-12-22
Reviewed by	PMB
MGT Board Approval	2018-12-22

Document History

Partner	Date	Comments	Version
KTH	2018-11-7	First draft	0.1
KTH	2018-11-30	Updates to early section	0.4
KTH	2018-12-3	Integrated contributions from UU and JUELICH, updates to GROMACS sections	0.5
KTH, JUELICH	2018-12-10	Updates to GROMACS and CPMD sections	0.6
UU	2018-12-10	Updates to HADDOCK sections	0.7
KTH	2018-12-17	Updates to pmx and esd sections	0.8
KTH	2018-12-22	Minor updates from PMB review	0.9

Executive Summary

Future software and hardware development will have a significant impact on all areas of scientific computing including the Life Sciences. Upcoming extreme-scale compute platforms will offer great opportunities for tackling important, large-scale scientific questions. In this document we update our previous analysis of the pre-exascale landscape from the perspective of biomolecular simulation software, including the pilot codes of BioExcel. These are largely unchanged from our previous deliverable 1.3 in this area, and so this report takes the form of an update to that report. Our findings are generally unchanged, and already well publicized among the European HPC stakeholders via several working groups which are involved in the development of EuroHPC, the newly updated PRACE Scientific Case, the ETP4HPC Strategic Research Agenda, and the EXDCI (http://exdci.eu) project in which BioExcel is leading the Life Science working group.

Bio-molecular simulation scientists in industry and academe require effective and usable simulation software that runs well on the hardware resources they can access now. This software must be portable to emerging platforms, because we cannot afford to replace it to run well at the exascale. When we achieve this, we will be able to support the design of new drugs on scales impossible today, obtain better understanding of biochemical pathways, and open new doors for further innovation. This deliverable gives an overview of what we currently see as potential directions and then implementation plans for each of the pilot codes that will suit those directions.

Contents

1 INTRODUCTION 7

2 HARDWARE/SOFTWARE ASSESSMENT FOR THE BIOMOLECULAR SIMULATION COMMUNITY 7

2.1 PROCESSORS 9
2.2 MEMORY AND I/O 10
2.3 NETWORK 12
2.4 CONVERGENCE OF HPC AND HPDA/AI ARCHITECTURES 13

3 CO-DESIGN AND EXTREME SCALE DEMONSTRATORS 15

4 LONG-TERM DEVELOPMENT PLANS FOR BIOEXCEL PILOT APPLICATIONS 17

4.1 LONG-TERM DEVELOPMENT PLANS FOR GROMACS 18
4.1.1 MODULARIZATION OF CORE MDRUN SIMULATION FUNCTIONALITY 19
4.1.2 EVOLUTION OF EXASCALE-SUITABLE TASK PARALLELISM 19
4.1.3 FEATURE ENHANCEMENTS TO SUPPORT DOCKING WORKFLOWS 20
4.1.4 UPDATED AND IMPROVED SAMPLING ALGORITHMS FOR FREE ENERGY CALCULATIONS 20
4.1.5 IMPLEMENT LONG-TERM STABLE C++ AND PYTHON LIBRARY APIS 20
4.1.6 ENABLE MORE ADVANCED AUTOMATED SIMULATION WORKFLOWS OF BIOEXCEL TOOLS 21
4.1.7 DEPLOYMENT AND ACCELERATION FOR NEW ARCHITECTURES AND ACCELERATORS 21
4.1.8 MOVING MORE ALGORITHMS TO ACCELERATORS – CONSTRAINTS AND UPDATE ON GPUS 21
4.1.9 SUPPORT FOR FAST MULTIPOLE ELECTROSTATICS FOR BETTER SCALING 22
4.1.10 NEW C++11 TEMPLATED SHORT-RANGED KERNEL IMPLEMENTATION, AND SUPPORT FOR TABULATED INTERACTIONS FOR BOTH CPU AND GPU ARCHITECTURES 22
4.1.11 MODERNIZATION AND MODULARIZATION OF SIMULATION PREPARATION AND ANALYSIS TOOLS 23
4.1.12 MODULAR INTEGRATION FRAMEWORK 23
4.1.13 MOLECULAR DYNAMICS APPLIED TO FLOW 23
4.1.14 FLEXIBLE INPUT FORMAT 24
4.1.15 DRIVING SIMULATIONS FROM EXTERNAL INPUTS 24
4.1.16 INTEROPERABLE TRAJECTORY AND ENERGY FILE FORMATS 24
4.1.17 CONTAINER-BASED DEPLOYMENT 24
4.1.18 CONTAINER-BASED CONTINUOUS INTEGRATION TESTING 25
4.2 LONG-TERM DEVELOPMENT PLANS FOR PMX 25
4.2.1 LIGAND MODIFICATIONS 25
4.2.2 TEST SETS 26
4.2.3 PMX WEBSERVER 26
4.2.4 TIGHT INTEGRATION WITH GROMACS 27
4.2.5 AUTOMATED FREE ENERGY CALCULATIONS 27
4.3 LONG-TERM DEVELOPMENT PLANS FOR HADDOCK 28
4.3.1 COMPLETE REWRITE OF THE HADDOCK WEBSERVER 28
4.3.2 BUILDING CURRENT WEB SERVER PRE- AND POST-PROCESSING STAGES INTO THE HADDOCK WORKFLOW 29
4.3.3 REPLACING CNS BY GROMACS? 29
4.3.4 IMPROVED, INTERACTIVE VISUALIZATION AND ANALYSIS OF RESULTS 30
4.3.5 BENCHMARKING AND RELEASE OF HADDOCK2.4 30
4.3.6 TOWARD EXASCALE INTERACTOME MODELLING WITH HADDOCK 31
4.3.7 HADDOCK-GROMACS WORKFLOW 31
4.4 LONG-TERM DEVELOPMENT PLANS FOR CPMD QM/MM 32
4.4.1 COUPLING TO OTHER MM CODES 33

4.4.2 POLARIZABLE MM CODES 33
4.4.3 MULTIPLE TIME STEPS 33
4.4.4 TIME-DEPENDENT DENSITY FUNCTIONAL THEORY 34
4.4.5 MACHINE LEARNING MODELS AND TOWARDS EXASCALE 34

5 CONCLUDING REMARKS 35

1 Introduction

A key focus for BioExcel is on the development of bio-molecular simulation software that has the feature set, documentation, reliability, scaling, and performance to meet the needs of its user community over the long term. Its users need the ability to run large workflows on the wide variety of possible hardware found in the future at the Exascale, and to be able to use their existing departmental clusters, cloud resources, and personal laptops to do a wide variety of simulation science. That software necessarily runs on real hardware that has both procurement and running costs, and so the software should be designed to use all kinds of hardware efficiently. This directly improves the ability of end users with a fixed budget to resolve a scientific question quicker, or answer more questions, or answer broader questions.

Such considerations have not changed significantly since our previous report in Deliverable 1.3.[1] There are new kinds of relevant hardware available now, and there has been progress along some industry roadmaps, but the overall outlook for biomolecular simulation software for seizing opportunities and managing risks is pleasingly similar. Accordingly, this report will take the form of an update to Deliverable 1.3. In section 2, it will focus on the aspects of the hardware and software environment that have changed (or are expected to change), and note the expected impact on the pilot applications. In section 3, it will update the development roadmaps for the pilot applications to meet both those challenges and the evolving needs of scientific users. Deliverable 1.3 included an Annex suggesting possible timelines for implementing some of the application roadmap items. These timelines were difficult to project and proved too optimistic, so we have omitted them from this deliverable, and note that in BioExcel-2 there is an early milestone to produce a more detailed plan that should be more useful to the development teams in working towards their goals for users.

It should be noted that some BioExcel software developers have current non-disclosure agreements with multiple hardware vendors, which have been honored in the preparation and publishing of this document.

2 Hardware/software assessment for the biomolecular simulation community

The overall requirements of biomolecular software have not changed drastically since the first version of this deliverable and the summary in Table 1 below (updated from that of Deliverable 1.3) is still viable. The interested reader is referred to D1.3 for further details of these requirements. PRACE has recently published an updated scientific case (http://www.prace-ri.eu/third-scientific-case/), which discussed scientific challenges and opportunities in our areas. Its Editor-in-Chief, Prof. Erik Lindahl also serves as the BioExcel Chief Scientist, and so in BioExcel we are well attuned to the latest developments – we both actively

[1] https://dx.doi.org/10.5281/zenodo.574605

influence policy directions to have future systems meet the needs of the science, and help adapt both BioExcel and external software to the resulting investments (which are necessarily based on compromises, including commercial ones).

Table 1. Exascale hardware aspects relevant to biomolecular simulation software.

Exascale aspects	Requirements
HPC System Architectures and Components	large width vector units, low-latency networks; high-bandwidth memory; fast transfer rates between CPUs<->accelerators; heterogeneous acceleration, floating-point
System Software and Management	dynamic (task) scheduling, support for adaptive scheduling of workflows, more capable batch queue resource managers
Programming Environments	standardization, portability, task parallelism, fast code driven by e.g. Python interfaces, implementations accessible also at sub-Exascale levels
Energy and resiliency	distributed computing techniques to handle resiliency/fault tolerance
Balance Compute, I/O and Storage Performance	post-processing on the fly, data-focused workflows, handling lots of small files in bioinformatics
Big Data and HPC Usage Models	proximity of data generation and analysis/visualization resources, workflows, machine learning for analyzing simulation data, high-throughput sampling, efficient HPC and HTC job scheduling
Mathematics and Algorithms for extreme scale HPC systems	multi-scale algorithms, task-parallel algorithms, electrostatics solvers, ensemble sampling & clustering theory, ensemble simulations

Over the whole HPC sector, significant progress has been made when it comes to energy efficiency. While the most energy efficient system reported in Deliverable 1.3 would have drawn 100 MW when scaled to the exascale, the most energy efficient system on the Green500 list from November 2018[2], RIKENs Shoubu system B utilizing the PEZY-SC2 hybrid accelerator, achieves 17.6 GFLOPS per Watt; scaling this system to the Exascale would draw over 56 MW.

Specific highlights since Deliverable 1.3 include the discontinuation of Intel's Xeon PHI developments, the appearance of first ARM-based HPC systems, the re-appearance of vector processors, and the more firm plans towards a European Processor and European Accelerator. On the memory and storage side, the anticipated increased usage of high-bandwidth memory and non-volatile memory express (NVMe) based burst buffers has been brought to market. In addition,

[2] https://www.top500.org/green500/lists/2018/11/

hardware for HPC and Big Data/AI applications is converging, with more and more systems being built to support both.

2.1 Processors

Deliverable 1.3 anticipated the combination of "heavy-" and "light-weight" CPUs, and this approach is gaining traction, particularly through the recent DOE systems *Summit* and *Sierra* that are using IBM Power processors together with Nvidia GPUs (for which BioExcel codes have already been accelerated). At the same time, there are still large-scale systems being built using homogeneous CPUs, most prominently the Japanese *Post-K* system that will be using Fujitsu ARM processors (BioExcel codes are already optimized for K, and we have recently started collaborations with RIKEN/Fujitsu to optimized for *post-K*). A common feature, though, is the use of wider SIMD/SIMT (vector)-units and lately also of "higher-order" matrix-matrix instructions like NVIDIAs tensor cores[3] or Intels VNNI[4]. GPU accelerator usage is still dominated by NVIDIA's CUDA language, or the OpenACC extensions that in theory should be portable, but current compiler implementations only target NVIDIA hardware (PGI is owned by NVIDIA).

Although some contenders have disappeared from the market, most prominently the Intel Xeon PHI manycore systems, new ones are gaining importance. ARM systems are now becoming available from multiple vendors and AMD is re-establishing itself as serious alternative to Intel on the x86 side. NEC is also re-entering the HPC market with their Vector processors and Europe is making serious plans to build a European Processor and Accelerator. Although it is not fully decided yet, it is anticipated that the European Processor will use ARM technologies, while the accelerator will be RISC-V based. These systems are expected to appear by 2023 with one of the European exascale systems planned by then being equipped with them. Other, perhaps less commonly used developments include Sunway's SW26010 processor [5] or the Matrix-2600 accelerator[6] designed by NUDT.

With these developments, our recommendation for Biomolecular simulation software to port all key implementations to accelerators and focus on hybrid hardware architectures till remains valid, although ports to ARM-based systems

[3] NVIDIA "NVIDIA Tesla V100 GPU Architecture", WP-08608-001_v1.1, http://www.nvidia.com/object/volta-architecture-whitepaper.html

[4] Sujal A. Vora "Future Intel Xeon scalable processor (codename: Cascade Lake-SP)", HotChips30, 2018, http://www.hotchips.org/hc30

[5] Jack Dongarra "Report on the Sunway TaihuLight System", University of Tennesse Tech Report UT-EECS-16-742, June 2016, http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf

[6] Jack Dongarra "Report on the Tianhe-2A System" University of Tennesse Tech Report ICL-UT-17-07, September 2017, https://www.icl.utk.edu/files/publications/2017/icl-utk-970-2017.pdf

need to be seriously considered, too. Experiences with existing systems have however shown that this is rather straightforward. To what extent Vector processors will be relevant is currently difficult to assess. For accelerators, we believe it is imperative that more codes implement support for open standards such as OpenCL or OpenMP 4.5 (this is already supported for GROMACS as part of BioExcel), in particular to remove architectural restraints for the EuroHPC systems, but also to ensure portability and competition in the market.

2.2 Memory and I/O

While Hybrid Memory Cube (HCM) technology has so far not seen wide uptake in the HPC market, High Bandwidth Memory (HBM) is being increasingly used, particularly on accelerators (both GPU and Vector processors), but also on standard CPUs. HBM is currently used in both high-end NVIDIA Volta and AMD Vega GPUs and in NECs SX-Aurora TSUBASA[7] vector accelerators. Furthermore, Fujitsu is planning to use HBM in their ARM-based A64FX processor[8] that will be used in the post-K computer. With current HBM2 technology these solutions offer up to 1.2 TB/s of per-package memory bandwidth compared to 0.7 TB/s achieved with traditional GDDR6 technology used by NVIDIAs Turing GPU architecture[9].

A parallel development is the emergence of storage class memory (SCM), with byte-addressable non-volatile memory technologies, i.e. NVDIMMs, perhaps being the most disruptive in terms of applications development. To fully exploit the capabilities of these new memory technologies, applications will have to utilize direct-access, kernel-bypass mechanisms to enable load/store-instruction based operations. Furthermore, special precautions in handling the performance critical caching of data in volatile memory will have to be taken to ensure consistency of the persistent information stored in the backing non-volatile memory. To ease the burden on application programmers, Intel has started a community-centered initiative for persistent memory programming[10].

While some applications like MD, whose memory requirements can be satisfied by L3 or L2 caches, will profit less from increased memory bandwidth, we expect significant benefits for biomolecular applications at large. Yet, in particular when using GPU-type accelerators, memory latency will have a significant impact on the performance. How to efficiently use the deeper memory hierarchy and who will manage data transfer (operating system, runtime system, application layer) is still

[7] Yohei Yamada, Shintaro Momose "Vector engine processor of NEC's brand-new supercomputer SX-Aurora TSUBASA", HotChips 30, August 2018, http://www.hotchips.org/hc30

[8] Toshio Yoshida "Fujitsu high performance CPU for the post-K computer", HotChips 30, August 2018, http://www.hotchips.org/hc30

[9] NVIDIA "NVIDIA Turing GPU architecture", WP-09183-001_v01, https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf

[10] Persistent Memory Development Kit Team, https://pmem.io/

an open issue that will need to be resolved. Application designers need support for new hardware to be provided in middleware (including back ends of compilers, libraries and runtime systems), because they cannot invest heavily in non-portable code that supports emerging technologies that may not prove useful or widely available.

Additional storage layers, like block addressable NVMe, are also gaining traction as expected. Many current systems use these layers as "burst buffers", providing I/O caches and checkpointing facilities. Alongside technological improvements, we anticipate largely enhanced usage models. A use-case particularly important for biomolecular applications is efficient data sharing among different applications in an ensemble or workflow, where for example multiple meaningfully different simulations can start from an almost identical input data set. This, and other technologies, are continued to be developed in the SAGE-2 project,[11] with which we have tight links.

Biomolecular applications will particularly benefit from pre- and post-processing of trajectories closer to the storage as well as dynamically changing simulations based on-the-fly analysis of the results of the previous time step. This will require application developers to refactor their code to understand the capabilities of multiple kinds of I/O environments, once the form of such APIs becomes clear, and then tailor the behavior at run time to make efficient use of what is available. However, until such systems are likely to be in the hands of bio-molecular simulation scientists, the impact from such code development effort will be quite low.

In addition, the convergence of HPC and HPDA applications is facilitated by the abovementioned developments. More and more important scientific questions are data driven and rely on the analysis of large amounts of raw data. Free energy studies via molecular dynamics simulations, biomolecular recognition for drug screening, macromolecular formation and reactions, studies of dynamic pathways such as ion transport or protein/DNA/saccharides/ligand interactions are some of the cases in which data is being generated computationally. Enabling on the fly pre-/post-processing via efficient workflows and executing them as close to the storage system as possible will considerably improve the time to solution. Similarly, "wet-lab" experiments such as cryo-EM also rely on the fast processing of massive amounts of imaging data, which ideally should be done as close to the source as possible.

Even though it is currently difficult to predict the nature of the impact of the upcoming I/O layers, this space is should be watched since it could provide significant opportunities if libraries are transparent or provide standardized APIs that can be used easily in biomolecular applications.

[11] http://sagestorage.eu/sites/default/files/Sage%20White%20Paper%20v1.0.pdf

2.3 Network

Since Deliverable 1.3, several interesting developments happened on the network side. In particular, evolutionary advances in the integration of large numbers of high-speed serial I/O interfaces (SerDes) onto a single chip are supporting a slow increase of switch radix towards 64 4-lane ports. In this context the announcement of an Ethernet switch supporting up to 256 single-lane ports by Barefoot networks [12] provides an interesting outlook on future high-fanout network topologies. Cray has announced the Slingshot network for their new Shasta architecture. The first implementation will support line rates of 200 Gb/s using custom 64-port switches and will be compatible with standard Ethernet to allow seamless integration of third-party hardware into the traditionally isolated high-performance interconnect of Cray supercomputers. With regards to intra-node interconnects, NVIDIA has developed an 18-port crossbar switch called NVSwitch[13] for their proprietary inter-GPU NVLink protocol. The NVLink protocol, introduced with NVIDIAs Pascal architecture, allows one GPU to transparently access another GPUs physical memory. Building on that the presented switch allows to create a scale-up GPU system in which up to 16 Tesla V100 GPUs can be connected to form a single 125 TFLOP/s accelerator cluster with shared memory access to a total of 512 GB of HBM2 memory. This could enable applications of GPUs to problems that so far have exceeded the memory capacity available on single accelerators. It could also mean that key biomolecular simulation software such as molecular dynamics simulations (e.g. GROMACS, where NVIDIA and BioExcel have joint co-design projects targeting both compute and network parts) could benefit from lower-latency implementations of critical halo-exchange operations.

Still, latency and small message performance seem to continue to lag processor and capacity improvements, requiring larger messages to utilize the available bandwidth. The BioExcel applications need to keep this in mind as a design constraint, but it is equally important that future hardware design and investments are guided by the needs of the programs that justify the investments. Current technology trends favor implementation strategies that reduce the need for all-to-all communication. As a consequence, fast-multipole electrostatics in GROMACS, actively developed by WP1, may prove more effective than PME at the Exascale, assuming that the trade-off between speed and accuracy can be effectively resolved. There are also clear indications that stringent limits on the maximum number of nodes over which latency-bound problems like molecular dynamics can be efficiently distributed will be imposed by network hardware capabilities. This has been particularly challenging on previous systems e.g. from Cray, where BioExcel has provided feedback about the problems of stochastic network congestion and latency fluctuations, including a BioExcel keynote by Lindahl at the 2018 Cray User Group. Preliminary results indicate many of these concerns have now been addressed in the new *Shasta* architecture announced just

[12] Patrick Bosshart ”Programmable forwarding planes at terabit/s speeds”, HotChips 30, August 2018

[13] Alex Ishii et. Al, ”NVSwitch and DGX-2 NVLink switching chip and scale-up compute server”, HotChips 30, August 2018

months ago. However, even with this type of improvements, it is unlikely that a single molecular dynamics simulation will run *efficiently* on a million MPI ranks. Instead, development strategies should target algorithms that couple large numbers of parallel simulations require it, which can provide much higher absolute efficiency (including superscaling) compared to single simulations running at low scaling efficiency. The focus of the Exascale HPC investments will be how to use the system as efficiently as possible to solve key application problems rather than more synthetic benchmarks.

Another long-going debate is what tasks should be executed by the network devices (NICs and switches) and what best to leave to the more general-purpose processors in the system. Here the Omni-Path architecture is leaning more towards "on-loading" or leaving more work to the application or system processor side of the system, whereas Mellanox and Atos try to offload more chores to the network, including some form of application specific processors into the network. The critical point here is to what extent applications can make use of the off-load capabilities offered by a network and to what extent intermediate libraries like MPI are adopted to do so. In extreme cases, as already demonstrated in the financial industry, parts of the application may be implemented embedded into the network, a strategy that may require careful planning and analysis of the costs and benefits. Another interesting trend is the ability to access resources on remote nodes. Such resources might include memory, accelerators or storage, without the intervention of any of the nodes' processors. For applications, this could permit checkpoint data to be pushed to remote nodes for safekeeping without impacting the perhaps totally unrelated computations at the destination.

In conclusion, it is still likely that applications will see a steady increase in network bandwidth, yet more and more care will be necessary to actually utilize the speed offered by next generation network technologies. This may require careful placement of data-structures in memory and utilizing multiple NICs in a node to release pressure on the intra-node interconnect; accessing network functionality from a sufficient number of threads to compensate for network latency; and reformulating algorithms to take advantage of the off-loading and in-network capabilities for instance by using non-blocking composite network operations like collectives.

2.4 Convergence of HPC and HPDA/AI Architectures

With the increasing importance of High Performance Data Analytics (HPDA) and Artificial Intelligence (AI) applications also in the traditional HPC domains, large-scale systems are increasingly being built to support both kinds of applications. This not only allows to run both workloads, but also to build efficient workloads combining HPC and HPDA/AI components.

The numerical intrinsic complexity in solving the quantum problem (Schroedinger equation) with an accuracy good enough to be experimentally useful, limits the possibility to perform routine electronic structure calculations and high throughput screening at quantum level.

The availability of exascale resources, in combination with efficient QM/MM interfaces (like MiMiC for the CPMD code, developed in WP1) and parallel enhanced sampling techniques (e.g. Multiple Walkers Metadynamics, Replica Exchange Molecular Dynamics) would open the way to perform ab initio ligand screening, i.e. virtual screening based on accurate first-principle free energy calculations and not simply on predictions based on generic chemical properties from large libraries of compounds. Parallel enhanced sampling techniques that allow speeding up the free energy reconstruction by exploiting almost embarrassingly parallel schemes are already available. However, the major bottleneck for reaching a high throughput screening based on QM/MM simulations is still the computing the quantum part of the forces.

It has been shown that the task of repetitiously solving the Schroedinger equation for those forces can be mapped onto a computationally efficient, data-driven supervised machine learning (ML) problem.[14] In these models, expectation values of quantum-mechanical operators are inferred in the subset of chemical space spanned by a set of reference molecular graphs, enabling a speedup of several orders of magnitude for predicting relevant molecular properties such as enthalpies, polarizabilities, and electronic excitations.[15] QM reference calculations provide training examples. After training, accurate property predictions for previously unseen molecules can be obtained at the base cost of the underlying ML model, provided that the new query molecule lies close to the space spanned by the reference data. So far, this technique has been only applied to small molecular systems treated fully quantum mechanically.

The improvement of machine learning technologies will have a big impact towards the realization of accurate ab initio ligand screening, by enabling machine learning-based QM/MM calculations for large biologically relevant systems as well. In future work on QM/MM for biomolecular simulations, such as planned for BioExcel-2, a careful eye will be needed to ensure that developments are portable to QM codes that have successfully deployed such methods, as the efficiency of such calculations could revolutionize the field.

As emphasized in the updated PRACE Scientific Case for Computing in Europe, both the AI workloads and the widespread adoption of ensemble approaches to scaling (thousands of simulations each using many nodes) will put hard requirements on future systems allowing jobs consisting of many loosely coupled tasks, as well as cloud-like access models with on-demand computing common e.g. for AI applications.

[14] M. Rupp, A. Tkatchenko, K. R. Müller, O. A. von Lilienfeld, Phys. Rev. Lett. 108, 058301 (2012); R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld, O., J. Chem. Theory Comput. 11, 2087–2096 (2015) / G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, O. A., New J. Phys. 15, 095003 (2013)

[15] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K-R. Muller, A. Tkatchenko, J. Phys. Chem. Lett. 6, 2326–2331 (2015)/ R. Ramakrishnan, M. Hartmann, E. Tapavicza, O. A. von Lilienfeld, J. Chem. Phys. 143, 084111 (2015)

With enough modularity, embarrassingly parallel software such as HADDOCK can make use of HPC resources to run specific parts of its workflow in the context of the modelling of thousands of complexes. Simulation of such interactome results in several thousands of docking runs that must be performed in parallel, each of them on dedicated nodes in order to reduce the processing time, reduce the communication overhead and optimize the computing resources usage. This is discussed further in sections 4.3.2 and 4.3.6.

3 Co-Design and Extreme Scale Demonstrators

An important milestone towards European Exascale systems will be the Extreme Scale Demonstrators (ESDs), conceived by the ETP4HPC, that will showcase various technological trends and allow to assess their potential towards the Exascale in a realistic setting. "*The EsDs are complete hardware and software systems designed in a strong co-design relationship between technology providers and application providers, which can be used in a production-like mode. They should facilitate fast commercialisation of the architectures and technologies, and thus, become a basis to build European capability in Exascale.*"[16] BioExcel has been driving the application focus in the ESD discussions, coordinating input from all the CoEs through a series of workshops. The findings and requirements reported in this deliverable have also been made available to the core ESD group as proper co-design from the beginning will be mandatory for building usable systems. The developments of BioExcel software are expected to make them efficiently usable on these future platforms, if the overall system design takes the requirements developed for biomolecular software into account.

With the establishment of the EuroHPC Joint Undertaking, it has become uncertain whether the ESD concept will be realized as originally planned or rather aspects of it will be adopted in the further development of the European Processor Initiative (EPI). In any case, BioExcel is prepared to work with relevant efforts and BioExcel partners have engaged in various co-design activities, both with individual vendors and larger consortia. Table 2 below gives an overview of these activities:

Table 2. Co-design activities between BioExcel partners and hardware vendors or consortia.

Partners/Project	Activities
KTH/**NVIDIA**	Fine tuning of CUDA kernels for GROMACS; improved build system support; curation of micro-benchmarks for correctness testing
KTH/**Intel**	Implementation of OpenCL interfaces in GROMACS for support of current and future streaming architectures;

[16] ETP4HPC Recommendations for the WP18-20

KTH/**Intel**	Fine tuning of GROMACS for Intel's Knights Landing
KTH/**Intel**	Acceleration of RELION on Broadwell, Skylake and Knights Landing
KTH / **RIKEN & Fujitsu**	Acceleration of GROMACS on K & post-K HPC systems
KTH/FZJ/MPG **SPPEXA project** and KTH/**Tokyo Tech**	Design and testing of fast-multipole method implementations in GROMACS
EPCC/**CRESTA project** (http://cresta-project.eu)	EPCC led CRESTA, one of the first three Exascale projects, which focussed on software challenged at the Exascale using a software co-design approach. One of its six key applications was GROMACS.
EPCC/**NextGenIO project** (http://www.nextgenio.eu/)	EPCC is currently leading the NEXTGenIO hardware co-design project which is developing a new HPC system based on Intel's next generation Xeon processor and their revolutionary 3D XPoint non-volatile memory technology which promises up to 6TB of NVRAM in DIMM form for each processor socket. Both HPC hardware and software, particularly focussed on data scheduling, are being developed to exploit this revolutionary new memory technology.
BSC/**MontBlanc project** (http://www.montblanc-project.eu)	MontBlanc is developing an Exascale architecture based on ARM processors; exploring different alternatives for the compute node from low-power mobile sockets to special-purpose high-end ARM chips.
Juelich, BSC/**DEEP-ER project** (http://www.deep-er.eu)	DEEP-ER is building a prototype based on the second generation Intel® Xeon Phi processor, a uniform high-speed interconnect across Cluster and Booster, non-volatile memory on the compute nodes, and network attached memory providing high-speed shared storage; optimising a selection of HPC applications for that system.

	The project runs bi-weekly inter workpackage meetings where interdisciplinary design and development topics are discussed. Workshops with application developers are also regularly organized.
KTH, Juelich/**Sage Storage Project** (http://www.sagestorage.eu)	SAGE improves the performance of data I/O and enable computation and analysis to be performed more locally to data wherever it resides in the architecture, drastically minimising data movements between compute and data storage infrastructures. With a seamless view of data throughout the platform, incorporating multiple tiers of storage from memory to disk to long-term archive, it enables API's and programming models to easily use such a platform to efficiently utilize the most appropriate data analytics techniques suited to the problem space.

BioExcel regularly monitors the output of hardware-software co-design projects targeting Exascale application deployment. EU-supported projects such as NextGenIO, MontBlanc, DEEP-ER, and Sage are designing and building prototype hardware and software. Similar efforts are underway in the US, China, and Japan. Our observations as application designers of these projects are that we should plan to depend on standard languages, compilers, runtimes, and libraries. We expect that new hardware will, wherever possible, support those, because it is not sustainable for multiple applications to port to custom infrastructure whose lifetime is uncertain. For example, new low-latency network hardware needs to come supported by modules that make them work in readily available MPI or PGAS libraries, rather than expect application designers that already support MPI to write a layer that functions like MPI. Similarly, while power consumption is a key design constraint for future Exascale machines, application designers can only take advantage of the capabilities of the hardware to the extent that a callable API exists, or that libraries like hwloc will accurately report on it. For example, GROMACS has almost no need of the memory provided on typical HPC systems. However, there is no API that permits a GROMACS simulation to instruct the operating system to turn off normal memory that is not required, which would save power.

4 Long-term development plans for BioExcel pilot applications

As mentioned in the introduction, the BioExcel application codes both consider the likely path of future hardware in their long-term planning and define the needs of the applications that should help guide future hardware development investments. Both the BioExcel codes and most other proven-impact scientific applications have solved this by choosing languages such as C/C++/Fortran and Python that have very large user communities and outstanding track record of portability. For programs such as GROMACS (where individual simulations are intended to be suitable for Exascale deployment) there are also a number of architecture-specific language extensions and libraries used. Additionally, scientific users need the simulation software to be both highly usable and highly adaptable, which drives the plan to make BioExcel software available via both command-line applications and workflows using API/web interfaces.

Public support for work on free and open-source software infrastructure is essential for making progress for usability and impact at the Exascale, and the BioExcel applications have been selected based on their proven impact track record. The pilot codes represent a selection with very different degrees of parallelization maturity, not to mention completely different challenges for parallelization as well as development quality assurance (QA). The strategy followed in the project is to help each of these user communities improve towards Exascale, but on their terms. For the selected applications, the BioExcel effort has particularly focused on improving support for new architectures, scaling, usability and sustainability in terms of better code documentation and testing, and gradually helping the broader communities involved in these codes to move to more modern development standards (which is not easy given the code size and scientific requirements). In addition to work directly funded by BioExcel, the project has also taken far-ranging responsibility for integrating development efforts as well as identifying future development needs for the applications where additional support will be required (i.e., beyond BioExcel).

4.1 Long-term development plans for GROMACS

GROMACS is a mature project that has a core development team at KTH in Stockholm, and numerous regular external contributors. In late 2018, there were eight full-time developers, six regular part-time developers, and numerous occasional contributors. BioExcel supports only a small part of the team, but for the first time this has enabled more professional project management including processes for both internal QA and external collaborations, not to mention a plan for required future work where BioExcel is able to coordinate and steer a much larger number of developers and contributions from outside of BioExcel.

A large number of important directions for future development have been identified. These are listed here, but are not ordered into a formal timeline. In practice, the requirements of other available funding resources will shift, and rarely can any developer focus solely on one or a few tasks in order to deliver it by a predictable time, given that the feature often depends on several others and the team spends an increasing part of their time on QA. The skill set of available developers often determines which targets can receive most effort. Progress is also contingent upon the availability of resources for testing, and since GROMACS has moved to full formal code review, it also requires time for the review of the code by other expert developers. It should be noted that GROMACS does not only target bio-molecular simulations, and the software is increasingly being used in other areas. For the first time, several of the targets reflect the needs of e.g. materials science simulations. Finally, as part of BioExcel and the mission of supporting all types of simulations for the community, the GROMACS project has started much closer collaborations (in particular for training) with Amber, NAMD and OpenMM, which are the other major codes in the field, and these new collaborations are likely to influence future development of all the codes.

4.1.1 Modularization of core mdrun simulation functionality

One of the key requirements to achieve a flexible code that can be drive by other programs to deploy workflows for users on the Exascale is to make the core routines modular and testable with modern QA strategies. It is also necessary to create the flexibility to deploy to future hardware features and architectures with minimal disruption to the code base while making it easy to see that implementations are correct and maintainable. This work will continue to port hundreds of thousands of lines of C89-style code to C++14, including use of RAII-style resource management, Doxygen developer documentation and unit testing coverage for old code. Several fundamental changes are required, including robust error handling that does not simply terminate the program, relaxing assumptions that all available hardware can and should be used, and that hardware will not fail over the lifetime of the simulation. This work is closely related to the new specifications of software engineering, testing and QA previously reported in BioExcel Deliverable 1.1, and covers all tasks reported below.

4.1.2 Evolution of Exascale-suitable task parallelism

The main challenge for molecular dynamics simulations to be relevant for Exascale deployment is to improve strong scaling for the often relatively small systems used in concrete biomolecular applications. While GROMACS has a good performance and scaling reputation, improving this even further is constant high priority, in particular for BioExcel. The major workload of these simulations is the computation of forces, and this often requires the execution of several very large kernels, and numerous small kernels. The latter will generally not parallelize over large numbers of cores because the overheads of preparing to run the calculation take an amount of time comparable with the execution time of the kernel. The problem is even more challenging when multiple computing units (GPUs, sockets of CPUs, nodes) have to coordinate work, because units need to do local work when that is all that is available, but be prepared to react to data from other computing units as soon as it becomes available. As we approach the Exascale, the number of computing units that need to be used will grow rapidly. Continuing to improve strong scaling that functions on a wide variety of hardware architectures and configurations will require a kind of adaptive scheduling that requires a task graph, rather than a pipeline of kernels that are assumed to be able to be spread over all available resources when it is time for each to be run. This will require a complete overhaul of the entire execution back end of GROMACS, which is supported by numerous of the code modernization and modularization activities mentioned here.

4.1.3 Feature enhancements to support docking workflows

A key outcome of collaboration with HADDOCK developers in BioExcel has been to identify the required set of new and improved features within GROMACS that would permit replacing the proprietary CNS molecular mechanics engine currently used in HADDOCK. New highly-ambiguous interaction forms are needed, along with support for rigid-body movement. The existing highly flexible molecular selection machinery must be re-deployed to permit dynamic selection support in mdrun, which aligns with other efforts intending to re-use the same machinery. The combination of GROMACS performance and HADDOCK flexibility may prove to be a game changer for biomolecular simulation at the exascale.

4.1.4 Updated and improved sampling algorithms for free energy calculations

While strong-scaling efforts provide one path to Exascale execution (which we are already pursuing for GROMACS in BioExcel), it will not be possible to efficiently parallelize a single MD simulation on hundreds of thousands of particles over millions of processing units. Supporting coupled sets of simulations is a key part of the GROMACS Exascale strategy. Numerous special-purpose algorithms are already implemented, including replica-exchange, extended-ensemble, umbrella sampling, free-energy perturbation, the adaptive weight histogram method, and computational electrophysiology. These require test suites and code updates to provide better tools for simulations that use computing resources more efficiently. We are working towards Exascale-era simulations using advanced sampling algorithms, such as Markov state models, milestoning, the string method with swarms of trajectories, and not least adaptive free-energy sampling. These will be implemented by an adaptive workflow engine that manages thousands of independent simulations within a high-performance environment. The Copernicus workflow system (http://copernicus-computing.org/) developed alongside GROMACS, and other similar systems, have proved this concept, however it will not be feasible to tolerate the overheads of job scheduling and file-based I/O at the Exascale. An MPI-aware server-client binary is most likely to work best, so that the available work and hardware are matched to run efficiently, even as the required work changes and perhaps the hardware environment changes as hardware fails or other jobs monopolize computing or network resources. Other alternatives, such as in-memory file systems may be an option, but to achieve the throughput needed for Exascale, simulations will require a much tighter integration between workflow, scheduler, I/O, and the simulation engine compared to what is available today.

4.1.5 Implement long-term stable C++ and Python library APIs

A central theme for modern scientific codes is that users increasingly want to compose modules, or even call a program as a library from another. Users need the ability to design and test simulation and sampling algorithms without being developers, and they also need to be able to manipulate the resulting output in automated fashion. For the future, we want to make both the modularized core mdrun library and all other tools available behind APIs that are intended for long-term stable support and to work intuitively both in C++11 and Python 3. Some initial work is currently done in collaboration with US GROMACS developers, with additional funding by NIH grant R01-GM115790-01A1. However, providing good APIs for the entire library will require much larger effort, but it will make it significantly easier to use the code in advanced scripts. The impact is expected to be very high, since it will allow a wide range of other codes to directly use the fast GROMACS simulation engine – one example is that it could enable BioExcel to use GROMACS as the modeling engine inside HADDOCK in the future.

4.1.6 Enable more advanced automated simulation workflows of BioExcel tools

The combination of workflow engines, a stable Python API, and advanced sampling algorithms will make it possible to automatically combine and integrate tools for automatic parameterization of drug-like molecules, and the PMX topology generator developed as part of BioExcel. This will enable users to deploy fully automated workflows in a high-level language that can compute quantitative relative free energies of binding of families of ligands to families of proteins. Such workflows should be highly adaptable, both to use the hardware efficiently despite heterogeneity or run-time failure, and to recognize how to sample efficiently in each particular sub-problem.

4.1.7 Deployment and acceleration for new architectures and accelerators

GROMACS has a strong track record of portability, and with the new modular CPU-side SIMD support layer recently developed (in combination with extensive automated unit test frameworks) it is close to trivial to port to new architectures. This layer decouples the CPU kernels from the low-level infrastructure that implements the required memory and arithmetic operations, and the code also has a very general nonbonded kernel architecture that is easy to adjust for different hardware profiles. As new flavours of Intel, AMD, ARM and Power chips emerge and seem relevant for high-performance computation, hardware-specific acceleration support for all these architectures will be added.

GROMACS currently supports devices that run NVIDIA's proprietary GPU acceleration language CUDA, and the industry standard vendor-neutral OpenCL language (on AMD, Intel, and NVIDIA devices). It is likely that any new accelerator will likely support either of these languages or OpenMP 4.5, and the main effort to support these would be to make minor adjustments to how the work is structured and mapped to execution units. There is also interest in making GROMACS able to run on a GPU-based fully free software stack, which should facilitate interactions with and optimization for new custom Exascale hardware developed e.g. in the Exascale demonstrator projects.

4.1.8 Moving more algorithms to accelerators – constraints and update on GPUs

The short- and long-range interactions in GROMACS are already being executed on accelerators such as GPUs, but as the performance gap between accelerators and CPUs is constantly increasing there is a need to move more parts of the execution to accelerators. It is expected that the cost-efficiency of performance increases on GPUs and accelerators is likely to continue to outstrip those for CPUs, and is a key requirement for molecular dynamics to function well at the Exascale where accelerators will likely deliver most of the flops. Co-design work partnered with Nvidia is underway to port remaining CPU-side compute and data-transfer bottlenecks to the GPU, and we expect GROMACS 2020 to feature a "GPU-only" port optimized both for consumer- and professional-grade accelerator hardware. Progress here is a key stage in the evolution of GROMACS to support the expected accelerator-heavy Exascale-era compute nodes described above.

4.1.9 Support for fast multipole electrostatics for better scaling

Methods like particle-mesh Ewald have been the *de facto* standard for treating electrostatics in MD simulations the past decades. The main advantages are high sequential performance due to the use of Fast Fourier Transforms (FFT) and that they are relatively simple to implement. But the 3D-FFT is ill suited for high parallelization because of the four 3D-grid transposes required per MD time step. Both strong and weak scaling of MD simulations is limited by the latency and overhead of the all-to-all MPI communication. The fast multipole method (FMM) provides better formal scaling as O(#particles) both for computation and communication (unlike N(log)N for FFT methods). However, standard FMM have been too slow to be usable in practice, and there are issues with energy conservation, which we have recently found a solution for. The GROMACS team is collaborating with groups at the Juelich computing centre and Tokyo Tech to adapt FFM methods for MD and integrate them into GROMACS. This will provide both better strong and weak scaling, with very large impact in the high-scaling regime. Furthermore, since FMM is by default non-periodic this enables exact electrostatics for Exascale simulations of flow which often use ten to hundreds of millions of particles.

4.1.10 New C++11 templated short-ranged kernel implementation, and support for tabulated interactions for both CPU and GPU architectures

To consolidate the GPU and CPU code paths, the next stage in the evolution of the hundreds of flavours of GROMACS high-performance short-ranged kernels is to move away from bug-prone preprocessor-based code generation and instead use C++11 extern templates. This will permit code analysis and debugging tools to work flawlessly, because conditionality can be implemented within the compiler, relying on it to eliminate code branches that will be dead at run time, and we will be able to provide full unit testing for any user-specified interaction form. After support for user-provided tabulated interactions is added, this will permit the removal of the deprecated group-style cutoff scheme, along with a great deal of supporting code, which will make it much easier to handle the transition to a modular library – and this in turn will enable us to expand the types of interactions and force fields supported. Many knowledge-based force fields, particularly in materials science, produce interaction functions that are not based on an easily-described mathematical formula. The potential and force for such interactions must be looked up from tables for each particle pair. Such kernels need care because one must choose interpolation schemes that are able to run fast on the hardware, while providing a bounded error. In principle, different hardware (e.g. CPUs and GPUs) could suit different schemes because of differences in instruction and cache latencies, but all such should provide an equivalent and thoroughly tested implementation so that users do not need to be aware that the difference exists. To make use of such support, the topology description within GROMACS must be extended so that it is possible to choose tables for unique atom pairs, or pairs of all types, etc. This would significantly enhance the usability of GROMACS for non-traditional applications, including both materials science and special potentials currently only available on slow or non-scaling simulation codes.

4.1.11 Modernization and modularization of simulation preparation and analysis tools

In addition to the core simulation engine and sampling algorithm libraries, GROMACS also packages numerous popular tools suitable for preparing simulation topologies and conformations, and analyzing simulation results. It would have large impact on users if all these tools had modern error-handling and user interfaces with simple data structures that could be accessed both from C++11 and Python, although this will require substantial focused efforts.

Many of the analysis tools combine features that were developed by different people and do not work correctly in combination, or have less tested implementations contributed by individuals focused on science and collaboration, rather than proven-quality software with good unit test coverage. An extensible and parallelizable framework for implementing tools has been partly designed, however many of the analysis tools are still using old and bug-prone implementations. To improve the experience of GROMACS users, we should greatly expand the test coverage, remove duplicate and broken features, and port to the new framework. This will further permit those modules to be deployed behind the API work, which in turn will facilitate collaborations with other codes and usage in workflows. Currently users might have to convert and store multiple versions of their trajectories on disk in order to use multiple analysis tools, which we will make more usable by re-implementing the tools to handle their own conversion requirements.

4.1.12 Modular integration framework

Molecular dynamics proceeds by integrating the equations of motion. Forces on particles are computed based on the positions of all particles, those forces are used to update their velocities, after finally the velocities used to update the positions. Numerous alternatives are available, and can have subtle strengths and weaknesses. It is important for the bio-molecular simulation field to be able to assess these alternatives on real-world problems, which requires that the time-consuming force calculation are provided by a high-performance package. The current GROMACS implementations of integrators require considerable clean up and improvement before this is possible, but it is something we are planning to address on the medium term. Such modularization will also be needed in order to support a GPU-only mdrun execution path. This work is underway, with additional support by the NIH grant R01-GM115790-01A1.

4.1.13 Molecular dynamics applied to flow

As technological and biomedical flow application move down to the micro- and nano-scale, the molecular nature of liquids becomes more apparent. In recent years, that has seen a strong increase in the use of large-scale MD simulations of flow to study molecular aspects. But although what is called the nano-scale is small, it is still very large on the MD scale with systems from millions to hundreds of millions of atoms. Unlike typical biomolecular applications, which have fixed size, these flow simulations are even a relevant target for single-machine Exascale supercomputers. The main issue here is that this is not simple weak scaling. As the system size increases, the time scales also increase. Thus, there is still the strong need to minimize wall-clock time per step by using more cores. To achieve this, the same parallelization strategies as for bio-molecular simulations are required (overlapping calculation and communication and a better tasking model). In particular, the optimization and integration of the fast multipole method for electrostatics is important to significantly push up the scaling limit.

4.1.14 Flexible input format

Historically, GROMACS defined its own plain text key-value input file, implementing its own parser and requiring users to understand how to use it. This was a good decision when it was made, but now that standard formats and libraries for handling structured text input files exist, we plan to transition to using these. This will eliminate many kinds of possible bugs in the code, and support the modularization efforts elsewhere. Users will be able to edit the structured files in a syntax-aware editor of their choice. Developers extending the functionality of GROMACS will be able to make changes only in their new module, and not in multiple parts of the code.

4.1.15 Driving simulations from external inputs

Numerous new experimental techniques provide exciting opportunities for users to run simulations that inform experiment, and vice-versa. Access to input data will be facilitated by a flexible input format, while access to parallel simulation data on complex HPC systems will be facilitated by code infrastructure that will provide a unified abstraction layer to access atom information that is spread out over compute nodes. This will speed up implementation of future simulation protocols like advanced Bayesian sampling algorithms into the regions consistent with the experimental data, so that users can expect a faster and more stable support of new simulation protocols from GROMACS. Work on cryo-EM refinement is already underway supported by the Carl Trygger Foundation and BioExcel, and we see large potential in extending this to other sources of experimental or bioinformatics data.

4.1.16 Interoperable trajectory and energy file formats

Existing enhanced sampling algorithms in GROMACS write output to a variety of custom output files that will be entirely unsuitable to run at the Exascale. Even existing algorithms impose the burden on users of coordinating dozens to thousands of output files. GROMACS currently supports the TNG next-generation trajectory format (http://dx.doi.org/10.1002/jcc.23495), which provides best-in-field compression of containerized molecular simulation data. All kinds of mdrun output need to be written to TNG, and analysis tools updated to use the new format. One of the important outcomes from the recent BioExcel-arranged workshop on sharing of biomolecular simulation data is that we will engage in definition of common onthologies followed by open specifications both of system descriptions and file formats to make it trivial for users both to share input data between simulations and archive trajectories.

4.1.17 Container-based deployment

Users will be able to use GROMACS effectively on cloud-based resources when a container can be obtained from a repository that will recognize the hardware and run an appropriately configured pre-compiled version of GROMACS. For some families of architectures, such as x86, this could be achieved via dynamic loading of shared libraries that match the hardware, but this approach is limited to hardware capable of running the same executable. For more heterogeneous situations, a dispatch algorithm is required.

4.1.18 Container-based continuous integration testing

The wide range of hardware, build configurations, and simulation algorithms supported by GROMACS requires rigorous automated testing. Portability is a key requirement for success during the transition to the Exascale, because it is not known what compilers, architectures and technologies will prove to be successful. This can only be assured through testing. However, maintenance of dozens of test machines is too time-consuming. We plan to replace these with container-based test environments that automate build and deployment via a script. These can be used to maintain a much wider range of test environments without needing to manually log into machines to maintain software versions. A fully automated benchmark suite would also be possible to deploy in this framework.

4.2 Long-term development plans for pmx

The long-term aim for pmx development comprises a highly automated software package providing means to setup alchemical free energy calculations compatible with the GROMACS simulation engine with minimal user intervention. The pmx based hybrid structure and topology generation for biomolecules encompasses amino acid and nucleic acid mutations. The treatment of arbitrary ligands poses a challenge more difficult than handling amino and nucleic acids. For the latter cases the diversity of the chemical libraries is low and the structure sets are well defined, thus mutation libraries were pre-generated and made available to the community. In contrast, every ligand mutation is unique and needs a separate treatment.

4.2.1 Ligand modifications

To include modifications of arbitrary organic molecules into the pmx package, firstly, prior to building a hybrid ligand structure/topology, an atom mapping needs to be established for the two compounds. A number of routes exists to establish a desired mapping. One way is to rely on a graph based common substructure search to identify the morphable atoms, while the rest of the atoms ought to remain dummies. Another approach utilizes the Euclidean distance to map the atoms. This method, however, requires an alignment of the structures with arbitrary numbers of atoms. Our aim is to automate these mapping procedures and incorporate them into a single tool in the pmx package.

Having obtained an optimal atom mapping, the actual hybrid structures and topologies need to be generated. For that we are aiming to utilize the already developed pmx routines and extend them to the more general use to handle the ligand alchemical morphs.

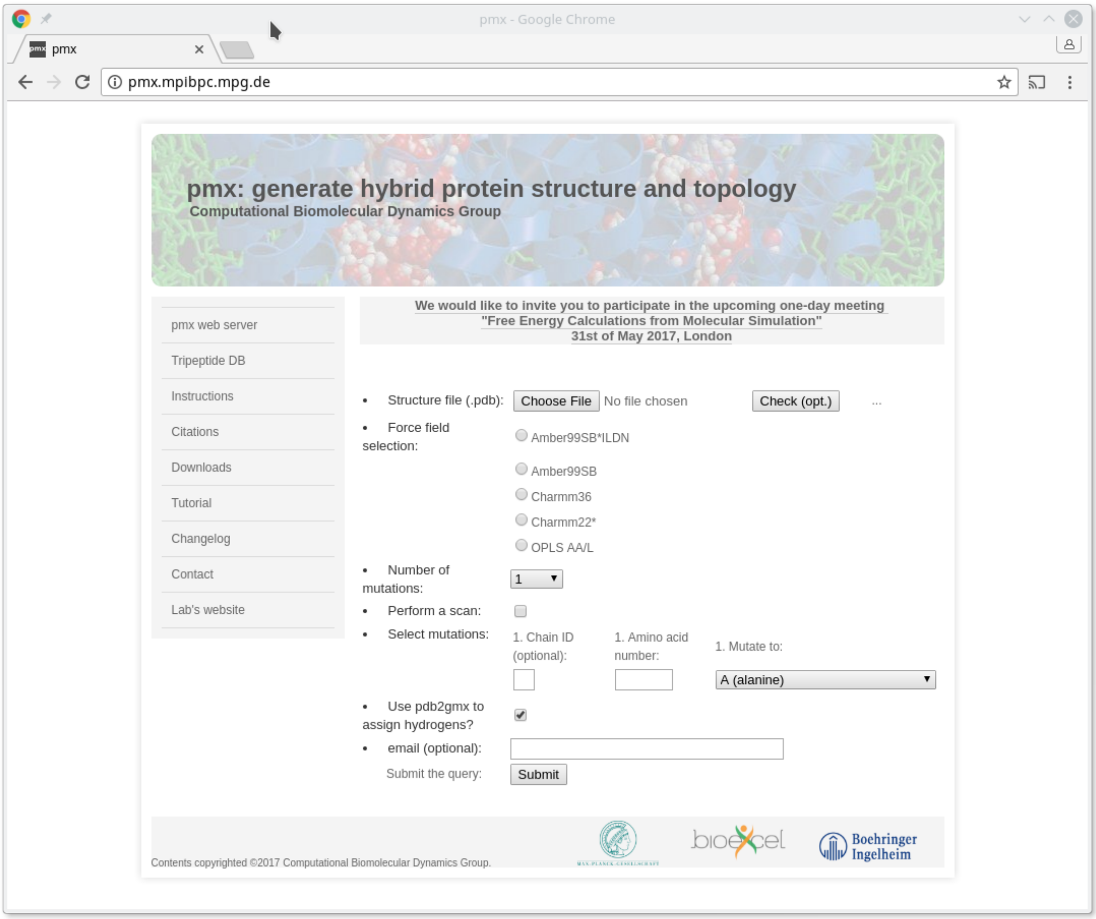
The final step in the ligand treatment is generation of the optimal ligand maps for investigating large chemical libraries. Based on the aforementioned atom mappings between the ligands, the similarity between the compounds can be quantified. This enables usage of the graph-based approaches to identify optimal paths to connect ligand pairs to fully explore the library of compounds of interest.

4.2.2 Test sets

While generation of hybrid structures for biomolecules is thoroughly tested and validated, covering and verifying correctness of the software for the whole chemical space is not feasible. Therefore, a set of tests for the created amino and nucleic acid mutation libraries, as well as the ligand hybrid structure/topologies has been created. Currently, all the consistency checks have been performed using an in-house software toolkit. We are aiming to bring this software to the level where it can be easily and robustly applied by the user to ensure the validity of a newly generated mutation library or a hybrid structure/topology. In addition to the testing software we plan to incorporate a set of short simulation protocols into pmx to check for the validity of the free energy calculations.

4.2.3 pmx webserver

In its original implementation, pmx was designed as a command line software tool. To facilitate usability of the software package, the pmx webserver has been created over the course of BioExcel project (see Figure 1). Currently the webserver provides automated support for the amino acid hybrid structure/topology generation in 5 contemporary molecular mechanics force fields. This allows setting up alchemical free energy calculations for amino acid mutations without the need to install the pmx software and therefore takes an important step in rendering complex biomolecular simulations accessible to non-specialist users. Furthermore, the web features have been extended by incorporating nucleic acid mutations in DNA. The main future plan of the webserver development entails incorporation of the alchemical ligand modification utilities. The latter provide a thorough means to perform lead optimization schemes based on accurate free energy based affinity estimates. This enables pharmaceutical workflows in academia as well as pharma industry to integrate high-quality *in silico* thermodynamic predictions and thus minimize the effort in synthetic chemistry and in vitro screening.



pmx: generate hybrid protein structure and topology

Computational Biomolecular Dynamics Group

We would like to invite you to participate in the upcoming one-day meeting
"Free Energy Calculations from Molecular Simulation"
31st of May 2017, London

- Structure file (.pdb): No file chosen
- Force field selection: ○ Amber99SB*ILDN ○ Amber99SB ○ Charmm36 ○ Charmm22* ○ OPLS AA/L
- Number of mutations: 1
- Perform a scan: □
- Select mutations: 1. Chain ID (optional): 1. Amino acid number: 1. Mutate to: A (alanine)
- Use pdb2gmx to assign hydrogens? ☑
- email (optional):

Submit the query:

Contents copyrighted ©2017 Computational Biomolecular Dynamics Group.

Figure 1 pmx webserver main page

4.2.4 Tight integration with GROMACS

In the long term we are planning to proceed with a tight PMX integration into the GROMACS package. To make this possible, we first need a stable Python environment as part of the GROMACS tools, as mentioned in the GROMACS plan, but this work is in the pipeline. The Copernicus workflow management system has a basis to support free energy calculations utilizing the GROMACS molecular dynamics engine. Therefore, it presents an attractive opportunity to integrate the pmx hybrid structure/topology generation into a fully automated setup to carry out large scale amino acid, nucleic acid mutation and ligand modification scans. Due to its GROMACS integration, PMX calculations will directly benefit from performance, optimization and scaling developments in the GROMACS package, as well as its broad, highly optimized support on a wide range of platforms and accelerators.

4.2.5 Automated free energy calculations

Having the main amino acid, nucleic acid and ligand modification modules in place opens the door to delivering the full alchemical free energy calculation package to the user. In the long term, the developed modules will be implemented in a unifying framework allowing to easily prepare relative free energy calculations utilizing a user specified protocol. In particular, the approaches based on the non-equilibrium molecular dynamics simulations offer an efficient and attractive solution, which will be of the main focus in the future PMX development

The key to bring PMX free energy calculations to the Exascale is automation: a fully automated, unsupervised workflow is a prerequisite to carry out massive mutation and ligand screens necessary for protein design and drug development. In close collaboration between the KTH and MPG teams, the PMX/GROMACS integration will be set up to achieve just that: an environment to carry out massively parallel alchemical free energy scans either in an HPC environment or in the cloud, in which only the list of end results is reported to the end user. All intermediate control, including error handling, convergence control to a user-set level, internal consistency assertion, analysis of raw data and translation to user-digestible, presentation-ready formats will all be carried out seamlessly on the fly. Such a setup would enable both academic and industrial end users, that do not need to be experts in biomolecular simulations, to operate highly complex workflows on massive mutation or ligand databases and thus make optimal use of future hardware generations for protein design as well as drug development.

4.3 Long-term development plans for HADDOCK

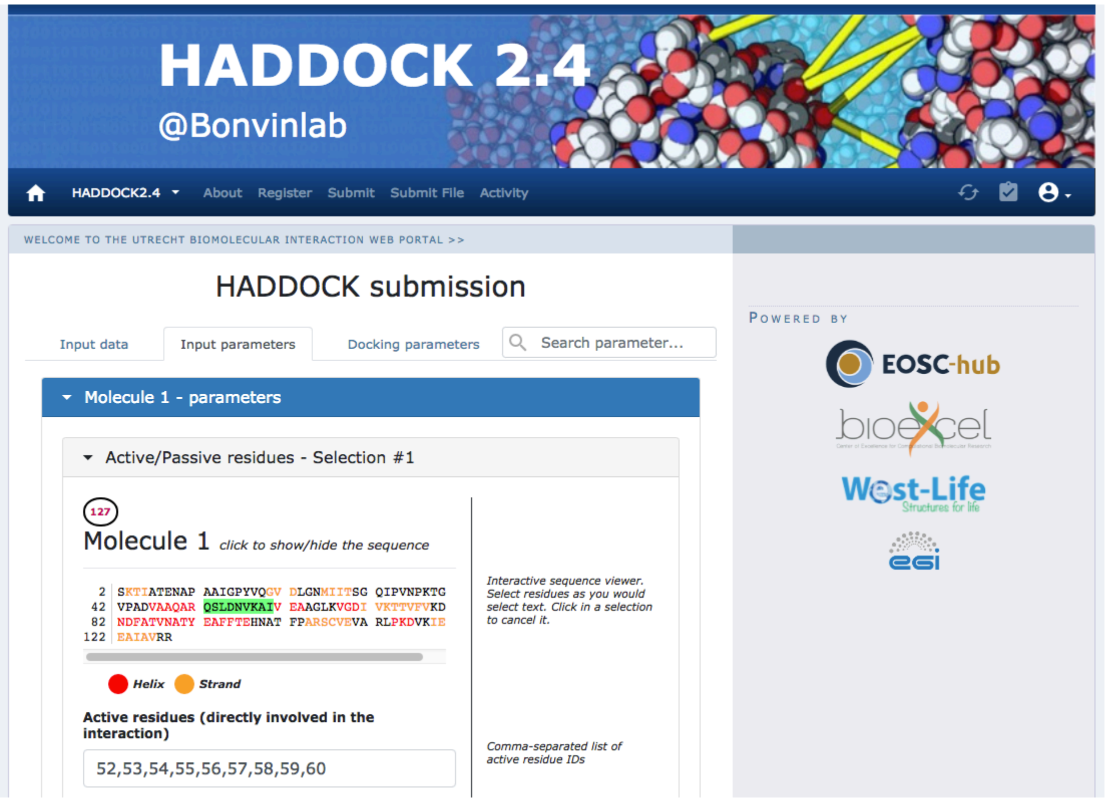
We reported in Deliverable 1.3 the development of a new version of HADDOCK at the CNS and front-end levels.

As a result, a first beta-version of the new HADDOCK web portal, labelled 2.4, is available via https://csbdevel.science.uu.nl/haddock2.4/ and it is showcasing the new design and features provided both at the front-end and software levels. Built upon the new version of the HADDOCK2.4 software, the new web portal has been developed with the Flask framework, which is a more modern approach to both front-end and back-end software development. It provides a thick layer of security and data protection to the users while facilitating the pre-processing steps of HADDOCK input data. The new portal operates from three docker containers (see below), which will give more flexibility for its deployment in the future. We expect to put this new portal into production by the end of Q4 2018, pending new hardware to host the complete suite of portals from the Utrecht partner.

Further, under BioExcel-2, we plan a complete rewrite of HADDOCK to make it modular, which will provide more flexibility in plugging in third party software to replace some current modules, couple it to Gromacs for post-docking stability simulations and allow users to customize their docking workflow. This is a major effort that will be based on a complete redesign. In parallel, we will keep maintaining and updating, when needed, the latest 2.4 version and its web portal, which we expect will be the main production versions for at least the next two years.

4.3.1 Complete rewrite of the HADDOCK webserver

As of today, all features present in the previous version of HADDOCK (2.2) have been successfully integrated into the new version (see Figure 2). On top of them, new features such as supporting cryo-EM restraints have been also integrated. Any new parameter implemented in HADDOCK2.4 is, if necessary, exposed to the user through the web interface. This includes for instance the possibility of shortening the post-processing steps by skipping computationally demanding analyses, which will improve the CPU efficiency on HPC systems.



HADDOCK submission

Molecule 1 - parameters

Active/Passive residues - Selection #1

127

Molecule 1 *click to show/hide the sequence*

```
  2 SKTIATENAP AAIGPYVQGV DLGNMIITSG QIPVNPKTG
 42 VPADVAAQAR QSLDNVKAIV EAAGLKVGDI VKTTVFVKD
 82 NDFATVNATY EAFFTEHNAT FPARSCVEVA RLPKDVKIE
122 EAIAVRR
```

● Helix ● Strand

Interactive sequence viewer. Select residues as you would select text. Click in a selection to cancel it.

Active residues (directly involved in the interaction)

52,53,54,55,56,57,58,59,60

Comma-separated list of active residue IDs

Figure 2 - New HADDOCK2.4 submission form interface

4.3.2 Building current web server pre- and post-processing stages into the HADDOCK workflow

In the current setup, the HADDOCK web portal handles a number of pre- and post-processing steps independently from the HADDOCK workflow managed at the Python level. This is the main difference between running a local version of HADDOCK and submitting jobs via the server interface.

In order to reach the Exascale for interactome modelling, we have redesigned HADDOCK pipeline to be able to execute all pre-processing steps at the web server level. This allows to generate all required input files and directories upon input data submission and then directly run HADDOCK from the generated files.

4.3.3 Replacing CNS by GROMACS?

Following the outcome of our first report on the possibility to replace HADDOCK core engine by GROMACS, we ended up postponing this task because of other priorities at the moment of the report and the lack of available effort that could be dedicated to it. There are also many features of CNS and energy functions for specific experimental data that are simply not available in GROMACS, making a full replacement by GROMACS impossible right now. As mentioned in section 4.1.14, a list of necessary features have been created by UU and discussed with GROMACS team. A summary (not an extensive list though) can be found below:

- Flexible selection syntax based on atom/residue selection and not atom/residue numbers (must be able to be done on the fly during a simulation)
- Implementation of X-ray, NMR, cryo-EM restraints, CM-restraints, symmetry restraints if not yet available in Gromacs
- Incorporation of HADDOCK effective distance calculation for ambiguous distance restraints (Sum $1/r^6$) $^{-1/6}$
- Support to N-molecules
- Automatic rebuild of missing atoms (can be outsourced to other software)
- Option to rigidify / make flexible parts of a structure (e.g. only allow on the fly flexibility of interface side-chains and backbone from a contact analysis at a given time in the simulation)
- Scriptable analysis tools (e.g. to calculate on-the-fly BSA, empirical desolvation energy)
- Automatic identification of a vector perpendicular to the interface and rotation around it (all at the script level)
- Dynamics in torsion angle space

Their feasibility and difficulty have been assessed during a hackathon in Nov. 2018 and a roadmap of the next efforts to be made has been drawn. The implementation of a number of missing features in GROMACS, together with the modularization of HADDOCK planned in BioExcel-2, we foresee that some modules in HADDOCK will in the future make use of GROMACS, while other will still depend on the CNS code.

4.3.4 Improved, interactive visualization and analysis of results

The Flask framework used underneath HADDOCK web portal allowed us to make our first steps towards a fully interactive results page. We are using the bokeh library (https://bokeh.pydata.org/en/latest/) to generate interactive plots based on HADDOCK results. Despite the amendable level of interactivity, we planned to integrate a molecular viewer in the results page and link it to the interactive plots in order to enrich the user experience of the portal.

4.3.5 Benchmarking and release of HADDOCK2.4

Tests and benchmarking of HADDOCK2.4 has already started using the beta version of the web portal. Based on a complete suite of test cases to assess HADDOCK correctness at each feature and bug-fixing version, this automated benchmark allows us to monitor any deviation in the data output by HADDOCK for different sets of application (protein/protein, protein/DNA, protein/RNA, multi-body docking, etc.). Proper benchmarking, however, requires running full regression tests, which consist in a full docking protocol to check the consistency of the results, something that remains CPU intensive. We are currently exploring the use of test at different levels of granularity in order to assess long-time and resource-expensive test cases. Those tests will aim to be automated and get proper reporting through standard Continuous Integration pipelines. They will be ranged from unit tests at modules levels (file parsing/validation, input consistency, etc.) to higher level tests like regression tests upon new parameters implementation or integration tests for case-by-case testing using the suite of test cases described above as input data.

The official version 2.4 will be released and with it a complete documentation that wraps up all new features and modalities of usage.
The web server has been developed to be easily deployed on cloud resources for usage such as workshops, courses, etc. It makes use of docker containers (Figure 3) to run the whole web portal, relying on a PostgreSQL database (either local or remote). The web portal can either link a temporarily created DB or our main user DB hosted at Utrecht University on local clusters. The deployment protocol is as much automated as HADDOCK complete workflow allows it, with only few manual steps required to make things operational. This new way of deploying the HADDOCK web server has been tested in the context of a HelixNebula Science Cloud Pilot project (see https://www.hnscicloud.eu/use-case-template).

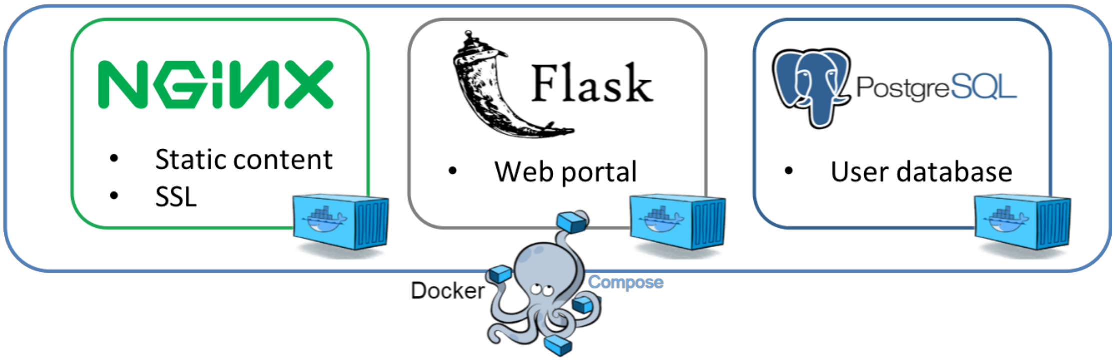


Figure 3 Schematic representation of the new Docker images interconnection and orchestration to run CSB web portals front-end.

4.3.6 Toward Exascale interactome modelling with HADDOCK

As reported in D1.5, the new web portal allows to entirely decouple all the pre-processing and validation steps from the computations. Then, the portal machinery can be used as standalone to prepare docking runs, which can be executed on HPC resources for the purpose of Exascale modelling of interactomes. This will also require wrapping the computational part of HADDOCK in a workflow manager that will handle the tens of thousands of complexes to be modelled. This has already been made possible thanks to the addition of HADDOCK as a microservice of MDStudio, which is part of UU effort in WP2. Furthermore, the new option to bypass the post-processing analysis, which is still sequential, will allow to run efficiently on HPC resources, without wasting precious CPU time.

4.3.7 HADDOCK-GROMACS workflow

As one of the BioExcel project Use Case (UC5), we started the development of a workflow involving two BioExcel flagship software, HADDOCK and GROMACS. This workflow will allow using GROMACS both upstream and downstream of HADDOCK, either to sample input models for docking with HADDOCK or to refine, rescore or test the stability of the docked models generated by HADDOCK.

A 3-step plan to reach this objective was reported in Deliverable 1.3 and summarize below:

1) Finalization of the workflow prototype on VU side with an extension of its capabilities to handle HADDOCK's input/output requirements.

2) Testing phase during which the workflow will be used to process our assembled benchmark.

3) Releasing the workflow and publish it in parallel. We plan to write several tutorials that make use of the workflow and organize workshops under BioExcel branding.

As reported in Deliverable 2.4,[17] VU partners finalized and distributed a first version of the workflow, MDStudio, and made it available (https://www.research-software.nl/software/mdstudio). It involved, on the UU side, the development of a Python interface to HADDOCK via its XMLRPC API. This interface allows now any Python pipeline to remotely access HADDOCK methods exposed via its API. It is also aiming to extend usual features offered through the web server to tightly control the process.
Step 2 reported in Deliverable 1.3 was actually performed outside MDStudio to save time and tune MD parameters to be used later on. The first application on a dataset of 11 protein-protein complexes and 7 protein-peptide complexes show promising results to improve the discrimination of native-like from non-native like interfaces when the HADDOCK score is not discriminant. The most selective parameter turns out to be the persistence of residual contacts at the interface, while the energy terms derived from the sampling methods were not meaningful in the context of short simulations.
At last, we are considering running MDStudio on Microsoft Azure resources. We have been contacted by Wolfgang Gentzsch, from Ubercloud, who wanted to test a docker container optimized for MD simulations on Microsoft Azure. We wrote a proposal that has been approved by Microsoft and got granted 5000 core hours for a preliminary test. We just started benchmarking the different computing resources and hope to have some good insights into the performance of this setting by the end of Q4 2018.

4.4 Long-term development plans for CPMD QM/MM

BioExcel, together four other European partners, has developed a novel HPC efficient QM/MM interface (MiMiC) for the quantum CPMD code. While the Center of Excellence will continue to promote and advertise the usage of this QM/MM interface in its dissemination and training activities, for strategic reasons BioExcel has decided to focus its development effort in the QM/MM area towards a new quantum code, CP2K, by coupling it to GROMACS as MM code. In fact, even if currently CP2K does not reach the excellent strong scaling performance of CPMD,

[17] Submitted to the EC, but not yet accepted, so not yet published on Zenodo.

the CP2K code has been built in order to go beyond intrinsic limitations in the weak scaling of CPMD. Therefore, dealing with systems with a quantum part larger and larger, CP2K is supposed to outperform CPMD in future. Moreover, unlike CPMD, CP2K has a GPL-like license, as for GROMACS, and therefore the planned task to combine the two codes is supposed to be more straightforward than it has been for CPMD. On the other hand, the development of MiMiC will continue by the other European partners and in the following we report the already planned or foreseen directions for this novel QM/MM interface.

4.4.1 Coupling to other MM codes

MiMiC has been implemented with a high degree of abstraction, having in mind the possibility to interface CPMD also with other classical molecular dynamics (MM) codes by minimal code intervention. For this reason, a natural medium/long term plan is to establish collaborations with the developers of the other commonly used biological oriented MM codes (e.g. NAMD (http://www.ks.uiuc.edu/Research/namd), AMBER (http://ambermd.org) and Desmond (https://www.deshawresearch.com/resources_desmond.html) in order to develop in this codes the required layer that allows them to communicate with the interface and through that with CPMD, as we are currently doing together with the GROMACS development team. This work should be facilitated by the new BioExcel-initiated collaborations between the codes. In this way, we would provide a quite general QM/MM interface of CPMD that can work virtually with any classical force field. This agility is a key attribute for effective function of QM/MM at the Exascale, where users might have lots of special reasons (not least personal preferences) for using a specific QM or MM code.

4.4.2 Polarizable MM codes

Within the funded period, we are going to make the new QM/MM interface compatible with force fields adopting atomic multipole-based electrostatics with explicit dipole polarizability. It is therefore quite natural to think to extend the list of the supported MM codes to molecular modeling software that uses polarizable atomic multiple force fields like AMOEBA[18]. Short-term this could be provided by TINKER (https://dasher.wustl.edu/tinker/), and long-term it could interface to ports of GROMACS that exist, but are not currently on the roadmap for the core code.

4.4.3 Multiple time steps

Another long-term plan is to enable the QM/MM interface to work with the multiple time step approach. Multiple time-scale algorithms exploit the natural separation of time-scales in chemical systems to greatly accelerate the efficiency of molecular dynamics simulations. The usefulness of these methods in systems where the interactions are described by empirical potential is well known[19]. At the same time, recent advances in this direction have allowed overcoming the

[18] Y. Shi, Z. Xia, J. Zhang, R. Best, C. Wu, J. W. Ponder, P. Ren, *J. Chem. Theory Comput.* **9**(9), 4046 (2013)

[19] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, *Mol. Sim.*, **6**, 121 (1991)

difficulties associated with splitting the potential for a quantum system into fast and slowly varying components, bringing to multiple time step integrators in *ab initio* molecular dynamics[20]. This permits computational speedups of 4-5x compared to standard quantum *ab initio* schemes. Currently, there are no multiple time step algorithms implemented in CPMD. In addition, the extension of these approaches to the QM/MM framework is expected to enable one to get similar speedups but for much larger systems such as the biological ones. A useful collaboration with the GROMACS development of multiple time step integration algorithms may emerge in the future.

4.4.4 Time-dependent density functional theory

Time-dependent density functional theory (TDDFT) is a quantum mechanical approach employed to solve the time-independent nonrelativistic electronic Schrödinger equation (SE) and consequently to investigate the properties and dynamics of many-body systems in the presence of time-dependent potentials, such as electric or magnetic fields. The effect of such fields on biological systems can be studied with TDDFT to extract features like excitation energies, frequency-dependent response properties, and photoabsorption spectra. TDDFT is one of the less computational demanding techniques enabling the calculations of optical properties, and therefore it is one of the most widely used approach to investigate optical properties of biological systems. Of course, the algorithmic complexity of such problems still requires HPC computational resources to be solved. CPMD implements TDDFT. However, the new QM/MM interface currently does not allow extending the application of TDDFT to hybrid QM/MM resolutions. Therefore, another long-term plan of the developer team of the QM/MM interface is to enable the interface to support TDDFT calculations.

4.4.5 Machine learning models and towards Exascale

The above-mentioned numerical intrinsic complexity in finding experimentally accurate solutions to SE limits the possibility to perform routine electronic structure calculations and high throughput screening.

When Exascale resources become available, the combination of efficient QM/MM interfaces developed here, and parallel enhanced sampling techniques would give us the possibility to perform *ab initio* ligand screening, i.e. virtual screening based on accurate first-principle free energy calculations and not simply on predictions based on generic chemical properties from large libraries of compounds. Parallel enhanced sampling techniques that allow speeding up the free energy reconstruction by exploiting almost embarrassingly parallel schemes are already available, such as the multiple walker metadynamics method implemented in CPMD. However, the major bottleneck for reaching a form of “high throughput screening” based on QM/MM simulations is still the quantum part.

It has been shown that the task of repetitiously solving the SE can be mapped onto a computationally efficient, data-driven supervised machine learning (ML) problem instead[21]. In these models, expectation values of quantum-mechanical

[20] N. Luehr, T. E. Markland, T. J. Martinez, *J. Chem. Phys.* **140**, 084116 (2014)

[21] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld, O., *J. Chem. Theory Comput.* **11**, 2087–2096 (2015)

operators are inferred in the subset of chemical space spanned by a set of reference molecular graphs, enabling a speedup of several orders of magnitude for predicting relevant molecular properties such as enthalpies, polarizabilities, and electronic excitations[22]. QM reference calculations provide training examples. After training, accurate property predictions for new as of yet unseen molecules can be obtained at the base cost of the underlying ML model, provided that the new query molecule lies close to the space spanned by the reference data. So far, this technique has been only applied to small molecular systems treated fully quantum mechanically. On a medium/long term period, we plan to implement the required machinery in the interface in order to enable the ML-based QM/MM calculations for larger biologically relevant systems as well.

5 Concluding remarks

In this report, we have presented the views of BioExcel's software developers on the requirements they have from HPC and HTC hardware and middleware in order that they can target Exascale workflows. Expected trends in the attributes of hardware have been identified, and lessons drawn for the design and implementation of the BioExcel pilot codes. Those have been integrated into long-term development plans suited to the particular context, scale and complexity of the pilot codes. The plans have been updated to reflect progress in the codes, the hardware/software environment around them, and changed priorities.

[22] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K-R. Muller, A. Tkatchenko, *J. Phys. Chem. Lett.* **6**, 2326–2331 (2015)/ R. Ramakrishnan, M. Hartmann, E. Tapavicza, O. A. von Lilienfeld, *J. Chem. Phys.* **143**, 084111 (2015)