H2020 EINFRA-5-2015

www.bioexcel.eu

Project Number 675728

D1.5 – Final project release of pilot applications

WP1: Software Scalability & Usability

Document Information

Deliverable Number	D1.5
Deliverable Name	Final project release of all pilot applications
Due Date	2018-10-31 (PM36)
Deliverable Lead	KTH
Authors	Mark Abraham (KTH), Emiliano Ippoliti (JUELICH), Mikael Trellett (UU), Alexandre Bonvin (UU), Vytautas Gapsys (MPG), Bert de Groot (MPG), Stian Soiland-Reyes (UMAN)
Keywords	Software Release
WP	WP1
Nature	Report
Dissemination Level	Public
Final Version Date	2018-10-29
Reviewed by	Erwin Laure (KTH), Stian Soiland-Reyes (UNIMAN)
MGT Board Approval	2018-10-31

Document History

Partner	Date	Comments	Version
KTH	2018-10-8	First draft	0.1
MPG	2018-10-10	Update	0.2
UU	2018-10-15	Update	0.3
JUELICH	2018-10-16	Update	0.4
KTH	2018-10-16	Integration, zenodo links	0.5
KTH	2018-10-19	Reviewed	0.5
UNIMAN	2018-10-22	Reviewed	0.5.1
KTH	2018-10-24	Initial response to review comments	0.6
MPG	2019-10-26	Response to review comments	0.7
JUELICH	2019-10-27	Response to review comments	0.8
UU	2019-10-29	Response to review comments	0.9
KTH	2019-10-31	Final version	1.0

Executive Summary

This document describes the contents of the final project release of software developed for pilot applications in BioExcel. The efforts of Work Package 1 are spread over the entire project timeframe, and some collaborations, previous deployments and benchmarks have also used preliminary new code. However, to provide reference versions of all codes used in the project (e.g. to integrate them in BioExcel workflows and facilitate use by other BioExcel teams), the Work Package also makes these formal code releases. This also comes with records describing the current status of the software that results from the planning, development, documentation, and testing processes described in Deliverable 1.1 (10.5281/zenodo.263908) and the BioExcel white paper on scientific software engineering (10.5281/zenodo.1194634). The release itself takes the form of a package of Open Source-licensed source code, containing the newly developed task-parallel and throughput-oriented modules, and can be downloaded from the BioExcel webpage at http://bioexcel.eu/software/code-repositories, or from the permanent repository at (10.5281/zenodo.1473685). For most of the codes this is a BioExcel-specific repository to sync the releases with other project needs, while general users are usually recommended to follow the documentation, download and installation instructions available on the main application web pages. This release updates the previous project release reported in Deliverable 1.2 (10.5281/zenodo.574459).

Contents

1 Introduction

The BioExcel pilot codes have been selected based on their external impact and active larger user communities, and because of this there is continuous active development both inside and outside of BioExcel. However, to provide recognizable output artefacts for the BioExcel project, BioExcel also provides project releases of the source codes, of which this is the second and final for this phase of the project. These periodic releases will additionally enable stakeholders including both the European Commission, compute centers, and members of the bio-molecular simulation community to monitor the development going on in the project and provide valuable feedback to BioExcel. The first public release was described in D1.2 (10.5281/zenodo.574459), provided at project month 17, and approved by the project reviewers.

Work package 1 covers "Software Scalability and Usability", and the software efforts invested in BioExcel have been focused on providing added value in these areas, both by writing new code where necessary and by integrating, testing, maintaining and disseminating other contributions to the codes. Presently, the work covers three pilot codes:

- GROMACS, a high-performance task-parallel molecular dynamics software simulation and analysis suite. GROMACS is one of the most widely used biomolecular simulation codes in PRACE, and the BioExcel work concerns scaling and performance improvements, as well as implementing modern techniques for modularization, documentation, code review and QA testing.
- Related to GROMACS, BioExcel also supports the PMX project for automatically preparing molecular topologies for advanced free energy calculations, which is a particularly important target for using massive amounts of molecular simulations both in academia and industry.
- HADDOCK, an implementation of bio-molecular docking based on a wide range of biochemical information. This code has a very different usage profile, and it has been selected because if the importance of improving support for high-throughput life science calculations. HADDOCK is a key component of the many workflows developed in BioExcel. Its main mode of use is via its web portal.
- CPMD, a high-performance plane-wave/pseudopotential implementation of density functional theory suited for *ab initio* molecular dynamics. This represents our third type of application work, where the BioExcel pilot efforts are focused on implementing libraries to make it possible to integrate different types of simulation codes for multiscale modeling.

In addition to software development focused directly on the applications themselves, another important part of the usability improvements in BioExcel comes from integrating them into workflows as presented in work package 2.

The BioExcel codes have very different degrees of parallelization maturity. They were selected to cover a range of proven-impact applications that face very different challenges when preparing for Exascale, but also to make sure the BioExcel project develops competence covering the entire scale of biomolecular

modeling used in academia and industry today, ranging from QM/MM models on the small atomic scale to simulations of very large systems and rapid integrative modeling. The modifications implemented in the pilot codes, interoperability libraries, and related workflows will facilitate using combinations of codes to solve application problems.

The contents of this release reflect the current status of the major ongoing effort of this work package on the pilot codes, covering the period subsequent to Deliverable 1.2 (10.5281/zenodo.574459) and preceding the end of the first phase of the BioExcel project. This naturally encompasses the results of end-user consultation, team planning, software development, documentation, quality assurance, testing, and bug fixing that is expected for the output of a software e-infrastructure like BioExcel. The software development procedures were documented in Deliverable 1.1 (10.5281/zenodo.263908) and the BioExcel white paper on scientific software design (10.5281/zenodo.1194634).

The project release is available from http://bioexcel.eu/software/code-repositories as a packaged source code archive for each of the three codes, as well as a convenience aggregate package (ALL_CODES). The documentation in this package provides references to the main repositories for the respective codes. The codes also provide extensive support and documentation, either through their existing web pages or fora such as the Bioexcel Discourse server at http://ask.bioexcel.eu/.

Although these BioExcel releases capture the improvements from BioExcel's effort, those changes have also been contributed back to each upstream project's code repository. Formally each pilot code's community manages their own formal releases separately and at different intervals. This is in line with the expectations of their existing user bases and different needs for release cycles, as the codes are developed by particular research groups and institutions supported by multiple streams of funding. However, for some of the codes (GROMACS, HADDOCK) BioExcel staff are co-responsible for the main management and steering of the software development.

The valuable support of BioExcel is acknowledged by each upstream project, and as detailed in Deliverable 1.1 (10.5281/zenodo.263908) and the BioExcel white paper on scientific software development (10.5281/zenodo.1194634), we have worked with them to harmonize software development practices, as well as clarify software licenses where appropriate. BioExcel has also had major impact on teaching and tutorials, including both user- and developer-focused activities to improve software quality. Valuable feedback and input into the requirements gathering process are also collected via BioExcel and thus the CoE helps defining future priorities for the individual developer communities.

BioExcel is committed to business-friendly open source licensing for all code developed in the project, but to ensure impact for the most widely used application codes we also need to ensure the code can be used with existing code bases and their licenses. This is handled by making all BioExcel code available as independent libraries or scripts, which can easily be adapted to other codes too

when more liberal licenses are available. The licenses used in BioExcel are the Lesser GNU General Public License (LGPL) v2.1 (GROMACS and CPMD QM/MM), LGPL v3 (pmx) and the Apache License 2.0 (HADDOCK scripts, as well as WP2 workflows). These are all OSI-approved and well-recognized Open Source licenses that grant the right to run the code for any purpose, to link it into closed commercial applications, as well as to modify and/or redistribute the code. The licenses differ mainly in minor details such as restrictions on derived works or protections against software patent infringements.

2 GROMACS

In BioExcel, GROMACS represents an advanced explicitly parallel code with multiple layers of parallelism (MPI, Pthreads/OpenMP, SIMD, CUDA/OpenCL). While the code can scale extremely well for sufficiently large systems, the BioExcel efforts are used to improve performance and scaling for typical medium-size systems used in scientific applications. BioExcel has also made it possible to invest significant efforts into introducing modern professional code development, testing practices, and better integration and testing of contributions from external users.

2.1 Software achievements in GROMACS

Over the last four years, the GROMACS project has moved from ad hoc feature-based releases that occasionally were up to a year late to enforcing a new policy of making annual source-code releases where features need to meet release deadlines. This follows a procedure starting around nine months before a planned release date, using internal alpha and public beta testing that leads to a time-based public release. The current release schedule made the first beta release of the 2019 version on 22 October 2018, to get feedback from the user community and begin performance testing. Further beta releases and release candidates will follow, and the public release will take place in very early January 2019, as suits a version labelled 2019.

To discourage users from mistakenly running program versions installed several years ago, GROMACS has changed the versioning scheme so that each new major version number indicates the year of the new stable release branch (e.g. 2018). Each such branch is actively supported for two years by incrementing the minor version number – at any point in time there are thus both “stable” and “very stable” release branches available for users, as shown in the timeline seen in Figure 1. The development branch is also publicly available and guaranteed to pass all existing unit and end-to-end tests.

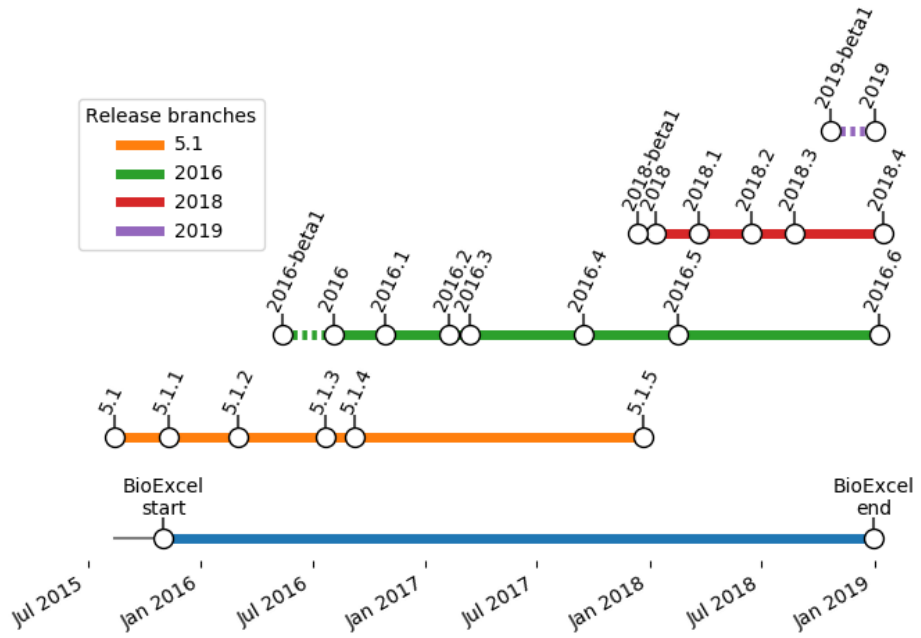

Figure 1 GROMACS releases from the branches under maintenance over the lifetime of BioExcel. When the 2018 branch was released, the 5.1 branch was retired from active support. Around when 2019 is released in January 2019, the 2016 branch will be retired, perhaps after a final update.

The main GROMACS project is managed by staff supported by BioExcel. The first major release that incorporated source code supported by BioExcel took place already in 2016, and a subsequent major release took place in 2018. Every 2-3 months a routine bug-fix update is made, so in addition to the *2016* major release, official versions *2016.1* through *2016.5* and *2018* through *2018.*3 have all incorporated major contributions from developers supported by BioExcel, as well as that from numerous other members of the global GROMACS development team. These releases have been advertised through the BioExcel channels, including the newsletter, webpage, and Twitter feed, as well as the traditional GROMACS mailing list distribution.

The final BioExcel project release version was coordinated with the GROMACS internal deadlines for the beta for GROMACS 2019 (released to the public on 22 October 2018), so benchmarks and workflow integration work will be done with a version that corresponds directly to a normal public stable release branch version. Considerable amounts of BioExcel-supported work maintained the *2016* and *2018* releases, and some of this is currently awaiting review in the GROMACS code-review system (https://gerrit.gromacs.org). Following long-standing practice in the GROMACS project, all code is released under the *Lesser GNU General Public License, version 2.1*, which permits unrestricted (including commercial) linking to the library as well as free or commercial modification or redistribution.

Over the lifetime of BioExcel, new task-parallel modules for *GROMACS* have been designed and released, substantially improving performance while providing a simple user interface. Users can now automatically offload the long-ranged PME task to a GPU, providing improved parallelism with remaining CPU tasks. This

permits users to buy hardware with cheaper CPUs, or add more GPUs to existing nodes, while retaining or improving performance.

In addition to scaling and performance advances, much of the BioExcel work has focused on improving usability and sustainability of the code by providing better user and developer documentation as well as testing. The formal release process has been extended with support for automated construction and testing of both the source-code tar package and documentation. Documentation for the latest version is available at http://manual.gromacs.org/documentation/.

In particular, to avoid the constant issues of outdated documentation, all documentation is now generated automatically from in-code source. This documentation takes the form of terminal output returned by e.g. "*gmx help select*" or "*gmx select -h*", which has content identical to "*man gmx-select*" and to that found on the version-specific webpages similarly generated automatically (e.g. see http://manual.gromacs.org/documentation/2016.3/onlinehelp/gmx-select.html). A Sphinx-based user guide, install guide, reference manual and release notes are also built based on the source code. In 2019, thanks to effort supported by BioExcel, all content will be available as cross-linked HTML and also as more than 600 pages in PDF for offline access by user and developers (available at http://manual.gromacs.org/documentation/2019-beta1). All these can be found online at http://manual.gromacs.org/documentation/ and is also included within the BioExcel project release of GROMACS. BioExcel has also made it possible to provide extensive installation documentation targeting both novice users and experts at compute centers, covering e.g. compiler and MPI library recommendations and specific instructions to create builds for accelerator hardware such as CUDA, OpenCL or Xeon Phi.

The formal release notes mentioned above are focused on things clearly of interest to users, including features added, deprecated and removed, along with bugs fixed and improvements to documentation and portability. However, BioExcel has also enabled a great deal of work not visible to users in these notes, including

- Development and maintenance of an automated infrastructure for code review, dissemination and testing,
- expanding the unit test coverage,
- extensive deployment of memory and thread sanitization builds as part of continuous integration to catch bugs
- usability testing of alpha/beta releases,
- organization of developer meetings and hackathons,
- adding developer documentation, and
- refactoring code for maintainability and future extension.

Notable achievements in the *2016* release series were reported in Deliverable 1.1 (10.5281/zenodo.263908). Those in the *2018* release (as of *2016.5*) relevant to the BioExcel objectives include:

- PME long-ranged interactions (critical for accurate modeling of electrostatics in the typical case of heterogenous simulation systems like membrane proteins) can now run on a single GPU, which means many fewer CPU cores are needed for good performance.

- Optimized SIMD support for recent CPU architectures likely to feature on the exascale landscape: AMD Zen, Intel Skylake-X and Skylake Xeon-SP, as well as maintenance and updates to ARM and POWER SIMD support.
- The AWH (Accelerated Weight Histogram) method is now supported, which is an exascale-suitable adaptive biasing method used for overcoming free energy barriers and calculating free energies (10.1063/1.4890371).
- A new dual-list dynamic-pruning algorithm for the short-ranged interactions, that uses an inner and outer list to permit a longer-lived outer list, while doing less work overall and making run performance less sensitive to parameter choices, which is a key usability enhancement.
- A physical validation suite is added, which runs a series of short simulations, to verify the expected statistical properties, e.g. of energy distributions between the simulations, as a sensitive test that the code correctly samples the expected ensemble.
- Conserved quantities are computed and reported for more integration schemes - now including all Berendsen and Parrinello-Rahman schemes.
- Key update, angle, and PME kernels received SIMD implementations.
- Improving performance and scaling for numerous simulation kernels running on GPUs (both NVIDIA and AMD) and CPUs (see Figure 2).
- Improvements of the continuous-integration testing infrastructure to guarantee that every commit in the development branch passes all unit tests on all major architectures.
- Expanding unit test coverage of source code, and integration with Debian and Fedora software testing projects to identify unit tests failing on rare hardware/compiler combinations.
- Making many minor enhancements and fixes to setup and analysis tools
- Making many fixes for corner cases of portability, particularly with little-used build configurations.
- Removing minor features no longer supported

Functionality changes coming in the upcoming 2019 release are:

- Updated implementation of domain decomposition based on “update groups” which complements recent improvements to force-calculation kernels by decomposing the whole computation into domains that suit both the force calculation and the update and constraint kernels
- Through co-design with Intel, added support for non-bonded interactions using OpenCL on Intel integrated GPUs, as a path to supporting a possible future device in the exascale era
- Added support for PME long-ranged calculations to OpenCL (currently on AMD and NVIDIA devices)
- Integrating support for the second version of the hardware locality library (hwloc) to take the layout of hardware threads, cores, sockets, NUMA domains and PCIe bus locations into account to automatically optimize parallelization.
- Early stages of the gmxapi Python and C++ API for GROMACS simulations, which will be key infrastructure for resilient, performant, exascale workflows using the GROMACS simulation back end.

- Improved usability and functionality for umbrella sampling free-energy simulations.
- Usability and correctness improvements to system preparation and analysis tools.
- Removing minor features no longer supported.

In particular, BioExcel funded the majority of the work on new CPU and GPU acceleration, parallelism, and virtually all work on improving sustainability through unit tests, documentation, and continuous integration. For the remaining features, BioExcel staff has been responsible for coordinating development and doing code review and testing. The present release will be used for the next phase of BioExcel work on workflow integration, training, and use cases involving large-scale free energy calculation, while the efforts in work package 1 will focus on further improving scaling and usability.

Highlight: In GROMACS *2018*, PME long-range electrostatic interactions were ported to run on NVIDIA GPUs. This is illustrated in Figure 2, showing dramatic performance improvements from GROMACS 5.0 to 2018. Further noticeable improvements are expected for GROMACS 2019, particularly from the use of update groups, although benchmarking efforts have only recently begun at the time of delivery of this report.

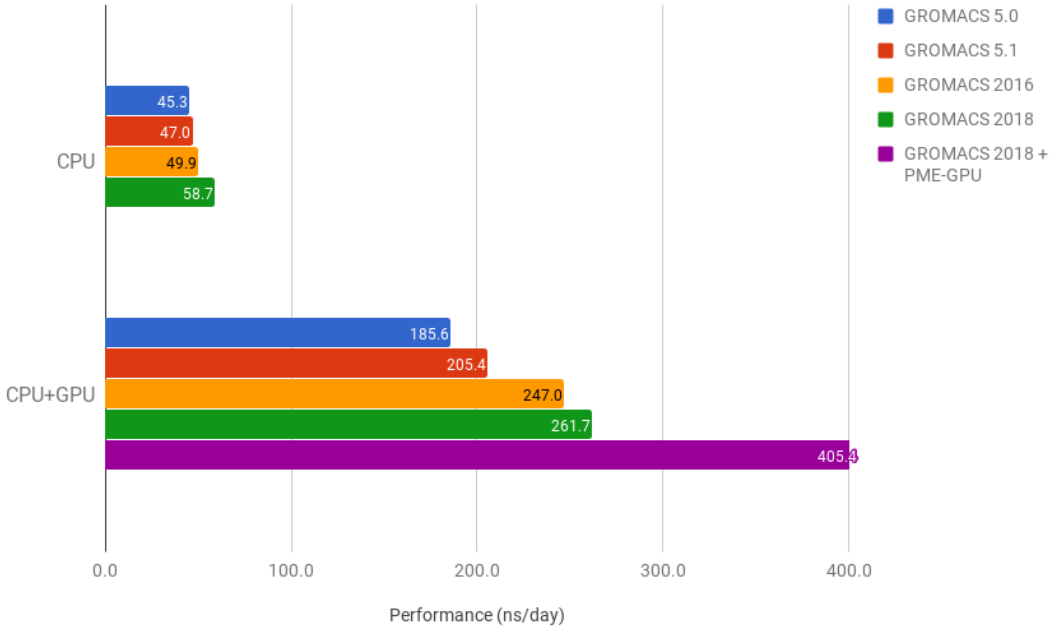

Figure 2 Performance improvements for CPU-only and CPU+GPU runs on a typical biomolecular simulation system, over the period from GROMACS 5.0 to GROMACS 2018, covering the full BioExcel project lifetime. The dramatic improvement from porting PME to GPUs is clearly seen, although other improvements also contributed. Benchmarks were run on the same single node containing 6-core Intel E5-2620v4 CPU and an NVIDIA GTX 1080Ti GPU, which is a combination believed to model the capability mix that might be found at the exascale.

The **pmx** package is available as an add-on set of scripts for GROMACS and recently also made available from a web interface at http://pmx.mpibpc.mpg.de/.

The webserver was created and is maintained in-house at MPG. It is hosted by MPG as well. The source code is available on request. The webserver is updated together with the updates in pmx. The web support has been added with BioExcel effort, as acknowledged within the interface. This makes pmx functionality for carrying out mutation free energy calculations accessible to all GROMACS users, without need for additional software, hence taking one step further in the BioExcel goal of making biomolecular simulations accessible to non-developers. In addition, support for nucleic acid mutations has been recently added, as well as support for arbitrary ligands, taking an important step towards free-energy supported lead compound optimization.

The pmx project is maintained at the third-party repository https://github.com/deGrootLab/pmxhttps://github.com/dseeliger/pmx. We have included a verified snapshot from that repository within this deliverable. Up to now, pmx was following the strategy of having a stable master branch with regular updates. The development was done in separate branches. In the future, pmx plans to adopt the approach followed by GROMACS and have yearly major releases.

2.2 Strategies for usage of extreme-scale resources with GROMACS

As reported in Deliverable 1.3 (10.5281/zenodo.574605), ensemble-level parallelism is a key part of the extreme-scale strategy for molecular dynamics simulations, because the problem sizes of interest do not have enough arithmetic work to justify using more than a handful of exascale-era compute nodes for a single trajectory, unless dramatic unforeseen improvements in data-transfer latencies become routinely available. Accordingly, the use of ensemble-based workflows (perhaps leveraging technologies such as PMX for providing topologies for multiple simultaneous trajectories on free-energy surfaces), is the encouraged strategy. Newly implemented features in GROMACS, such as the AWH enhanced-sampling algorithms, leverage the use of multiple trajectory walkers, and will form a key part of simulation strategies used at the exascale. Illustrative large-scale calculations with AWH in GROMACS are underway to present in the final report at the conclusion of the project.

Free energy calculations that aim for a high throughput mutational scans present a particularly suited target for massive parallelization. Previously, pmx based automation of the simulation setup procedures allowed performing large scale amino acid as well as nucleic acid mutation scans in proteins and protein-DNA complexes. With the latest pmx developments it is now possible to perform ligand modifications in an automated manner. This opens a possibility to run high throughput parallel screens for lead identification and drug optimization. Each free energy calculation in such a setup can be executed independently enabling the use of ensemble-based parallelization workflows.

Early stages of the integration of the NIH-supported gmxapi work, providing programmatic Python and C++ APIS has taken place. This key infrastructure work has been supported by the modularization and modernization efforts undertaking

within BioExcel. Other adaptive ensemble-based workflows will be implemented using these APIs in the future, including new efforts for simulations guided by experimental data, e.g. available from cryo-electron microscopy data facilitated by RELION. These will be complemented by the ongoing efforts of Work Package 2 of BioExcel, which will put into the hands of biomolecular simulation scientists tools that can address end-to-end use cases. These will need to leverage whichever of bioinformatics tools, molecular docking, and molecular dynamics simulations address the scientific problems at hand.

3 HADDOCK

The BioExcel work on HADDOCK is focused on improving HPC performance and usability of the throughput-focused research in life science that traditionally has found it very difficult to exploit HPC. This type of work is highly dependent on advanced automation and workflows, and some of the key BioExcel challenges for HADDOCK are related to enabling the code to be used efficiently in workflows and diverse HPC/Cloud environments. The main and most commonly used mode of access to HADDOCK remains direct submission to its web portal, which exposes a XMLRPC API. BioExcel-supported work on HADDOCK has led to the development, maintenance and release of numerous pre- and post-processing scripts for setting up and analyzing its docking runs. These are currently found at https://github.com/haddocking/haddock-tools, and https://github.com/haddocking/pdb-tools in addition to the BioExcel release repository.

3.1 Software achievements in HADDOCK

We released v1.2.0 of our pdb-tools which includes BioExcel-related additions (690f2b5). This is the third stable release. pdb-tools contains a variety of python utilities for the processing of PDB files which should be valuable to a large variety of applications. Their use is in particular described in various HADDOCK online tutorial (see Deliverable 4.5 section 2.7.1, 10.5281/zenodo.574620).

The recent pdb-tools release adds a new script, pdb_tidy.pdb, that adds missing END/TER statement to PDB files when residue IDs and/or chain IDs breaks are detected. List of all the changes linked to their commits is available on the release page (https://github.com/haddocking/pdb-tools/releases/tag/1.2.0).

Since 28th of September, our pdb-tools scripts are also part of the SBGrid supported applications and accessible through their different installation interfaces (https://sbgrid.org/software/titles/pdb-tools). The SBGrid consortium gathers about 340 members (most being teams and laboratories like the BonvinLab that is part of BioExcel) dispatched in 114 institutions and it aims to centralize and harmonize the usage of tools (~410) dedicated to structural biology. To do so, they have created a large library of applications together with tools to automate their installation on any type of resources, from laptops to HPC clusters for the labs subscribing to SBGrid. The consortium also organizes webinars around the tools present in the library and offer dedicated support to

their members. HADDOCK and DISVIS are two software developed by the Utrecht partner that are included in the SBGrid software distribution. The pdb-tools is a recent addition. As of today, only version 1.1.0 is available on SBGRid but version 1.2.0 (current stable version) will be added soon.

The second release of the haddock-tools is v2.0.0. The haddock-tools includes variety of script utilities for the setup of HADDOCK calculations. They concentrate on PDB files manipulations and restraints generation and validation. This new release adds three new scripts that wrap up daily routines performed by most of HADDOCK users. Several modifications and bug fixes have been made as well and a full list of the changes can be found on the release page (https://github.com/haddocking/haddock-tools/releases/tag/2.0.0). The scripts include functionality for:

- Defining interaction restraints to be used during a run of HADDOCK
- Validating restraints before launching a HADDOCK run
- Finding inter-chain or inter-segment contacts
- Wrapping the molprobity utility, for predicting histidine residue protonation states
- Cleaning structure descriptions in PDB files
- Mutating one residue to another in a PDB file
- Extracting residue sequences from PDB files
- Extracting HADDOCK scoring terms from HADDOCK models
- Get list of passive residues from a list of active ones
- Convert *.web HADDOCK parameter files into JSON to facilitate their parsing/manipulation

These scripts are released under the Apache License 2.0, which permits free use by all.

HADDOCK is intended to be used via a web interface found at http://milou.science.uu.nl/services/HADDOCK2.2/, which is free to use for non-profit users upon registration. This is the most-used access mode to HADDOCK with just under 12.000 registered users to date, but it is possible for users to obtain a license for using the code locally too. The web portal however performs several pre- and post-processing steps that are not available in the local install. The current official release of the local code is version 2.2, which was defined as background in BioExcel and shared with all partners.

HADDOCK has a critical dependency upon the third-party software CNS (http://cns-online.org/v1.3/) for the core computational work, which effectively restricts HADDOCK to those licensed to use CNS (free for non-profit). As part of WP1 activities and as reported in D1.2, BioExcel has explored the feasibility of working towards relaxing this restriction, e.g. by replacing CNS functionality with GROMACS and this led to a negative conclusion with regards to the current PM dedicated for such task.

We have released an improved Python interface to HADDOCK via its XMLRPC API. This interface allows 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. This interface facilitates the integration of HADDOCK into workflows and is currently used as main endpoint within MDStudio workflow (see UC4 – Molecular Recognition).

As announced in D1.2 and as part of WP1 effort, a new version of HADDOCK web portal and software is now in beta test and available to a limited number of user on a per-demand basis.

The new software version associated with the new web portal contains several new features, among others:

- support for cryo-electron microscopy maps as restraints to drive the docking
- Addition of a z-restraining potential as implicit membrane potential
- implementation of Martini coarse-grain models for both protein and nucleic acids
- support of new modified amino-acids (the full list is accessible from: http://milou.science.uu.nl/services/HADDOCK2.2/library.html).
- options to turn off, or limit the post-processing analysis to clustering only (required for moving to exascale modelling of interactomes)
- rebuilding of missing-side chains in the context of the complex for the refinement protocol.

The beta-version of the HADDOCK2.4 web portal, still under development, is accessible from: https://csbdevel.science.uu.nl/haddock2.4/ We have recently connected it to our centralized new user registration system compliant with the EU GDPR.

Together with the implementation of new features at the software level, we also redesigned and rewrote from scratch the web portal using the Flask framework. This new web portal aims at:

(1) facilitating any new future development and implementation of new input features,
(2) improving the user experience with more interactivity and feedback along the submission process
(3) improving the user experience with more interactivity in the results analysis (interactive plots based on bokeh library)
(4) shortening the full submission pipeline by gathering the pre-processing and validation steps at the same level
(5) providing direct feed-back about problems during the interactive submission process.

Submission is now divided into 3 dependent steps:

1. Input data – PDB files of the different partners together with some related parameters (N-ter and C-ter charged, Chain IDs, etc.)
2. Input parameters – Active/passive residues, Histidine protonation states or cryo-EM restraints can be input at this stage.
3. Docking parameters – All HADDOCK parameters exposed to the users.

Each step takes information of what has been provided in the previous step to adapt its content. For instance, the second step displays an interactive sequence viewer to select important residues for the docking and highlights the secondary structure of the molecule used as input (see Figure 3).

Some efforts have been also made in the results page to replace the previous static plot images of different HADDOCK terms compared to each other to new interactive ones. Those news plots allow for some deeper exploration of HADDOCK scoring terms per model and kick-off the baseline for some other interactive representations of the huge amount of data generated during each docking process.

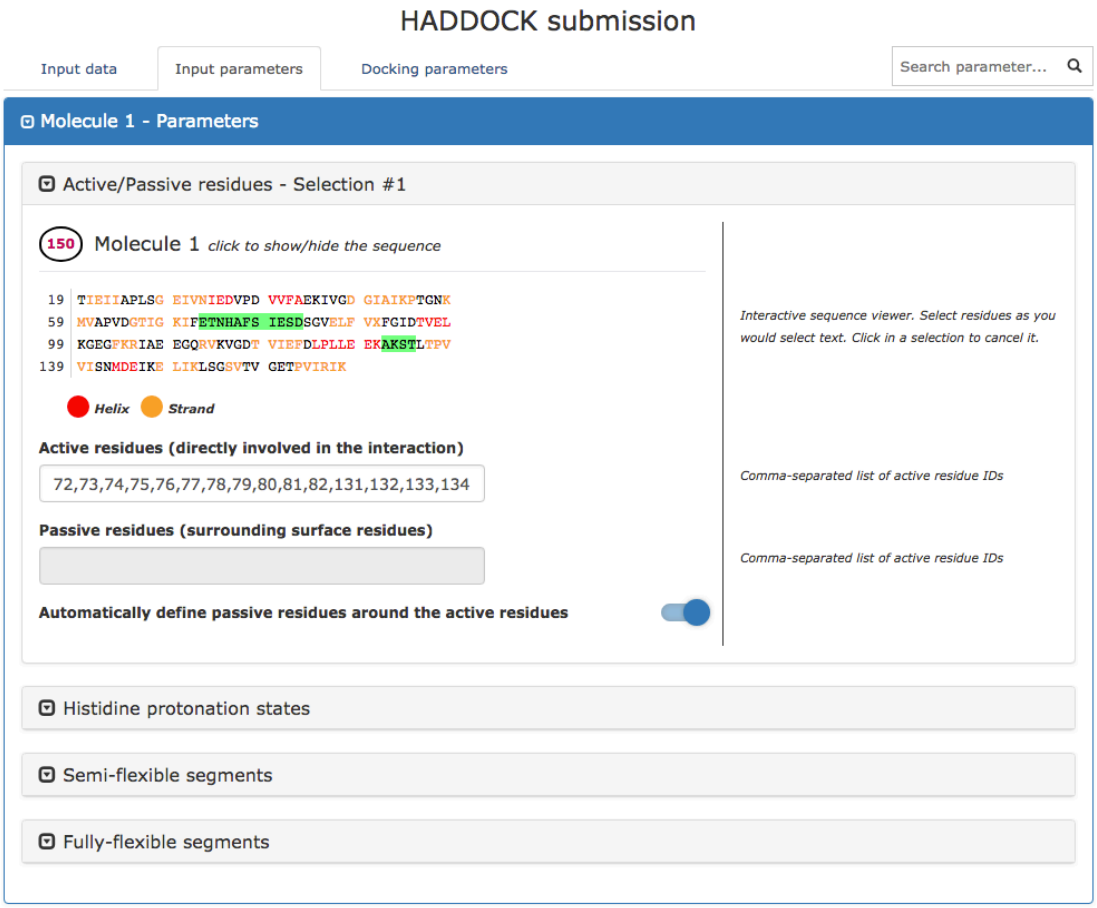

Figure 3 Input restraints interface highlighting the interactive sequence viewer module. Sequence and secondary structure information come from the pre-processing of input PDB files provided in the 1st step.

This new version was developed in parallel of the new CSB user portal (https://nestor.science.uu.nl) and those two are now bridged under the same framework. This allows users to monitor, access and take actions regarding their HADDOCK jobs via their profile page (see Figure 4).

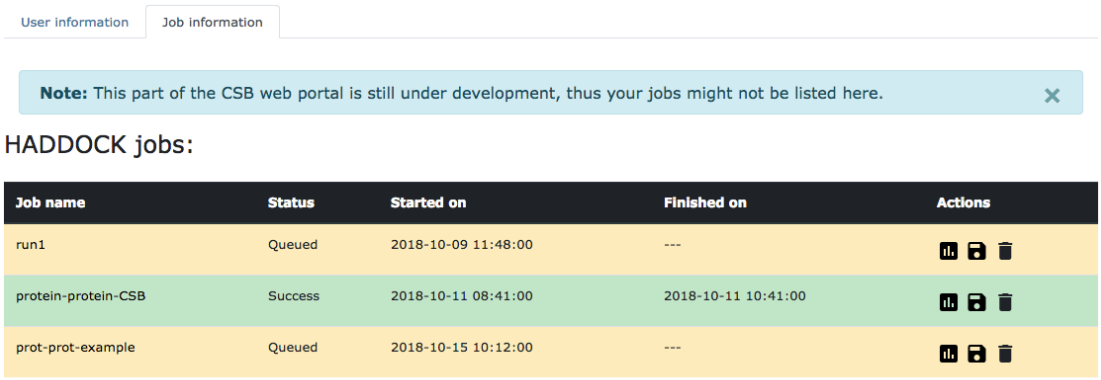

Figure 4 Extract of the new user profile page where lists of his current jobs can be seen. Several actions are made available to access/save/delete jobs.

The different modules deployed to provide the HADDOCK2.4 portal and the user management portal have been containerized with docker and a significant effort has been made to automate the deployment process. This allows us to now deploy the complete front-end (see Figure 5) on any new resources (dedicated or cloud) within an hour. Some automation of the HADDOCK software installation still needs to be performed to really automate the full deployment pipeline but preliminary tests on cloud resources made available through the Helix Nebula project showed great success.

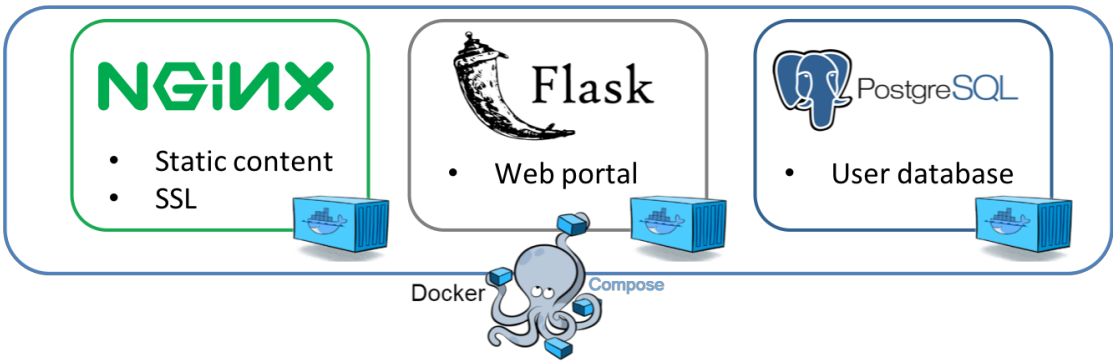

Figure 5 Schematic representation of the new docker images interconnection and orchestration to run CSB web portals front-end.

Despite the fact that we have a beta version of the new HADDOCK2.4 portal currently used for testing, the code is still evolving. Once we will have a stable version (which is planned before the end of BioExcel) we will make the portal code public on GitHub (https://github.com/haddocking/haddock-webserver-flask). This will be probably licensed under the Apache 2.0 license unless some other license is found more suitable for potential commercial activities planned in BioExcel2.

3.2 Strategies for usage of extreme-scale resources with HADDOCK

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 as 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 in sequential analysis steps.

4 QM/MM in CPMD

MiMiC is the HPC-oriented QM/MM interface for CPMD supported by BioExcel. The BioExcel team has built up a network of collaborations to implement this interface. This deliverable contains both the last release of the CPMD-MiMiC communication library and the MiMiC patches for GROMACS and CPMD codes, developed by JUELICH under the auspices of BioExcel. The communication library is the part of the MiMiC modular infrastructure is designed to facilitate communication between arbitrary QM and MM packages. The intent is to establish the data interaction between the QM and MM packages and transfer all the needed data (coordinates, forces, constraints, etc). This is the fundamental component of MiMiC QM/MM interface that enables the multiple program multiple data (MPMD) approach adopted for QM/MM interface. The library is implemented in C++11 based on the MPI 2.0 functionality. The library routines are provided behind a C API so that a stable application binary interface (ABI) is possible and to facilitate calling from FORTRAN, C and C++ programs. This flexibility is intended to permit different pairs of codes to use a common infrastructure, which offers the best prospects for ongoing access to these kinds of QM/MM biomolecular simulations.

A consortium made up of non-BioExcel collaborators has contributed to the development, including Prof. U. Rothlisberger (EPFL, Lausanne), Dr. T. Laino and Dr. V. Weber (IBM Research Zurich), Viacheslav Bolnykh (RWTH Aachen University & Cyprus Institute), Dr. J. M. Haugaard Olsen (The Artic University of Norway, Tromsø) and Dr. Simone Meloni (University “La Sapienza”, Rome). It is planned to make a formal open release of the MiMiC communication library and the necessary patches to CPMD upon publication of a scientific paper from this consortium. The patches for GROMACS allow it to interface with the MiMiC communication library and are already directly implemented in GROMACS and will be available in the 2019 release.

To install MiMiC and run a QM/MM simulation by employing CPMD and GROMACS, also CPMD needs to be modified. Thanks to the collaboration with IBM that is the owner of the CPMD code, all the required modifications will be directly available in the next release of CPMD, which IBM estimates for 2019. Meanwhile, the user can employ the patches available in this deliverable in order to add the required changes to the current CPMD 4.1 release, together with the MiMiC communication library.

4.1 Software achievements for QM/MM

MiMiC was designed to provide a code-agnostic solution in order to achieve high flexibility and modularity of the new framework. It is built into CPMD as an additional module implementing the additive QM/MM scheme described in [10.1063/1.1462041]. It uses a loose coupling scheme within which both CPMD and GROMACS are running independently, communicating in the beginning and in the end of each time step.

Benchmarks performed on model systems has shown that MiMiC reach much higher scaling efficiency compared to the previous popular implementation of QM/MM in CPMD based on tight coupling to GROMOS96 code (see Figure 6).

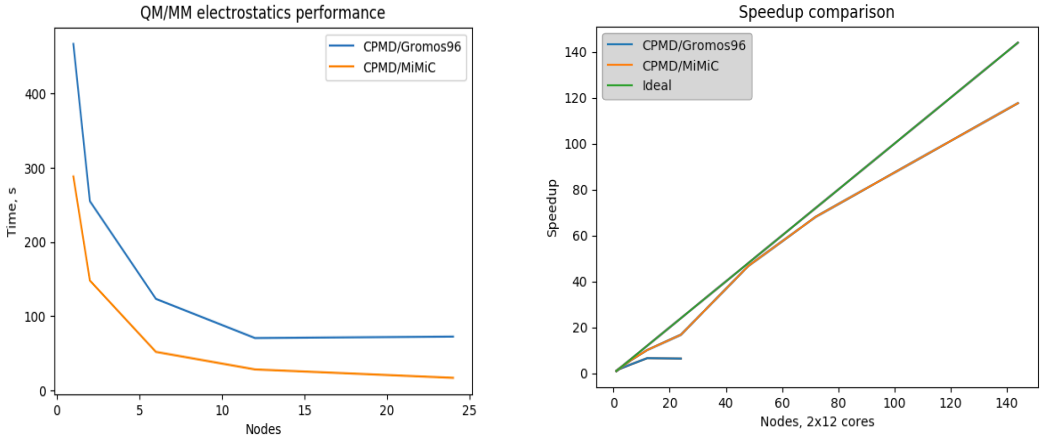

Figure 6 Left: absolute timings of QM/MM electrostatic computation with respect to the number of nodes employed. Right: scaling performance in a QM/MM molecular dynamics simulation for a protein system in solution (50,000 atoms) using the BLYP exchange-correlation functional.

Such high scalability have been achieved by using a multi-level parallelization scheme. This scheme take advantage of the inherent CPMD parallel infrastructure as described in [10.1109/IPDPS.2014.81]. In particular, the highest level belonging to the so called *task groups* has been used to distribute the atoms of the MM description between processes belonging to different groups. This allows the QM/MM algorithm to approach the excellent scaling performance characteristic to CPMD. The middle layer of parallelization uses the slab decomposition of the QM electronic sub-system across MPI tasks within the task group in order to speed up the computation of the QM/MM electrostatic. Finally, on the lowest level, threading with OpenMP is used to further distribute the calculation of atomic contributions to energy, forces, and potential across threads. Having implemented such a multi-layer parallelization scheme, we have achieved ~10x improvement in scaling compared to the previous QM/MM implementation. Moreover, the additional strategy aimed at preserving cache coherence and vectorization support allowed us to improve the performance of the most computationally expensive routines by a factor of two. Finally, a new efficient constraint solver has been implemented based on SHAKE [10.1016/j.jcp.2005.03.015], which eliminates

the largest bottleneck in the CPMD integration algorithm that would have prevented to reach very high scaling efficiency in QM/MM molecular dynamics simulations using MiMiC.

To sum up, MiMiC has been successfully coupled to both GROMACS and CPMD and it is going to be supported by the upcoming releases of the programs. The new interface shows high scaling performance, enabling the application for PRACE computational resources grants. The initial validation has been performed by the development team and the internal user testing has begun within the collaboration network.

The MiMiC communication library and patch to CPMD will be released under the conditions already described, and available openly using the Lesser GNU General Public license, version 2.1, the same as GROMACS.

4.2 Strategies for usage of extreme-scale resources with QM/MM

The preliminary benchmarks reported in the previous section show that not only MiMiC outperforms any previous QM/MM approach based on CPMD, but above all it is able to reach really HPC scaling performance: when employed on a state-of-the-art biological system (see Figure 6, Right), MiMiC was able to efficiently scale over more than 140 nodes (i.e. 3,360 cores) without seeming to approach its limit: only practical issues with the testing cluster prevented us to test the code over a larger set of nodes. CPMD in a full QM configuration is well known to have very good parallel efficiency. MiMiC has been able to extend its excellent scaling performance to QM/MM simulation regime.

In addition, as for GROMACS, the ensemble parallelism can further extend the scaling capabilities of this code. In fact, enhanced sampling techniques based on system replications with almost independent copies are very efficient methods to speed up the sampling of QM/MM molecular dynamics simulations as well. Therefore, employing MiMiC in replica-based free-energy simulations can be a promising candidate of biological-relevant simulations that might have impact at extreme scale.

Moreover, a task-based parallelism similar to the one of GROMACS can be implemented in QM and QM/MM codes as within a single step they need to compute several contributions to the total potential that are, in principle, independent. However, in case of a QM code the number of these contributions is not constant and may vary, depending on the simulation setup. Therefore, it requires finer control to achieve better load distribution. On the other hand, the computational load of each processor within such simulation keeps approximately constant, as the domain size remains the same. Thanks to that, a simple tool tuning such a load distribution can be implemented. This tool will need to be called before the simulation begins in order to find the best load distribution among processes. Using this approach one can setup a QM/MM simulation that would step even closer to the exascale-level performance.

5 Contributions to other software development

In future phases, BioExcel plans to extend its capacity and sustainability as an e-infrastructure, by adding support for more codes. It is already preparing to do so by making contributions to other free and open-source software of interest to biomolecular simulation scientists. In particular, we have provided assistance to

- RELION-GPU (http://www2.mrc-lmb.cam.ac.uk/relion). This is a new GPU-accelerated parallel version of the most popular program to do Bayesian-based refinement of 3D reconstructions of cryo-EM densities. This work was enabled by extensive knowledge transfer from the BioExcel team, combined with two developers funded by Stockholm university. This involved a wide range of software development methodologies, including coding for GPUs, multi-threading, testing and build systems. Since fall 2016, RELION-GPU has become the dominant code to perform 3D reconstruction in cryo-EM, and the scientific paper including BioExcel researchers is cited multiple times per week.
- DisVis (https://github.com/haddocking/disvis), a tool to visualize and quantify the accessible interaction space of restrained biomolecular complexes, by providing a support forum on the BioExcel Discourse server,
- PowerFit (https://github.com/haddocking/powerfit), a program to fit high-resolution atomic structures in cryo-EM densities, by providing a support form on the BioExcel Discourse server, and

6 Concluding remarks

The pilot codes have made the final project release of the software developed with the support of BioExcel, including new task-parallelism and throughput modules. This release updates the previous project release reported in Deliverable 1.2 (10.5281/zenodo.574459). All new modules have been released under free and open business-friendly licensing, as planned. Valued contributions have been made to other related biomolecular simulation software, some with very high impact. For HADDOCK, GROMACS, and pmx, ongoing software development will continue as part of BioExcel-2, to support the use cases described in that follow-up project. Because all the codes in BioExcel are in widespread use in the scientific simulation community, there are non-BioExcel efforts ongoing on all of the BioExcel WP1 software projects, so the value delivered by WP1 will continue to have impact on users well after the project concludes. Note that Deliverable 1.3 (10.5281/zenodo.574605) already presented the long term roadmap of the pilot codes, which will be updated in the final Deliverable 1.4. Strategies for extreme-scale usage of the BioExcel pilot codes have been described, although work here is still ongoing as computing resources at these scales are still being designed and deployed.