H2020 EINFRA-5-2015

bioexcel

Center of Excellence for Computational Biomolecular Research

www.bioexcel.eu

Project Number 675728

D2.1 – State of the art and gap analysis

WP2: Portable Environments for Computing and Data Resources

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	D2.1
Deliverable Name	State of the art and gap analysis
Due Date	2016-03-30 (PM5)
Deliverable Lead	IRB
Authors	Adam Hospital (IRB), Anna Montras (IRB), Stian Soiland-Reyes (UNIMAN), Alexandre Bonvin (UU), Adrien Melquiond (UU), Josep Lluís Gelpí (BSC), Daniele Lezzi (BSC), Steven Newhouse (EBI), Jose A. Dianes (EBI), Mark Abraham (KTH), Rossen Apostolov (KTH), Emiliano Ippoliti (Jülich), Adam Carter (UEDIN), Darren J. White (UEDIN)
Keywords	Tools, Workflows, Technological gaps
WP	WP2
Nature	Report
Dissemination Level	Public
Final Version Date	2016-03-30
Reviewed by	Erwin Laure (KTH), Alexandre Bonvin (UU), Ian Harrow (IHC)
MGT Board Approval	2016-03-30

Document History

Partner	Date	Comments	Version
IRB	2016-02-11	First draft	0.1
BSC, EBI	2016-02-25	New sections added: Introduction, initial roadmap proposal	0.2
IRB, BSC, UEDIN, UNIMAN, UU, KTH, Jülich	2016-03-04	Technological gaps analysis descriptions completed with all pilot use cases with contributions from all partners.	0.3
IRB	2016-03-07	First D2.1 draft distributed for review	0.4
IRB, BSC, UU, KTH	2016-03-18	Comments from the review addressed	0.5
IRB, BSC	2016-03-21	Initial Roadmap Proposal updated, executive summary extended	0.6
IRB, UU	2016-03-21	Comments from WP2 2016-03-21, 15h teleconference addressed Second D2.1 draft distributed for review	0.7
IRB, KTH, UNIMAN	2016-03-23	Final comments addressed, executive summary and introduction improved for clarity	0.8
UNIMAN	2016-03-30	Use case 6 related to core	0.9

Executive Summary

This deliverable describes the *state of the art* and gives a *technological gap analysis* in the portable environments for computing and data resources of BioExcel.

We review the commonly used technologies for *computational infrastructures*, a selection of *workflow managers* for computational biology and three important *repositories* for biomolecular data. We then provide a *catalogue of tools* that are supported by BioExcel partners, which will become the *building blocks* used in the pipelines and transversal workflow units of our pilot use cases.

We then describe the seven BioExcel *pilot use cases.* To help identify potential issues in developing the corresponding pipelines, the use cases have been individually described and analyzed, focusing on the set of functionalities (from the tool catalogue and elsewhere) that form a complete workflow. *Interoperability* between building blocks and data models are explored using workflow diagrams. Finally, we summarize the *technological gaps* for each use case.

We analyzed the *user feedback* from WP3 to highlight key focus areas for BioExcel's future work. From the initial WP3 survey together with previous HADDOCK and GROMACS surveys we identified three main areas of potential user interest: *Interoperability*, *usability* and *remotely accessible tools*. For the interoperability issue, we found that the need for manual interaction needs to be reduced, for instance by incorporating workflow managers to integrate processes and input/output data. For the usability part, we found that improvements could be made to the main codes (GROMACS, HADDOCK and CPMD) to ease their usage, such as web portals providing assistance on how to run, install or use advanced configuration options. Finally, we realized that a high number of users would be interested in using remote tools, although several concerns have been raised about this, namely data privacy, reliability, and lack of control.

Based on the analysis of the pilot use cases and the user survey, we present a summary of the identified technological gaps in section 5 “Global observations”.

The final section of the deliverable describes the immediate future *technology roadmap* presenting how BioExcel will utilize *cloud infrastructure*, develop *workflow building* blocks and provide a *tool deployment system* integrated with EGI and ELIXIR services. The initial setup will consist in the deployment of software blocks to perform the most commonly demanded operations, as gathered from Use Case analysis. These blocks and workflows will be deployed, tested and verified in the already available Barcelona Supercomputing Center (BSC) cloud infrastructure, and eventually transferred to the production BioExcel’s portal hosted at the European Bioinformatics Institute (EMBL-EBI).

Contents

1 Introduction

Biomolecular research has experienced a significant change over the past decades, and now strongly involves computational techniques across almost all areas of biology, including genomics, the understanding of structure and dynamics of macromolecules, and the simulations of molecular processes.

The use of computers in biology is so ubiquitous that the advance of the research itself is often conditioned by the advance of computer equipment or software engineering.

New parallelization strategies and computer accelerators, like *Graphical Processor Units* (GPUs) and *Field-Programmable Gate Arrays* (FPGAs), have allowed molecular dynamics to reach simulation times with biological significance (μs and beyond).

Current technology can now match computational analysis times with the rhythm of production of modern sequencing centres, enabling large scale projects where thousands of individual genomes are included, paving the way for personalized medicine.

Life Sciences is one of the largest and fastest growing communities in need of high-end computing. However, this fascinating technical improvement has unfortunately not met any parallel development in the user communities.

With a large ensemble of computational tools available for biomolecular research, it can be extremely challenging for a user to acquire a comprehensive view of the field, or even to be able to choose the most appropriate tool for a given problem.

The *BioExcel Centre of Excellence* aims to contribute to solving this challenge. There is however no unique solution, as different biological problems usually require different kinds of tools, and very often a combination of them.

There is already general agreement in the community about the need for a systematic way to *discover and access data and tools* within a unified and standardized computational environment; and also to be able to reuse tools in different environments, *scaling* for increased problem size (e.g. metagenomics or public health genomics).

The vast amount of data, both in the genomics and the structural fields, makes it impossible to manually coordinate a computational analysis. This converts, for instance, the process of setting up a protein system for simulation (which can be easily performed by any experienced modeler using a series of helping software tools) into an impossible task when the same process has to be made at the proteome level.

An increasing set of tools now automate the most usual computational procedures in the biomolecular research field (e.g. MDWeb [1], NAFlex [2], SeaBed [3], HADDOCK [4]). Most of these make use of finely tuned workflows where individual tools are combined in the most appropriate way to fulfill a particular procedure.

This new paradigm is already accepted in the field of genomics, but it is still in the early stages in the structural biology and simulation fields. Automation require a new set of software elements, *workflow managers* and programming models (e.g. KNIME [5], Taverna [6], Galaxy [7], Copernicus [8], COMPSs) and quite often, *web-based interfaces*. These are traditionally absent in HPC computing, where most calculations are still file and command-line based.

Additionally, complex workflows may require a large collection of complementary software, which can generate incompatibility issues, not only for data interoperability, but also differences in hardware requirements and computer architectures.

Computational *cloud infrastructures* come in hand to solve many of these issues, providing *virtualized environments* to package complex software installations and configurations in a portable and reliable way. This enables the same procedure to be performed in more than one computational environment, and even scale the infrastructure to accept problems of different sizes, without the need for a system administrator to individually install and optimize workflow components and tools.

The building of such *portable packaging* is one of the objectives of the BioExcel CoE. We believe there is no need to generate new software. The present offer of tools (the basic "building blocks"), workflow managers, and computational environments are suited enough to cover most aspects of biomolecular computational research.

There is however, a lack of organization and connectivity of the different components, and there are no recommended off-the-shelf solutions to allow general users to approach large-scale procedures. BioExcel aims to build portable software environments covering a large enough set of computational operations.

A series of *use cases* and *workflows*, representative of common operations, have been analyzed. This document presents a general analysis of the *components* that are available, the missing *technology* gaps that we observe in this process, and an initial *roadmap* to solve the observed issues.

2 State of the art of portable environments for computing

Global overview and consortium expertise:

2.1 Portable environments for computing

Technological advances in the recent years have eased the deployment of bioinformatics tools. New infrastructures, such as *Virtual Machines* or *Docker containers* allow encapsulating informatics packages to distribute them minimizing the tedious installation process.

Biomolecular simulations field, on the other hand, has been taking advantage of *High Performance Computing (HPC)* services for many years now. HPC, either in supercomputers or on GRID platforms (a *High Throughput Computing* platform (*HTC*)), are used to execute single and complex applications onto a large number of processor cores, using parallelization of the code and/or distribution of the computations into a large number of single jobs. However, simulations are just one step of a usual biomolecular study. Pre (system setup) and post (analyses) processes are time-consuming and critical steps that require expertise and need to be integrated to build a complete pipeline. This integration is commonly done using scripting languages (mostly Perl and/or Python), but specific managers to organize these workflow tasks exist.

The state of the art of **infrastructures** and **workflow managers** for the biomolecular simulation field are discussed in the next points.

2.1.1 Infrastructures

Tools, workflows and web portals offered by BioExcel will be deployed and run in the most appropriate software environments, depending on their specific requirements. This will require the use of state-of-the-art architectures such as web-services, virtual Machines or Docker containers, together with more traditional HPC systems. Next sections describe in more detail these infrastructures.

2.1.1.1 Virtual machines

Virtual machine (**VM**) is software that emulates dedicated hardware. The end user has the same experience on a **VM** as they would have on dedicated hardware. Usage of **VMs** has increased exponentially in the recent years, mainly due to the explosion of *cloud computing* platforms (section 2.1.1.5).

Virtual machines offer a number of advantages over physical machines:

- Ease of software installation process: Working with complex pipelines using a variety of programs with different dependencies makes the installation of them tedious in most cases. With a **VM**, one can prepare it to run a determined workflow installing all the software needed, and then just distribute the **VM** itself, with no need of subsequent modification.

- Ease of cloning: Cloning a **VM** is as easy as copying a file, whereas cloning a physical machine requires installing the same hardware pieces and software packages together. This also helps in backups, as all the information needed to reproduce the **VM** is condensed in just one file.

- High availability: Distributing load across **VMs** we can ensure high availability of applications and data: we can have the same **VM** running in more than one physical computer, or we can easily boot up a new one with minimal downtime or data loss.

- Scalability: Related to the previous point, **VMs** allow scalability on demand, as a new one can be easily launched if needed. But they also can be expanded much easier than physical machines, adding RAM memory or increasing the number of processors in just a few minutes.

VMs also have some disadvantages, mostly related to security, as in the majority of the cases they are run in public *cloud computing* platforms such as Amazon Web Services (AWS). When using them for distributing software tools or pipelines, the weakness point is the machine size in disk, as a complete OS is always saved, regardless of what libraries/dependencies the software needs to run. This is one of the main reasons for the appearance of *Docker* containers discussed in the next section.

2.1.1.2 Docker

Docker is a Linux container virtualization platform that is popular for distributing and running server and command line applications in a reproducible manner, and to form a distributed **microservice architecture**.

A Linux container is a special kernel feature, which similarly to *chroot* jails, behave as a separate machine, but unlike **VMs** described in the previous section; do not have the overhead of virtualization of hardware.

Docker is popular in the *devops* movement as it provides an easy way to install dependencies for software development and deployment, e.g. to run servers for mySQL, Apache Solr or node.js.

In brief, a **Docker** Image contains a virtual Linux file system (e.g. a miniature Debian installation). A ***Docker Container*** is a particular execution of a **Docker** Image, which typically runs a single process as installed within the container, and may have network ports exposed to the world, or have parts of the host computer's file system mounted within the inner container.

One great advantage of **Docker** is that it simplifies tool installation, as each **Docker** image is a self-contained Linux distribution, which doesn't have to be compatible with the host computer (beyond the kernel), and it's easy to try out a different tool or tool version without causing irreversible changes.

Docker runs on Linux natively; for Windows and OS X users **Docker** automatically manage a virtual machine running the Linux containers. **Docker** containers can also be deployed on the cloud or a local cluster, e,g. using Docker Machine.

Docker images can be created from a ***Dockerfile***, which basically lists the commands to run to prepare the image. **Docker** images can be chained together using base images - for instance to build on an image with mySQL, the Dockerfile says FROM mysql followed by additional commands like ADD (to include new files) or RUN (to run a command within the container).

Thus **Docker** is also an important tool for reproducibility, as these images can be automatically kept up to date and are distributed through the Docker hub. In bioinformatics, this has led to Bioboxes, a standard for creating interchangeable bioinformatics software containers.

Docker is not compatible with all Grid/HPC architectures - as it requires certain Linux kernel features and the nodes often run older distribution. Another potential blocker for HPC users is that central **Docker** base images assume an amd64 processor architecture - using **Docker** on other CPUs would require compiling all Docker base images yourself - which would negate some of the advantages.

2.1.1.3 High Performance Computing (HPC): Supercomputers, GPUs, Grid, Cloud

High Performance Computing (**HPC**) most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering or business.

The present generation of computers takes benefit of parallelism and accelerators to speed up calculations. The most used bioinformatics software packages have been long ago compatible with the *Message Passing Interface* (*MPI*), a protocol for computer-to-computer communication that permits work sharing between processors. When a large number of computer cores can be used simultaneously, MPI can greatly reduce the computation time. With the current availability of large **supercomputers**, usually built-up with more than 10,000 processors, MPI has become the most popular technique to run, for example, MD simulations or NGS read alignments.

On the other hand, with just a few years, a new hardware resource coming from the field of computer gaming has risen as the best platform to perform massive parallel calculations: *Graphical Processing Units* (**GPUs**). These processors, specifically designed to accelerate the generation of frames per second in 3D-games, have found to be extremely efficient for running for example MD algorithms. They can deliver over an order of magnitude more floating-point operations per second than classical central processing units

(CPUs). Unfortunately, parallelization of processes across GPUs is difficult, because communication between **GPUs** remains slower than communication between classical processors, although advances toward direct GPU-GPU communication have been recently presented, which would certainly improve scalability. New HPC strategies are thus going to a combination of MPI and **GPU** processes.

In a completely opposite way to the supercomputing approach, large initiatives use **Distributed Computing** (DC) or **Grid Computing** (GC). These approaches use a collection of computer resources from multiple locations to perform calculations. The idea is to divide a huge amount of data into little independent pieces and send them to the distributed computers. Using this approach, it is possible to overtake the performance of a supercomputer. This divide-and-conquer approach has tackled one of the main challenges in molecular science: protein folding (http://folding.stanford.edu). It is also the mechanism used by the HADDOCK portal, one of the BioExcel flagship software, to distribute computations on the EGI grid resources (see below).

Cloud Computing is an internet-based computing model for enabling ubiquitous, on demand access to a shared pool of configurable computing resources. In essence, cloud computing provides hardware and/or software as a service, over the Internet. Where this hardware and/or software are located and how they work is hidden in background, within the Internet *cloud*.

Cloud computing together with **VMs** (described in section 2.1.1.1) and most recently also with **Docker** containers (section 2.1.1.2) is widely used in business, and is starting to be popular in life sciences. A great example in Europe is the **European Grid Infrastructure** (**EGI**) infrastructure, that combines **Grid** and **Cloud Computing**.

EGI (http://www.egi.eu) is a publicly funded e-infrastructure put together to give scientists access to more than 530,000 logical CPUs, 200 PB of disk capacity and 300 PB of tape storage to drive research and innovation in Europe. EGI provides both high throughput computing and cloud compute/storage capabilities. Resources are provided by about 350 resource centers who are distributed across 56 countries in Europe, the Asia-Pacific region, Canada and Latin America.

The **EGI Federated Cloud** consists of a seamless grid of academic private clouds and virtualized resources, built around open standards and focusing on the requirements of the scientific community. Federated Cloud resources are generated in a collaborative way. Developers create the appropriate virtual machines published as Virtual Appliances under specific virtual organizations (VOs). **EGI** FedCloud provides IaaS (Infrastructure as a service), following the traditional **Cloud** approach. Users get access to the appropriate cloud sites and manage the virtual resources incorporating the necessary VOs, and data.

Following the cloud principles, web services and interactive applications can be easily integrated in the infrastructure, the computing environments can be finely tuned to satisfy user's needs in terms of software (OSs and software packages) and hardware (number of cores, amount of RAM, etc.) and, many solutions are available to store, update and access big amounts of data. Usage models enabled by the **EGI** Federated Cloud can be classified as follows:

- **Service hosting:** the EGI Federated Cloud can be used to host any IT service as web servers, databases, etc. Cloud features, as elasticity, can help users to provide better performance and reliable services.

- **Compute and data intensive:** applications needing considerable amount of resources in term of computation and/or memory and/or intensive I/O. Ad-hoc computing environments can be created in the FedCloud sites also to satisfy very hard HW resource requirements.

- **Datasets repository:** the EGI Federated Cloud can be used to store and manage large datasets exploiting the big amount of disk storage available in the Federation.

- **Disposable and testing environments:** environments for training or testing new developments.

2.1.1.4 Web Servers, Web services, Workflows managers

The number of on-line bioinformatics web servers in life sciences is growing at an incredible speed. To illustrate that, we can refer to the *Bioinformatics Links Directory* of *Nucleic Acid Research* (http://bioinformatics.ca/links_directory), currently (2016) containing an impressive number of 1,548 web server tools registered. The same *NAR* journal publishes every year a specific issue dedicated to Web Servers. Currently, 13 different web server issues have been published, presenting an average of 100 on-line tools each. And that is just a small number of the whole web-based projects available, most of them advertised only via publications, laboratory web pages or even existing in relative obscurity.

Although these interactive resources have been of enormous benefit to the scientific community over the years, there is still a growing demand for programmatic interfaces allowing the linkage of databases and on-line tools in automated analysis pipelines. A technology that allows this linkage by definition is becoming increasingly popular in life sciences: *Web Services* (*WS*). WS can be easily accessed from most programming languages, and joined together to build complex workflows. In fact, this is one of the strongest points of WS, their capacity to be chained together to form complex workflows.

Some *Workflow managers* and *Graphical User Interfaces* (*GUI*) to build and manage workflows from WS have been designed during the past years (e.g. *Taverna*, *Galaxy)* and they are described briefly in the next section of the deliverable. A repository of public workflows is available at http://www.myexperiment.org. *myExperiment* is an online research environment that supports the social sharing of bioinformatics workflows, currently containing more than 3,700 workflows registered. myExperiment users are developers interested in contributing their workflows into the repository for sharing them with the scientific community and also scientists wishing to discover workflows to be reused in their own research. *myExperiment* currently has over 10,000 members, showing the great interest from the scientific community for bioinformatics workflows.

2.1.2 Workflow Managers

Complex scientific studies require elaborate software pipelines, interconnecting different tools and information data. This is clearly reflected by the BioExcel pilot use cases presented in section 3. In some cases, the complexity of the pipeline forces the definition of processes dependencies, especially when those processes can be launched asynchronously in HPC systems. To tackle this particular issue, we will use specific programs named workflow managers, already introduced in the previous section. Some of the commonly used as well as newly developed workflow managers in computational biomolecular field are collected and described in the following sections:

2.1.2.1 Copernicus (www.copernicus-computing.org)

Copernicus is a peer to peer distributed computing platform designed for high level parallelization of statistical problems. It provides:

- Easy and effective consolidation of heterogeneous compute resources
- Automatic resource matching of jobs against compute resources
- Automatic fault tolerance of distributed work
- A workflow execution engine to easily define a problem and trace its results live
- Flexible plugin facilities allowing programs to be integrated to the workflow execution engine

Copernicus consists of four components: the Server, the Worker, the Client and the Workflow execution engine. The Server is the backbone of the platform and manages projects, generates jobs (computational work units) and matches these to the best computational resource. Workers are programs residing on the computational resources. They are responsible for executing jobs and returning the results back to the Server. Workers can reside on any type of machine - desktops, laptops, cloud instances or a cluster environment. The Client is the tool for setup of projects and their monitoring. In fact, nothing is running on the Client ever, it only sends commands to the server. That way the researcher can run the Client on a laptop, fire up a project, close the laptop, open it up after some time and see the progress of the project. All communication between these three components is encrypted and has to be authorized.

2.1.2.2 Galaxy

Galaxy is an *open, web-based platform for data intensive biomedical research*. **Galaxy** can be accessed on a free public server http://usegalaxy.org/, or installed locally in the lab.

Rather than building a workflow up-front, Galaxy uses a *data playground* approach, effectively building a workflow implicitly by applying a series of operations on the data items, keeping a *History* of all intermediate data items that are produced (and how they were made), making it easy to rerun parts of the workflow and share the results with others.

Galaxy has tight integration with a large collection of tools for genomics and sequence analysis, and is therefore popular for Next-Gen Sequencing data analysis. Adding a new tool is done by making a little Python wrapper and a description.

Maintaining a **Galaxy** instance can be a challenge, as it means also keeping track of all the installed tools and reference datasets. Recently **Galaxy** is also available as a Docker image, which simplifies the installation.

Galaxy is working on Common Workflow Language support.

2.1.2.3 KNIME

The open source KNIME workflow system is popular in cheminformatics for data analysis, statistics and visualization. **KNIME** runs as a graphical desktop application, but can also be used on the command line.

KNIME workflows are written as a *dataflow*, connecting a series of operations passing table-based data items. A typical workflow operation will extend the table by adding new columns (e.g. calculated properties) or summarize inputs to a new, smaller table.

KNIME have rich visualization and plotting for supported data types, and allow each operation to be run step by step, or when data or services have changed, re-run all dependent upstream operations as in a Makefile.

A **KNIME** workspace contains a workflow and the data values produced by the latest executions, and can be shared as a ZIP file or folder. KNIME can be extended with plugins developed in Java.

KNIME is heavily used in Open PHACTS and by pharmaceutical companies.

2.1.2.4 Apache Taverna

Apache Taverna (incubating) is a Java-based ***scientific workflow system*** with a graphical design interface. **Taverna** workflows can combine many different service types, including REST and WSDL services, command line tools, scripts (e.g. BeanShell, R) and custom plugins (e.g. BioMart).

Taverna workflows can be executed on the desktop, on the command line, or on a **Taverna** server installation, which can be controlled from a web portal, a mobile app, or integrated into third-party applications.

Taverna is used in a wide range of sciences for data analysis and processing, including bioinformatics, cheminformatics, biodiversity and musicology. Workflow engine features include provenance tracking, implicit parallelism/iterations, retry/failover and looping.

Taverna workflows are commonly shared on myExperiment, and can either be created graphically in the Taverna workbench, programmatically using the Taverna Language API or by generating workflow definitions in the SCUFL2 format.

2.1.2.5 COMPSs/PyCOMPSs (https://www.bsc.es/computer-sciences/grid-computing/comp-superscalar)

COMPSs is a framework, composed of a programming model and a runtime system, which aims to ease the development and deployment of distributed applications and web services. The core of the framework is its programming model, which allows the programmer to write applications in a sequential way and execute them on top of heterogeneous infrastructures exploiting the inherent parallelism of the applications. The COMPSs programming model is task-based, +allowing the programmer to select the methods of the sequential application to be executed remotely. This selection is done by means of an annotated interface where all the methods that have to be considered as tasks are defined with annotations describing their data accesses and constraints on the execution of resources. At execution time this information is used by the runtime to build a dependency graph and orchestrate the tasks on the available resources.

The **COMPSs** programming model syntax enables the easy development of applications as composite services. A composite, called Orchestration Element (OE), is written as a sequential program from which other services and regular methods, namely Core Elements (CE), are called. Therefore, composites can be hybrid codes that reuse functionalities wrapped in services or methods, adding some value to create a new product that can also be published as a service. Besides, all the information needed for data-dependency detection and task-based parallelization is contained in a separate annotated Core Element Interface (CEI).

Any **COMPSs** application can be composed of two different kinds of CE: Method CE and Service CE. Method CEs are regular methods of the application selected to be run remotely. To pick a method CE, the programmer declares the method in the CEI, adding the @Method annotation indicating the implementing class.

On their turn, Service CEs correspond to SOAP Web Service operations described in WSDL documents. To select a SOAP operation as a CE, the developer declares the service operation together with the @Service annotation describing the service details (namespace, service name and service port). The location of the service is not included in the CEI, but instead in the runtime configuration that actually decides which server will run the task; thus, the programming model syntax remains completely unaware of the underlying infrastructure.

One important feature of the **COMPSs** runtime is the ability to exploit the cloud elasticity by adjusting the amount of resources to the current workload. When the number of tasks is higher than the available cores, the runtime turns to the cloud looking for a provider offering the type of resources that better meet the requirements of the application and with the lowest economical cost. Analogously, when the runtime detects an excess of resources for the actual workload, it will power off unused instances in a cost-efficient way. Such decisions are based on the information on the type of resources that contains the details of the software images and instance templates available for every cloud provider. Since each cloud provider offers its own API, COMPSs defines a generic interface to manage resources and to query about details concerning the execution cost of multiple cloud providers during one and the same execution. These, called connectors, are responsible for translating the generic requests to the actual provider's API.

COMPSs does not provide only a programming model. The framework is complemented with a set of platform tools which facilitates (i) the development of the **COMPSs** applications by means of an Integrated Development Environment (IDE); (ii) the deployment of applications in distributed infrastructures by means of the Programming Model Enactment Service (PMES); and (iii) the monitoring of executions by means of the Monitoring and Tracing tools.

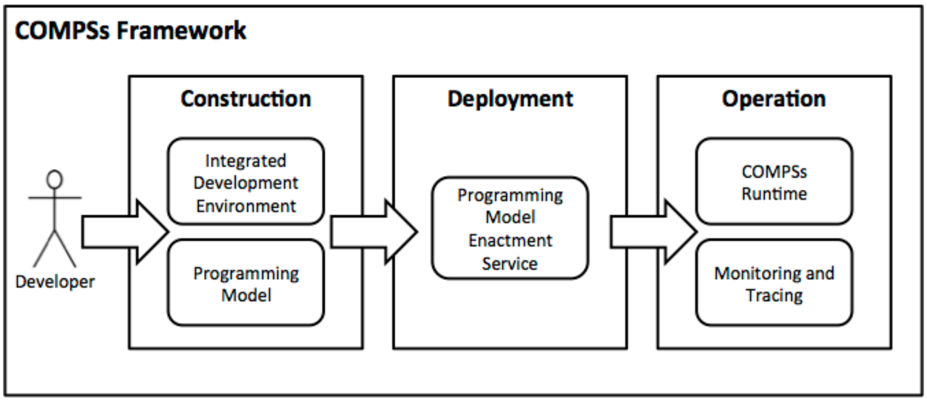


Fig. 2.1 – COMPSs Framework architecture

The transparent deployment of **COMPSs** applications on cloud infrastructures is delegated to the PMES PaaS component, whose architecture is depicted in Fig. 2.2. Via a Basic Execution Service (BES) interface, the PMES exposes the needed operations to the **COMPSs** IDE dealing with the intricacies of the deployment and contextualization operations, and the installation of the application packages, the required libraries, and the monitoring processes. A dashboard is also available for the configuration of the user cloud environment.

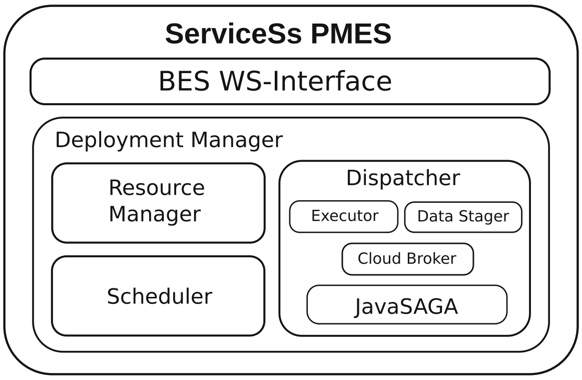


Fig. 2.2 – PMES architecture

The runtime of **COMPSs** provides some information at execution time so that the user can follow the progress of the application through a web interface that shows real-time information on the tasks being executed and on the usage of the resources.

At the end of each execution or file transfer, the **COMPSs** runtime also creates usage records. The usage records contain information about the resources involved in the task execution, the source and destination resources in data transfers, and the start and end time of each operation. Once the application completes, all these usage records can be processed by the Tracing tool in order to perform a post-mortem reconstruction of the application execution across the different cloud resources. This reconstruction can be visualized by tools such as *Paraver* in order to detect bottlenecks and unbalanced parts of the application which could be fixed to increase the application performance.

2.1.3 Data repositories

BioExcel tools and workflows will be integrated with a set of biomolecular data resources: Genomic data, protein structure & targets, chemical and MD repositories.

This section gives an overview of the main biological data repositories that will be incorporated in BioExcel functionalities.

2.1.3.1 European Bioinformatics Institute (EMBL-EBI)

EMBL-EBI, maintains the world’s most comprehensive range of freely available and up-to-date biological databases and reference data repositories These databases can be accessed through the EBI web portal, or programmatically, using the available web services and APIs.

Databases offered cover a broad range of life sciences’ fields:

- **DNA & RNA**: genes, genomes and variations.
- **Proteins**: sequences, families and motifs.
- **Gene expression**: RNA, protein and metabolite expression.
- **Structures**: Molecular and cellular structures.
- **Systems**: Reactions, interactions and pathways.
- **Chemical biology**: Chemogenomics and metabolomics.
- **Ontologies**: Taxonomies and controlled vocabularies.
- **Literature**: Scientific publications and patents.
- **Cross domain**: Cross-domain tools and resources.

In particular, some of the most popular databases in the computational biomolecular field are hosted in the EBI servers: Ensembl, UniProt, PDBe (Protein Data Bank in Europe), and Reactome.

EMBL-EBI databases follow a set of principles of service provision:

- **Open** - data and tools are freely available, without restriction. The only exception is potentially identifiable human genetic information, for which access depends on research consent agreements.

- **Compatible** - EMBL-EBI is a world leader in the development of global bioinformatics standards, which are key to data sharing.

- **Comprehensive** - EMBL-EBI resources are comprehensive and up to date. EMBL-EBI works with publishers to ensure that biological data must be placed in a public repository and cross-referenced in the relevant publication.

- **Portable** - All of the data and many of the software systems can be downloaded and installed locally.

- **High quality** – EMBL-EBI databases are enhanced through annotation: highly qualified biologists add value to databases by incorporating features of genes or proteins from other sources, and automated annotation is subjected to rigorous quality control.

2.1.3.2 Institute for Research in Biomedicine (IRB)

IRB institute hosts a set of databases, data portals and APIs focused on macromolecular structures and in particular to macromolecular structure's dynamics and flexibility. The three main databases that will be available through BioExcel are briefly described in the following sections:

- **PDB mirror**: IRB maintains an up-to-date mirror of the PDB data bank, that can be accessed programmatically through a complete RESTful API (http://mmb.irbbarcelona.org/api/).

- **Molecular Dynamics Extended Library (MoDEL)**: Database of protein molecular dynamics simulations, holding currently 1,800 different simulations, with compressed trajectories available for downloading.

- **BigData Nucleic Acids Simulations (BigNASim)**: Database of nucleic acids molecular dynamics simulations, with the possibility to download trajectories and meta-trajectories formed joining different trajectories.

2.1.3.3 Open PHACTS

The **Open PHACTS Discovery Platform** has been developed to reduce barriers to drug discovery in industry, academia and for small businesses. It is going to be used by BioExcel as a pharmacological dataset. The main points of Open PHACTS platform are stated below:

- It contains all the data sources you already use, integrated and linked together so that you can easily see the relationships between compounds, targets, pathways, diseases and tissues. Data sources included are: ChEBI, ChEMBL, ChemSpider, ConceptWiki, DisGeNET, DrugBank, Gene Ontology, neXtProt, UniProt and WikiPathways.
- The platform has been used to answer complex questions in drug discovery and results have been published in peer reviewed scientific journals.

- The platform was built in collaboration with a large consortium of major academic and commercial organisations involved in drug discovery.

- The platform is founded on semantic web and linked data principles and uses industrial strength tools such as Virtuoso to provide fast and robust access to the chemistry and biological data sources that you trust.

- Data can be accessed via the Open PHACTS API or explored using the Open PHACTS Explorer and many other apps developed using the API.

- The data within the platform is available in a variety of formats to suit the applications you already use. Formats include *JSON*, *XML*, *TSV* and *RDF*.

2.2 Application Building Blocks

Here we present a set of available modules that will be the basis for the center solution-oriented workflows. All together, this set of building blocks covers a large portion of what is needed for the development of the project and for the progress of the pilot use cases described in section 3 of this document.

The complete list of tools is introduced in the next steps, divided in six main areas, from the more general to the more specific.

Information is displayed in tables, divided in:

- **Tool**: Name of the tool and link to its web page, if available.
- **Institute**: Institute hosting or authoring the tool.
- **Type**: How the tool is presented (Web portal, software, etc.).
- **Platform/Interface**: Where the tool is implemented and runs (VM, Web, HPC, etc.).
- **Workflow management**: Workflow manager used in/for the tool.
- **Dependencies**: Specific software dependencies.
- **Potential users**: Potential users of the tool.
- **Description**: Brief description of the tool.

Before listing the set of tools, we summarize the identified technologies that can be used to organize workflow tasks already described in section 2.1.2. As previously mentioned, they basically differ on the platform they are able to work with. They will be used in the generation of workflows within the project, as well as the pipelines produced in the different pilot use cases.

Table. 2.2 – Library of modules: Workflow managers

Tool	Institute	Type	Platform	Description
Copernicus	KTH	Peer to peer distributed computing platform	HPC/Cloud	Automatic HPC workflow generation
COMPSs	BSC	Programming Framework	HPC/Cloud	Automatic workflow generation
Galaxy	-	Scientific Workflow System	Web/cloud/server	Data-driven. Publishable workflows.
Apache Taverna	Uniman	Scientific Workflow System	Desktop/server/web	Flow-driven, components. Shared on myExperiment
KNIME	-	Scientific Workflow System	Desktop	Flow-driven, embedded run data

2.2.1 Sequence analysis

Set of tools to retrieve annotations from genes, get gene expression, run comparative genomics and search for known protein/domain 3D structures. From annotation, read aligners and NGS general analysis to prediction of pathological mutations from protein sequences.

Table. 2.2.1 – Library of modules: Sequence Analysis

Tool / Portal / Workflow	Institute	Type	Platform	Workflow Management	Dependencies	Potential Users
PMut	IRB	Web Portal	Web	No	-	Health field with interest in pathological mutations
BWA /SamTools /Picard	BSC	Cloud Env.	VM	-	-	NGS sequencing users
Maker	BSC	Cloud Env.	VM	-	-	Genome annotation users
GATK	BSC	Cloud Env.	VM	-	-	NGS sequencing users

Bowtie	BSC	Cloud Env.	VM	-	-	NGS sequencing users
Tophat	BSC	Cloud Env.	VM	-	-	NGS sequencing users
transPLANT/INB Cloud	BSC	Cloud Env.	Web / Web Service / Galaxy / HPC	COMPSs	-	Genome analysis users
BCBio-nextgen	KTH	Software	Command Line	Galaxy	-	NGS sequencing users
EBI Tools Portfolio	EMBL-EBI	Web Portal	Web	Programmatic interface available for client side workflows	-	Global

Tool / Portal / Workflow	Description
PMut	Pmut is a software aimed at the annotation and prediction of pathological mutations, and in particular to answer the following question: given a mutation happening at a specific location in a protein sequence, can we say whether it can be pathological (that is, it can lead to disease for the carrier) or non-pathological/neutral (no effect on the carrier's health)?
BWA /SamTools /Picard	Aligner / Mapper for NGS reads, utilities for sequence management
Maker	*Genome annotation tool*
GATK	Analysis of NGS sequence data
Bowtie	Read aligner
Tophat	Read aligner for RNAseq data
transPLANT/INB Cloud	The Transplant Cloud environment is an integrated environment based on openNebula Cloud manager, powered by COMPSs/PMES. Includes generic genome analysis tools and other more oriented on plant genomics. Access is possible through a web portal, web services (SOAP) and through a Galaxy interface.
BCBio-nextgen	A python toolkit providing best-practice pipelines for fully automated high throughput sequencing analysis. You write a high-level configuration file specifying your inputs and analysis parameters. This input drives a parallel pipeline that handles distributed execution, idempotent processing restarts and safe transactional steps. The goal is to provide a shared community resource that handles the data processing component of sequencing analysis, providing researchers with more time to focus on the downstream biology.
EBI Tools Portfolio	EMBL-EBI hosts over 100 tools (both well-known public tools and internally developed software) that are openly available for users around the world - http://www.ebi.ac.uk/services. Programmatic interfaces to many of our services and tools are documented - http://www.ebi.ac.uk/Tools/webservices/.

2.2.2 Molecular Dynamics

A large variety of tools dealing with molecular simulations: software to run atomistic and coarse-grained molecular dynamics and quantum mechanics simulations; web portals to setup and run MD simulations; databases storing trajectories and metadata from MD simulations; web portals to compute protein conformational transitions; software to compress MD trajectory files.

Table. 2.2.2 – Library of modules: Molecular Dynamics

Tool / Portal / Workflow	Institute	Type	Platform/Interface	Workflow Management (present)	Dependencies	Potential Users
Gromacs	KTH	Software	Command-line	Copernicus	-	Expert users interested in Molecular Dynamics simulations
CPMD	Jülich	Software	Command-line	-	-	Expert users interested in quantum mechanics simulations
MDWeb	IRB	Web Portal	Web	No	-	Non-experts users interested in Molecular Dynamics simulations
MDMoby	IRB	Web Services / Workflows	Web / Command-line	Taverna	BioMoby	Non-experts users interested in high-throughput Molecular Dynamics simulations
MDdMD	IRB	Web Portal	Web	No	-	Non-experts users interested in macromolecular conformational transitions
GOdMD	IRB	Web Portal	Web	No	-	Non-experts users interested in macromolecular conformational transitions
DISCRETE (DMD)	IRB	Software	Command-line	Not yet	-	Life Science field with interest in macromolecular (protein) flexibility
MoDEL	IRB	Web Portal	Web	No	-	Life Science field with interest in macromolecular (protein) flexibility
BIGNASim	IRB	Web Portal	Web	No	-	Life Science field with interest in macromolecular (nucleic acids) flexibility
FlexServ	IRB	Web Portal	Web	No	-	Life Science field with interest in macromolecular (protein) flexibility
DNAlive	IRB	Web Portal	Web	-	-	Life Science field with interest in macromolecular (nucleic acids) flexibility
NAFlex	IRB	Web Portal	Web	No	-	Life Science field with interest in macromolecular (nucleic acids) flexibility
PCASuite	IRB	Software	Command-line	Not yet	Numerical Libraries (C)	Life Science field with interest in macromolecular (protein) flexibility
GROMACS grid-enabled WeNMR portal	UU	Web Portal	Web	python / csh scripts / gLite middleware for grid submission	Currently runs gromacs 4.5.3 - remotelly deployed on grid sites	Life Science field with interest in macromolecular (protein) flexibility

Tool / Portal / Workflow	Description
Gromacs	A Molecular Dynamics package primarily designed for biomolecular systems such as proteins and lipids.
CPMD	Parallelized plane wave/pseudopotential implementation of density functional theory (DFT), particularly designed for ab-initio molecular dynamics.
MDWeb	Web-based platform to help access to molecular dynamics (MD). The platform provides tools to prepare systems from PDB structures mimicking the procedures followed by human experts. It provides inputs and can send simulations for three of the most popular MD packages (Amber, NAMD and Gromacs). Tools for analysis of trajectories are also incorporated.
MDMoby	Set of semantic Web-Services to help access to molecular dynamics (MD) simulations. Semantic information is added using the BioMoby library and a MD Ontology.
MDdMD	MDdMD is a web portal for determining pathways for conformational transitions in macromolecules based on the use of discrete molecular dynamics and biasing techniques based on a combination of essential dynamics and Maxwell-Demon sampling techniques.
GOdMD	GOdMD is a web portal for determining pathways for conformational transitions in macromolecules based on the use of discrete molecular dynamics and biasing techniques based on a combination of essential dynamics and Maxwell-Demon sampling techniques.
DISCRETE (DMD)	Coarse-Grained Molecular Dynamics simulation package
MoDEL	Database of protein Molecular Dynamics simulations, with 1800 trajectories representing different structural clusters of the PDB.
BIGNASim	Database of nucleic acids Molecular Dynamics simulations
FlexServ	Web portal offering a complete set of macromolecular flexibility analyses from a coarse-grained or atomistic MD trajectory
DNAlive	DNAlive is a tool for the analysis of structural and physical characteristics of genomic DNA.
NAFlex	NAFlex is a web tool for the analysis of nucleic acids flexibility, both isolated and protein-bound.
PCASuite	Software tool to handle PCA analyses of macromolecular coarse-grained or atomistic MD trajectory
GROMACS grid-enabled WeNMR portal	The WeNMR GROMACS web portal combines the versatility of this molecular dynamics package with the calculation power of the eNMR grid. This will enable you to perform many simulations from the comfort of your internet browser anywhere in the world. The server is furthermore aimed to provide a user friendly and efficient MD experience by performing many preparation and optimization steps automatically. Note: requires registration with the WeNMR Virtual Research Community (www.wenmr.eu), registration with the enmr.eu VO (requires thus a valid X509 personnal certificate). The portal uses a single sign-on mechanism implemented in WeNMR

2.2.3 Molecular Modeling

Set of web portals to generate 3D-structures, both for proteins and nucleic acids, from just an amino acid or nucleotide sequence, including modeling of protein mutants.

Table. 2.2.3 – Library of modules: Molecular Modeling

Tool / Portal / Workflow	Institute	Type	Platform	Workflow Management (present)	Dependencies	Potential Users
CS-Rosetta WeNMR webportal	UU	Web Portal	Web	python scripts / cron deamons	Rosetta3 + various analysis tools	Life Sciences researchers, structural biologist (in particular NMR ones)
3D-DART	UU	Web Portal	Web	no	3DNA software	Life Sciences researchers, structural biologists
NAFlex	IRB	Web Portal	Web	no	AmberTools	Life Sciences researchers, structural biologists interested in Nucleic acids structures
MDWeb	IRB	Web Portal	Web	No	-	Non-experts users interested in Molecular Dynamics simulations
MDMoby	IRB	Web Services / Workflows	Web / Command-line	Taverna	BioMoby	Non-experts users interested in high-throughput Molecular Dynamics simulations

Tool / Portal / Workflow	Description
CS-Rosetta WeNMR webportal	CS ROSETTA is a protocol which generates 3D models of proteins, using only the 13CA, 13CB, 13C', 15N, 1HA and 1HN NMR chemical shifts as input. Based on these parameters, CS ROSETTA uses a SPARTA-based selection procedure to select a set of fragments from a fragment-library (where the chemical shifts and the 3D structure of the fragments are known). The fragments are assembled using the ROSETTA protocol. The generated models are rescored based on the difference between the back-calculated chemical shifts of the generated models and the input chemical shifts. Note: requires registration with the WeNMR Virtual Research Community (www.wenmr.eu), registration with the enmr.eu VO (requires thus a valid X509 personnal certificate). The portal uses a single sign-on mechanism implemented in WeNMR
3D-DART	The 3D-DART server (3DNA-Driven DNA Analysis and Rebuilding Tool) provides a convenient means of generating custom 3D structural models of DNA with control over the local and global conformation. 3D-DART uses the DNA rebuild functionality of the well-known software package 3DNA Lu et al. and extends its functionally with tools to change the global conformation of the DNA models.
NAFlex	NAFlex is a web tool for the analysis of nucleic acids flexibility, both isolated and protein-bound, with the possibility to generate nucleic acids structures (A/B-DNA, A/B-RNA, right-handed, left-handed, etc.) from a nucleotide sequence. NAFlex uses the nucleic acid builder (nab) program from AmberTools package.
MDWeb	Web-based platform to help access to molecular dynamics (MD). The platform provides tools to prepare systems from PDB structures mimicking the procedures followed by human experts. It provides inputs and can send simulations for three of the most popular MD packages (Amber, NAMD and Gromacs). Tools for analysis of trajectories are also incorporated. It can be used to generate protein mutants.
MDMoby	Set of semantic Web-Services to help access to molecular dynamics (MD) simulations. Semantic information is added using the BioMoby library and a MD Ontology. It can be used to generate protein mutants.

2.2.4 Docking

Set of web portals to run macromolecular flexible docking (protein-protein) and small ligand screening (protein-ligand), modeling biomolecular complexes.

Table. 2.2.4 – Library of modules: Docking

Tool / Portal / Workflow	Institute	Type	Platform	Workflow Management (present)	Dependencies	Potential Users
HADDOCK	UU	Web portal	Web (support xml-rpc access)	Internal workflow - python + job management systems (local batch system)	Multiple software required (CNS, Molprobity, PRODRG, PROFIT, NACCESS, R)	>6500 registered users worldwide for the web server
HADDOCK grid-enabled	UU	Web portal	Web (support xml-rpc access)	Internal workflow - python + job management systems (local batch system + script for grid submission for the grid-enabled portal (used both gLite and DIRAC4EGI submission scripts - various versions of the portal))	Multiple software required (CNS, Molprobity, PRODRG, PROFIT, NACCESS, R)	>6500 registered users worldwide for the web server
SeaBed	IRB	Web Portal	Web	No	-	-

Tool / Portal / Workflow	Description
HADDOCK	HADDOCK (High Ambiguity Driven protein-protein DOCKing) is an information-driven flexible docking approach for the modeling of biomolecular complexes. HADDOCK distinguishes itself from ab-initio docking methods in the fact that it encodes information from identified or predicted protein interfaces in ambiguous interaction restraints (AIRs) to drive the docking process. HADDOCK can deal with a large class of modeling problems including protein-protein, protein-nucleic acids and protein-ligand complexes. Offers multiple access interfaces to the users: - the Easy interface - the Prediction interface - the Expert interface (requires Expert level access) - the Refinement interface (requires Expert level access) - the Guru interface (requires Guru level access) - the Multi-body interface (requires Guru level access) - the File upload interface - generate AIR files for multibody docking
HADDOCK grid-enabled	
SeaBed	The SEABED web server integrates a variety of docking and QSAR techniques in a user-friendly environment. SEABED goes beyond the basic docking and QSAR web tools and implements extended functionalities like receptor preparation, library editing, flexible ensemble docking, hybrid docking/QSAR experiments or virtual screening on protein mutants.

2.2.5 Chemoinformatics

Set of tools to extract chemical patterns from hits, refine docking processes, and associate annotations (e.g. toxicity) on lead candidates.

Table. 2.2.5 – Library of modules: Chemoinformatics

Tool / Portal / Workflow	Institute	Type	Platform	Workflow Management (present)	Dependencies	Potential Users
SeaBed	IRB/BSC	Web Portal	Web	No	-	-

Tool / Portal / Workflow	Description
SeaBed	The SEABED web server integrates a variety of docking and QSAR techniques in a user-friendly environment. SEABED goes beyond the basic docking and QSAR web tools and implements extended functionalities like receptor preparation, library editing, flexible ensemble docking, hybrid docking/QSAR experiments or virtual screening on protein mutants.

2.2.6 Pharmacology queries

Set of workflows allowing integrated access to the Open PHACTS discovery platform.

Table. 2.2.6 – Library of modules: Pharmacology queries

Tool / Portal / Workflow	Institute	Type	Platform	Workflow Management (present)	Dependencies	Potential Users
Open PHACTS	Uniman	Web services	Web, Docker	KNIME, Taverna, Pipeline Pilot	-	Pharmacology, proteomics, genomics

Tool / Portal / Workflow	Description
Open PHACTS	Open PHACTS bringing together pharmacological data resources in an integrated, interoperable infrastructure, accessible as an REST API, the browser and workflows, all of which are free to use (with a high per-user call quota). The Open PHACTS dataset integrate data from ChEBI, ChEMBL, ChemSpider, ConceptWiki, DisGeNET, DrugBank, FAERS, Gene Ontology, neXtProt, UniProt and WikiPathways using Linked Data, identity mapping and chemical structure matching. Open PHACTS is currently working on adding the SureCHEMBL chemical patent dataset to the platform. The Open PHACTS platform is developed as open source, and can also be installed by third-parties using virtual machines and Docker. Open PHACTS was bootstrapped from IMI funding, and is now run by the Open PHACTS Foundation where several pharma companies are members. The foundation is also a partner of the BigDataEurope H2020 project.

3 Technological gaps based on the defined use cases

Analyses focused on finding technological gaps for the set of 7 pilot use cases (UCs) stated in the project are presented in the next sections. For an easy identification of the technological gaps within the different pipelines, analyses were divided in four main sections:

- **Description**: Brief description of the pilot use case.

- **Functionalities**: Building blocks contained in the use case pipeline. Almost all of them coincide with tools listed in the previous section (application building blocks). N/A: Not Applicable; ?: still not known.

- **Diagram**: Graphical representation of the pipeline, showing dependencies, input, output and intermediate data formats and building blocks interoperability.

- **Discussion**: Brief discussion about tools and data repositories used, and specific technological gaps identified in the particular pilot use case.

The global technological gaps identified after the analysis of the different pilot use cases are discussed in section 5 (Global observations).

3.1 Pilot Use Case 1: Genomics

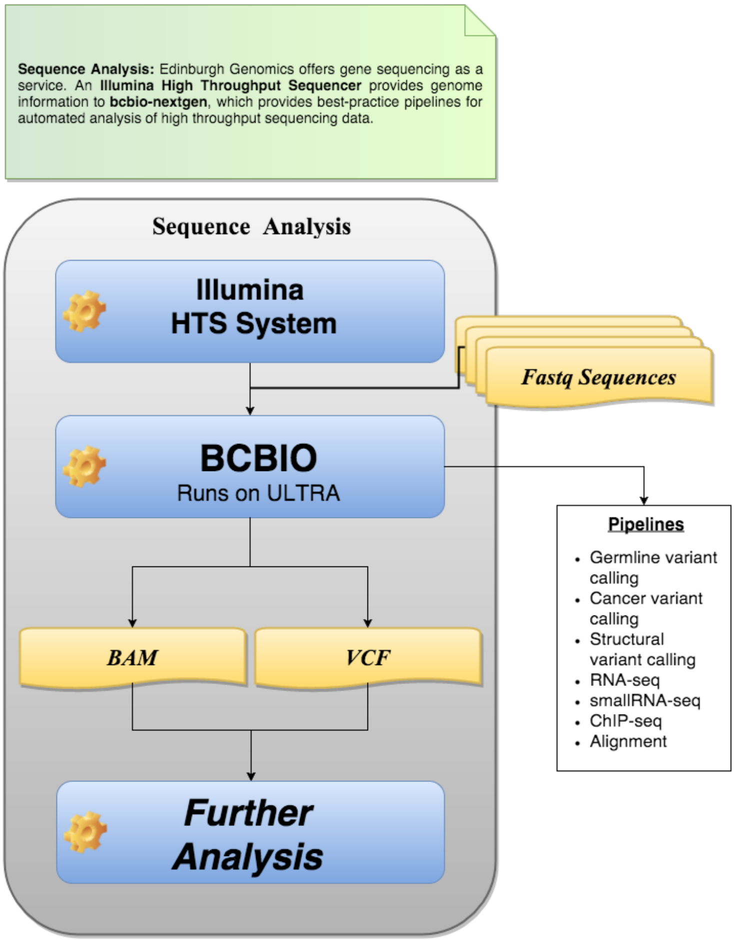
- **Description**:

Pilot use case 1 is based on the genome sequencing service provided by Edinburgh Genomics. An Illumina High Throughput Sequencer (HTS) machine provides genome information, which is passed to the BCBio-nextgen package, providing a range of best-practice pipelines for automated analysis of high throughput sequencing data. The package includes pipelines for variant calling, alignment, RNA-seq analysis and ChIP-seq analysis. The development and implementation of sequencing in HPC and “Big Data” analytics, in particular workflow and data management, is a key interest in this use case. BCBio-nextgen development towards this is well underway, and the implementation of the pipelines on HPC systems is an avenue of future work currently being assessed. BCbio-nextgen is also AWS and Docker compatible.

- **Functionalities**:

Functionalities	Gap Analysis		
	Tool availability	Required Input Data	Access/management mode
Standard alignment	BCBio-nextgen pipelines	FASTQ sequences BCBio-nextgen configuration files	CLI, python scripts.
Germline variant calling			
Cancer variant calling			
Structural variant calling			
RNA-seq			
smallRNA-seq			
ChIP-seq			
			No global workflow management

- **Diagram**:



Sequence Analysis: Edinburgh Genomics offers gene sequencing as a service. An **Illumina High Throughput Sequencer** provides genome information to **bcbio-nextgen**, which provides best-practice pipelines for automated analysis of high throughput sequencing data.

Fig. 3.1 – Pilot Use Case 1: Genomics

- **Discussion**:

Input data for this pilot use case comes from an Illumina High Throughput Sequencing (HTS) system, as a set of Fastq sequences. The package BCBio-nextgen co-developed in the Science for Life laboratory in Stockholm, Sweden, is used to run a complete set of analysis (pipelines) on those sequences. Further analysis, still to be determined, will be run as a last step of the workflow. There's no global workflow manager defined to run the entire pipeline.

3.2 Pilot Use Case 2: High-throughput ensemble molecular simulations

- Description:

Finding valid pathways through free-energy landscapes: implementation of the "string of swarms" method using **Copernicus** (www.copernicus-computing.org) as a workflow manager, and **GROMACS** as a compute engine.

- Functionalities

Functionalities	**Gap Analysis**			
	Tool availability	**Required Input Data**	**Interoperability**	**Access/management mode**
Model physics	GROMACS, third parties	Choice of force field, any extra parameterization, dynamics settings	N/A	User interaction
Reaction coordinate	N/A	Starting and ending configurations, reduced-dimensionality description of transition	N/A	User interaction
System preparation	GROMACS, pmx, etc.	Starting and ending configurations	N/A	Shell scripts, command-line tools, web portals
Minimization	GROMACS	Prepared initial coordinate, topology and minimization input files	N/A	Copernicus, from command line
Thermalization & Equilibration	GROMACS	Minimized coordinates, topology and dynamics input files, run-time optimization settings	N/A	Copernicus, from command line
Swarms	GROMACS	Equilibrated coordinates, topology and dynamics input files, run-time optimization settings	N/A	Copernicus, from command line
Reparameterization	GROMACS analysis tools	PMF estimate from swarms	N/A	Copernicus, from command line

- Diagram:

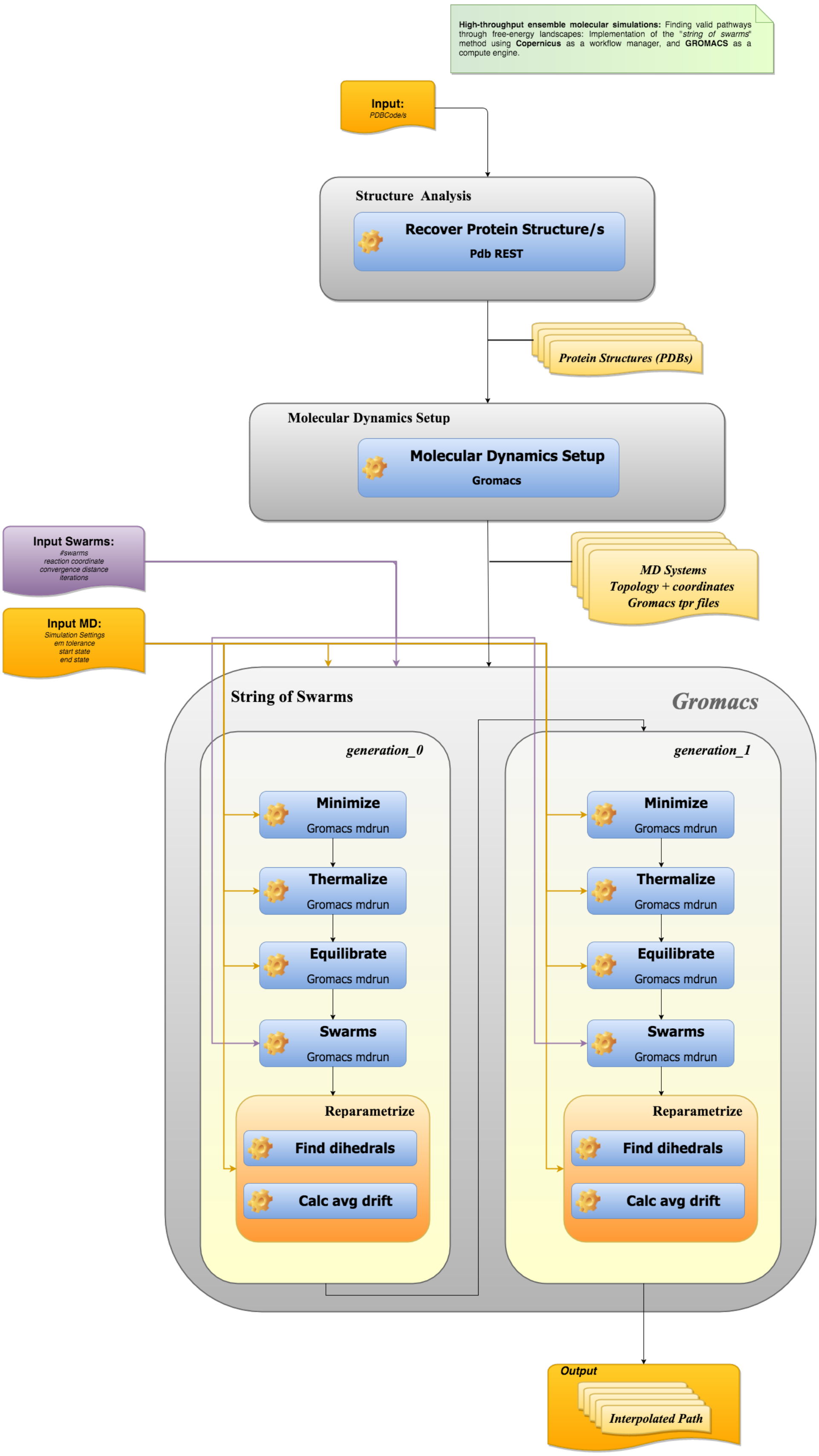


Fig. 3.2 – Pilot Use Case 2: HT ensemble molecular simulations

- **Discussion**:

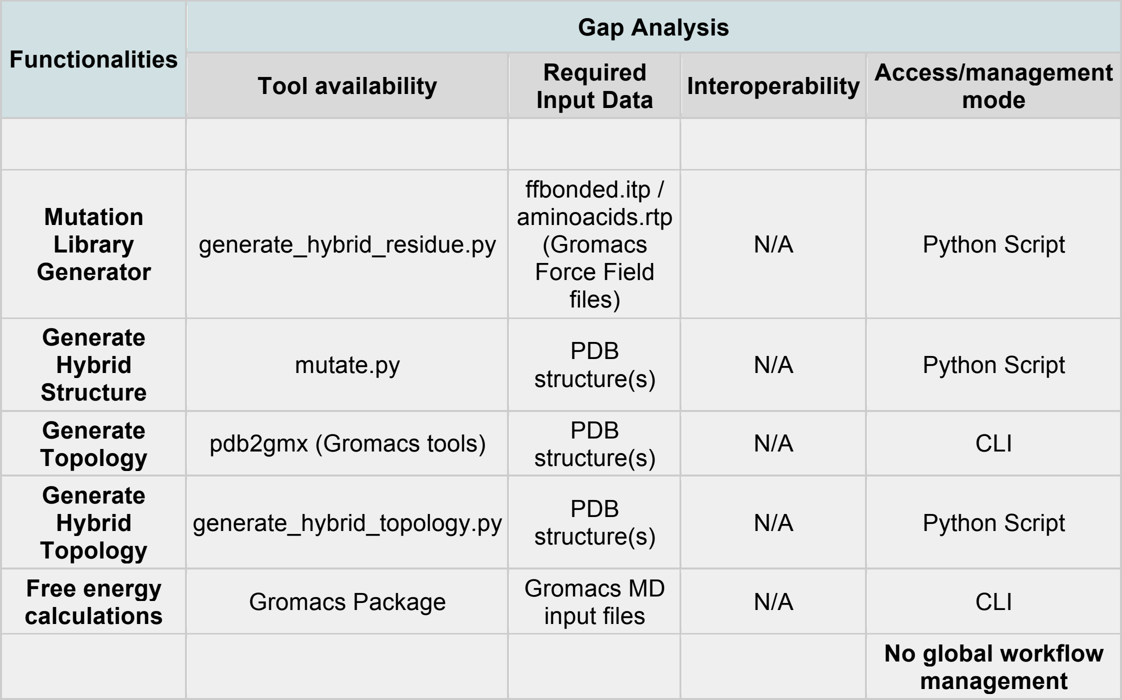
Input data needed for this pilot use case is taken from the PDB databank using RESTful services (EBI, IRB). All calculations within the pipeline are done using GROMACS MD package (system preparation, minimization, equilibration, MD run, swarms and reparametrization), thus, there are no interoperability problems. Copernicus Workflow manager will manage the entire execution of the workflow. There's a particular step that needs user interaction, the definition of the reaction coordinate.

3.3 Pilot Use Case 3: Free energy simulations of biomolecular complexes

- Description:

Free energy simulations of biomolecular complexes: development and application of the **pmx** framework to generate optimal mappings for arbitrary amino acid mutations in several modern molecular mechanics force fields and **GROMACS** MD package for free energy simulations.

- Functionalities



Functionalities	**Gap Analysis**			
	Tool availability	**Required Input Data**	**Interoperability**	**Access/management mode**
Mutation Library Generator	generate_hybrid_residue.py	ffbonded.itp / aminoacids.rtp (Gromacs Force Field files)	N/A	Python Script
Generate Hybrid Structure	mutate.py	PDB structure(s)	N/A	Python Script
Generate Topology	pdb2gmx (Gromacs tools)	PDB structure(s)	N/A	CLI
Generate Hybrid Topology	generate_hybrid_topology.py	PDB structure(s)	N/A	Python Script
Free energy calculations	Gromacs Package	Gromacs MD input files	N/A	CLI
				No global workflow management

- Diagram:

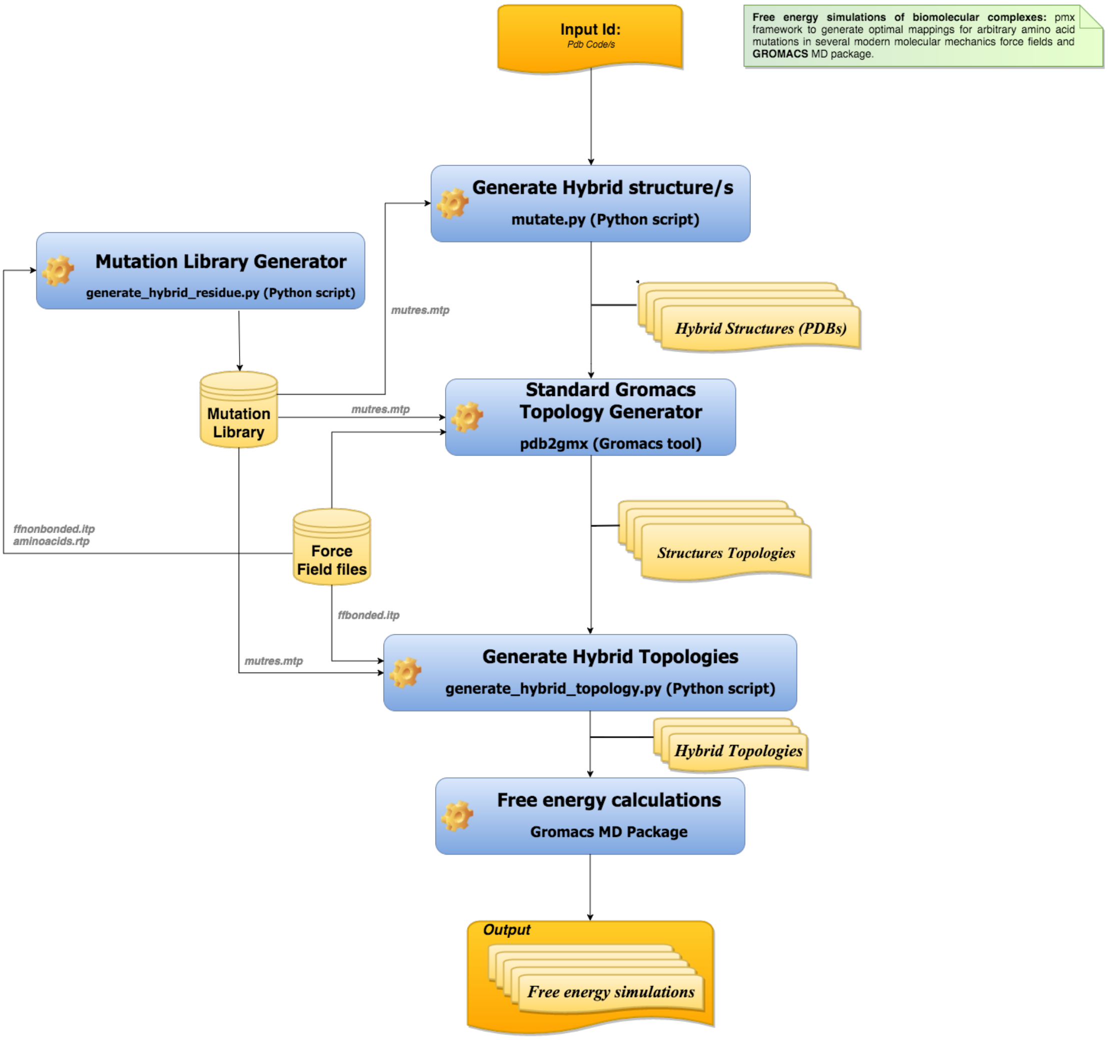

Free energy simulations of biomolecular complexes: pmx framework to generate optimal mappings for arbitrary amino acid mutations in several modern molecular mechanics force fields and **GROMACS** MD package.

Fig. 3.3 – Pilot Use Case 3: Free energy simulations of biomolecular complexes

- **Discussion**:

Input data for this pilot use case is a protein structure, that can be taken from PDB database (EBI, IRB). All internal executions are run using Python scripts with system calls to GROMACS MD package. There's no global workflow manager defined to run the entire pipeline. The current **pmx** version supports multiple flavors of the modern molecular mechanics force fields OPLS, AMBER, and CHARMM and all amino acid mutations (including charge changing mutations) except for mutations including proline. **pmx** is using the GROMACS infrastructure for alchemical free energy calculations and as such is compatible with implemented equilibrium free energy schemes (thermodynamic integration and free energy perturbation) as well as non-equilibrium schemes (Jarzynski equality, Crooks fluctuation theorem). A recent extensive benchmark indicates an overall accuracy for changes in thermostability upon mutation of approx. 1 kcal/mol for modern molecular mechanics force fields.

3.4 Pilot Use Case 4: Multi-scale modeling of molecular basis for odor and taste

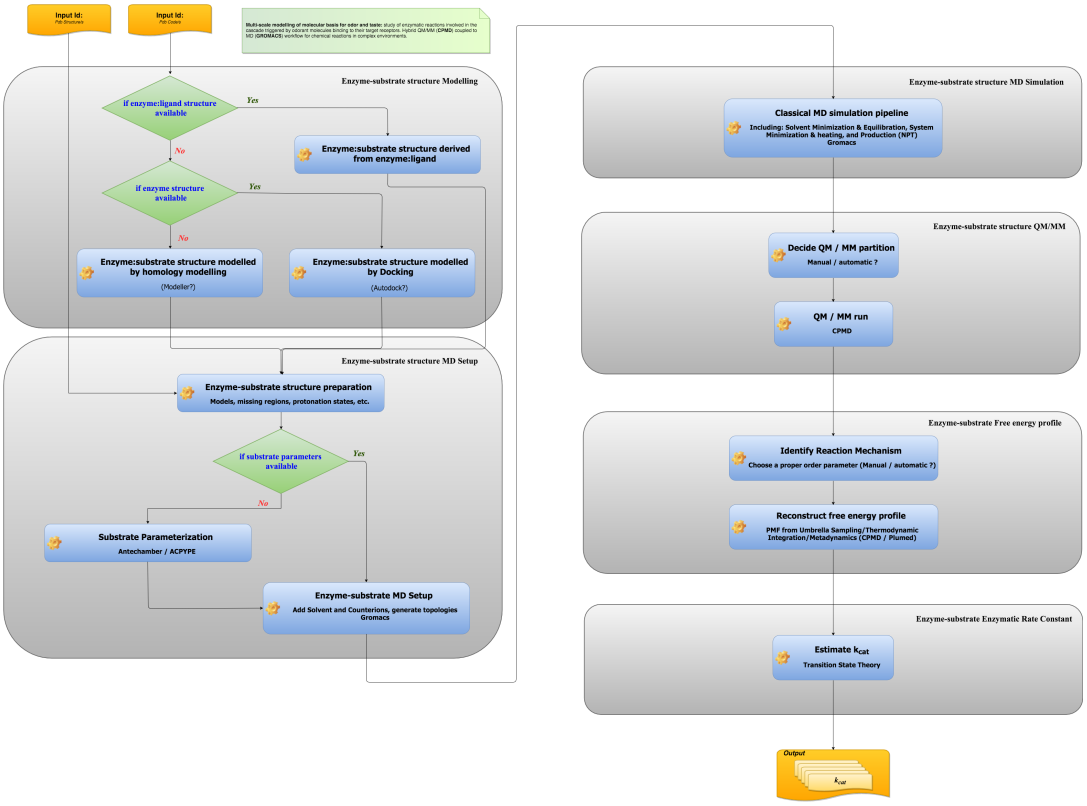
- Description:

Multi-scale modeling of molecular basis for odor and taste: study of enzymatic reactions involved in the cascade triggered by odorant molecules binding to their target receptors. Hybrid QM/MM (**CPMD**) coupled to MD (**GROMACS**) workflow for chemical reactions in complex environments.

- Functionalities:

Functionalities	**Gap Analysis**			
	Tool availability	**Required Input Data**	**Interoperability**	**Access/management mode**
Recover Protein Structure	PDB Rest (EBI, IRB)	PDB Id	Id mapping	REST
Enzyme-Substrate Structure Modelling	**Not available: Comparative Modelling Tool Modeller ? / Autodock ?**	PDB structures(s)	?	?
Substrate parameterization	**Not available: Antechamber ? / ACPype ?**	PDB structures(s)	?	?
MD Setup	Gromacs / MDWeb	PDB structures(s)	N/A	WEB & Scripting
MD Simulation	Gromacs	Gromacs MD input files	N/A	WEB / CLI / Copernicus
QM/MM Simulation	CPMD	PDB structure(s)	N/A	CLI
Free Energy Profile	CPMD / Plumed	CPMD Trajectory	N/A	CLI
Estimate Kcat	CPMD	Free energy profile	N/A	CLI
				No global workflow management

- Diagram:



Multi-scale modelling of molecular basis for odor and taste: study of enzymatic reactions involved in the cascade triggered by odorant molecules binding to their target receptors. Hybrid QM/MM (**CPMD**) coupled to MD (**GROMACS**) workflow for chemical reactions in complex environments.

Input Id:
Pdb Structure/s
Input Id:
Pdb Code/s
Enzyme-substrate structure Modelling
if enzyme:ligand structure available
Yes
No
Enzyme:substrate structure derived from enzyme:ligand
if enzyme structure available
Yes
No
Enzyme:substrate structure modelled by homology modelling
(Modeller?)
Enzyme:substrate structure modelled by Docking
(Autodock?)
Enzyme-substrate structure MD Setup
Enzyme-substrate structure preparation
Models, missing regions, protonation states, etc.
if substrate parameters available
Yes
No
Substrate Parameterization
Antechamber / ACPYPE
Enzyme-substrate MD Setup
Add Solvent and Counterions, generate topologies
Gromacs
Enzyme-substrate structure MD Simulation
Classical MD simulation pipeline
Including: Solvent Minimization & Equilibration, System Minimization & heating, and Production (NPT)
Gromacs
Enzyme-substrate structure QM/MM
Decide QM / MM partition
Manual / automatic ?
QM / MM run
CPMD
Enzyme-substrate Free energy profile
Identify Reaction Mechanism
Choose a proper order parameter (Manual / automatic ?)
Reconstruct free energy profile
PMF from Umbrella Sampling/Thermodynamic Integration/Metadynamics (CPMD / Plumed)
Enzyme-substrate Enzymatic Rate Constant
Estimate k_{cat}
Transition State Theory
Output
k_{cat}

Fig. 3.4 – Pilot Use Case 4: Multi-scale modeling of molecular basis for odor and taste

- **Discussion**:

Input data for this pilot use case is a protein structure (enzyme) and its substrate (ligand). If that particular structure is solved, it can be taken from the PDB databank (EBI, IRB). If it isn't solved, it needs to be built using some modeling tool, such as Modeller or Autodock (currently not available in BioExcel list of tools). Next step in the workflow is a MD run. If MD parameter libraries for the substrate are not available, the substrate needs to be parameterized. This can be done with programs such as AnteChamber from Ambertools package, or ACPype (currently not available in our list of tools). MD setup and run is done using either MDWeb or GROMACS MD package. Next step involves a QM/MM calculation, which will be run using CPMD program. Reconstruction of the free energy profile is also computed with CPMD, and the final estimation of the k_{cat} is done by exploiting the assumption that transition state theory is valid and knowing the free energy profile obtained in the previous step.

This pipeline contains a couple of crucial steps, that needs user interaction: the QM/MM partition definition (definition of the regions that will be treated in a Quantum Mechanic or Molecular Mechanic way), and the identification of the reaction mechanism (order parameter) that will be used for the free energy profile calculation. On top of that, there's still no workflow manager defined to run the entire pipeline.

3.5 Pilot Use Case 5: Biomolecular recognition

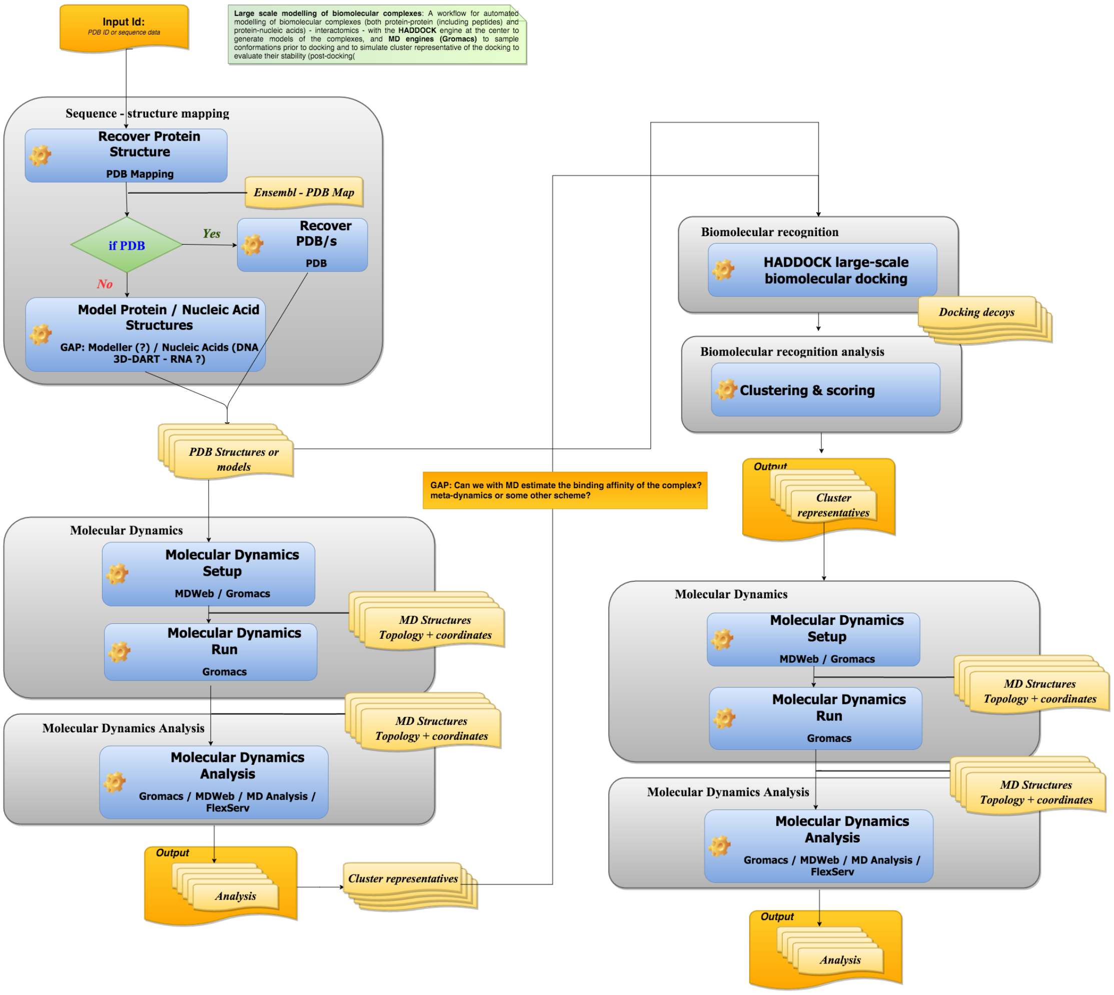
- Description:

Large scale modelling of biomolecular complexes: A workflow for automated modelling of biomolecular complexes (both protein-protein -incuding peptides- and protein-nucleic acids) - *interactomics* - with **HADDOCK** engine at the center to generate models of the complexes, and MD engines (**Gromacs**) to sample conformations prior to docking and to simulate cluster representative of the docking to evaluate their stability (post-docking).

- Functionalities:

Functionalities	**Gap Analysis**			
	Tool availability	**Required Input Data**	**Interoperability**	**Access/management mode**
Recover Protein/Nucleic Acids Sequence	Uniprot Rest (EBI, IRB)	Uniprot Id	Id mapping	REST
Recover PDB Code(s)	Uniprot Rest (EBI, IRB)	Uniprot Id	Id mapping	REST
Recover Protein/Nucleic Acids Structure	PDB Rest (EBI, IRB)	PDB Id	Id mapping	REST
Model Protein/Nucleic Acids Structure	**Not available: Comparative Modelling Tool (Modeller?)** Nucleic Acids (DNA/RNA - 3D-Dart / NAFlex)	Sequence & Variants	Variant mapping	Python based scripting (if Modeller)
Molecular Dynamics Setup	MDWeb (IRB)	PDB structures(s)	N/A	WEB & Perl Scripting
Molecular Dynamics Run	MDWeb / Gromacs	MD System	N/A	WEB / CLI / Copernicus
Molecular Dynamics Analysis	Some available: MDWeb / FlexServ / Gromacs	MD Trajectory	Trajectory formats	Mixed: WEB / CLI
Biomolecular recognition	HADDOCK large-scale biomolecular docking	PDB structure(s) (Clusters)	N/A	CLI
Biomolecular recognition analysis	Clustering / Scoring (HADDOCK)	PDB structure(s) (Docking decoys)	N/A	CLI
				No global workflow management

- Diagram:



Large scale modelling of biomolecular complexes: A workflow for automated modelling of biomolecular complexes (both protein-protein (including peptides) and protein-nucleic acids) - interactomics - with the **HADDOCK** engine at the center to generate models of the complexes, and **MD engines (Gromacs)** to sample conformations prior to docking and to simulate cluster representative of the docking to evaluate their stability (post-docking(

Fig. 3.5 – Pilot Use Case 5: Biomolecular recognition

- **Discussion**:

Input data for this pilot use case is a protein or a nucleic acids structure, either coming from a PDB code or a protein sequence (EBI, IRB APIs). If the protein isn't solved, it needs to be built using some modeling tool, such as Modeller or Autodock for proteins (currently not available in our list of tools) or 3D-Dart / NAFlex for nucleic acids. Next steps in the workflow are MD setup, run and analyses, done using either MDWeb or GROMACS MD package. A list of structure cluster representatives from the MD simulations will be used as inputs for HADDOCK large-scale biomolecular docking. Outputs of this docking procedure will be scored, and decoys selected will be eventually passed to a MD pipeline (setup, run and analysis) again.

A functionality that is missing in this pilot use case is how can the binding affinity of the complex be estimated. Also, there's no workflow manager defined to run the entire pipeline.

3.6 Pilot Use Case 6: Leveraging integrated pharmacological datasets for cross-domain queries

- Description:

Get approved drugs from a pathway of interest. This use case is based on a **KNIME** workflow for **Open PHACTS** (myexperiment.org/workflows/4292) published in PLOS One (doi:10.1371/journal.pone.0115460.g004). This workflow finds approved drugs that have potent activity against any target in the selected pathway, combining data from ChEMBL and DrugBank through the Open PHACTS platform. This workflow was developed for the KNIME workflow system by Emiliano Cuadrado et al.

The future goals of this use case are:

- Create workflow-neutral building blocks (e.g. **Docker**, Common Workflow Language) for **Open PHACTS API** calls
- Recreate the use case workflow in multiple workflow systems
- Make the workflow configurable for different API endpoints – should run on a local cloud installation of the **Open PHACTS** platform
- Expose the workflow as a command line tool

While this use case does not use the BioExcel core applications (CPMD, HADDOCK, GROMACS), it has potential to help select candidate compounds or targets for further biomolecular simulation and modelling, as shown in Use Case 7.

- Functionalities

Functionalities	**Gap Analysis**			
	Tool availability	**Required Input Data**	**Interoperability**	**Access/management mode**
Pathway information	Open PHACTS (UNIMAN)	WikiPathways URI	Id mapping	REST
Targets in Pathway	Open PHACTS (UNIMAN)	WikiPathways URI	Id mapping	REST
Pathways for target	Open PHACTS (UNIMAN)	Target URI	Id mapping	REST
Target information	Open PHACTS (UNIMAN), DrugBank	Target URI	Id mapping	REST
Target pharmacology	Open PHACTS (UNIMAN), ChEMBL	Target URI	Id mapping	REST
Compound information	Open PHACTS (UNIMAN)	Compound URI	Id mapping	REST
Filter	KNIME, Taverna (UNIMAN), grep	List, Selection criteria	N/A	Built-in, scripts

- Diagram:

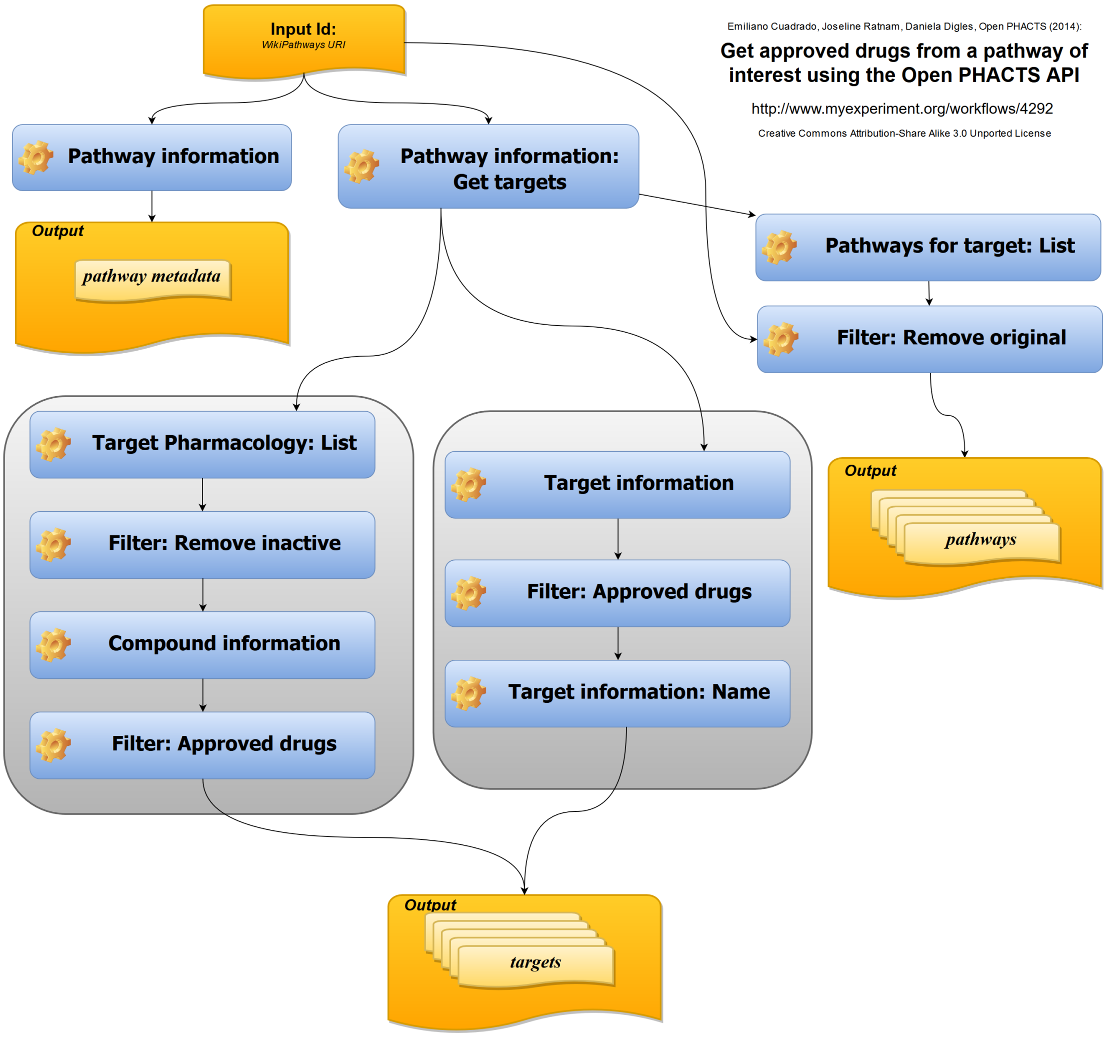

Emiliano Cuadrado, Joseline Ratnam, Daniela Digles, Open PHACTS (2014):
Get approved drugs from a pathway of interest using the Open PHACTS API
http://www.myexperiment.org/workflows/4292
Creative Commons Attribution-Share Alike 3.0 Unported License
Input Id:
WikiPathways URI
Pathway information
Pathway information: Get targets
Output
pathway metadata
Pathways for target: List
Filter: Remove original
Output
pathways
Target Pharmacology: List
Filter: Remove inactive
Compound information
Filter: Approved drugs
Target information
Filter: Approved drugs
Target information: Name
Output
targets

Fig. 3.6 – Pilot Use Case 6: Leveraging integrated pharmacological datasets for cross-domain queries

- **Discussion**:

While **Open PHACTS** provide an integrated API for querying pharmacological datasets, and the Open PHACTS KNIME nodes provide building blocks for creating such workflows, we have identified several issues that prevent workflow portability and reuse within the pharmacoinformatics community:

- Bound to a particular workflow system
  - e.g. a **KNIME** workflow won't open in **Apache Taverna** or **Galaxy**

    - Even adapting a workflow manually takes a large effort, as the **KNIME** workflow components are not portable to other systems.
    - Harder to integrate into wider conceptual workflow/architecture
- Bound to a particular version of the public **Open PHACTS API**
    - Cumbersome to update for newer API releases
    - Difficult to test if a newer API gives different results
    - Hinders reproducibility (e.g. the published workflow from 2014 uses **Open PHACTS** 1.3 API, which is now decommissioned)
- Difficulties in finding the right identifiers to start with (Name to URI mapping)
- Pharma companies don't want to share their current research interest publically – often they want a private installation of the **Open PHACTS** platform.
    - However **Open PHACTS** Workflows (and Workflow Components) are bound to a particular API endpoint and must be updated one by one.
    - Installing the platform is not straight forward as it requires ~200 GB of data and a stack of about 10 services. **VM** and **Docker** helps, but still training is needed to set it up correctly.
    - Adding own private data requires identity mapping and query modifications. Training is needed to understand how to customize the platform.

3.7 Pilot Use Case 7: Virtual screening

- Description:

Run ensemble docking using **Open PHACTS** to obtain pharmacological compounds, **Gromacs** MD engine to prepare MD ensembles and **Haddock / Seabed** to run biomolecular recognition.

- Functionalities

Functionalities	**Gap Analysis**			
	Tool availability	**Required Input Data**	**Interoperability**	**Access/management mode**
Recover Protein Structure	PDB Rest (EBI, IRB)	PDB Code	Id mapping	REST
Recover drug structures	Open PHACTS (UNIMAN)	PDB URI	Id mapping	REST
Prepare MD ensembles	**See Use Case 2**			
Enhance Sampling	**GAP: Essential Dynamics (?)**	?	?	?
Virtual Screening	SeaBed (IRB)	Drug parameterization	Drug structure formats	Web
Docking result analysis	SeaBed (IRB)	SeaBed Virtual Screening output	N/A	Web

				GAP: No global WF management

- Diagram:

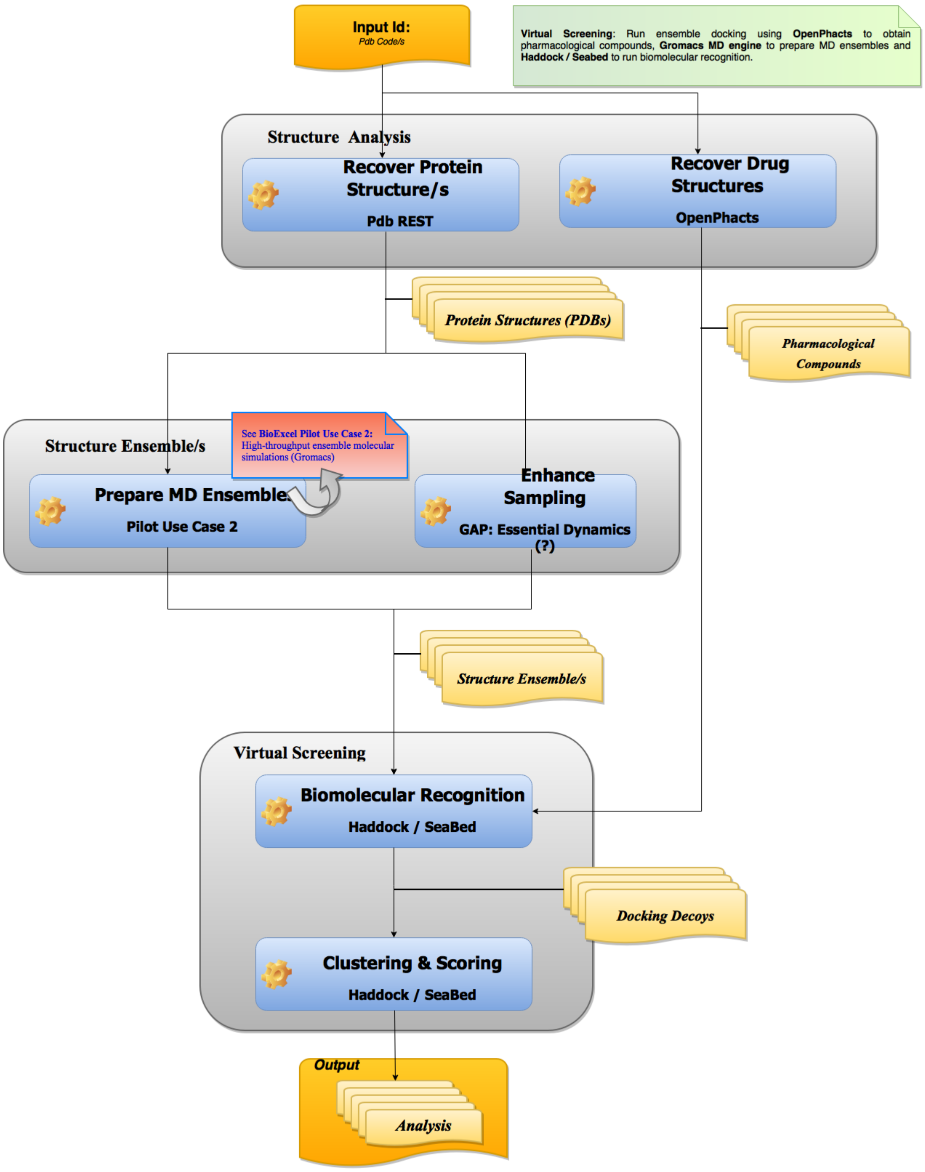

Input Id:
Pdb Code/s
Virtual Screening: Run ensemble docking using OpenPhacts to obtain pharmacological compounds, Gromacs MD engine to prepare MD ensembles and Haddock / Seabed to run biomolecular recognition.
Structure Analysis
Recover Protein Structure/s
Pdb REST
Recover Drug Structures
OpenPhacts
Protein Structures (PDBs)
Pharmacological Compounds
Structure Ensemble/s
See BioExcel Pilot Use Case 2: High-throughput ensemble molecular simulations (Gromacs)
Prepare MD Ensembles
Pilot Use Case 2
Enhance Sampling
GAP: Essential Dynamics (?)
Structure Ensemble/s
Virtual Screening
Biomolecular Recognition
Haddock / SeaBed
Docking Decoys
Clustering & Scoring
Haddock / SeaBed
Output
Analysis

Fig. 3.7 – Pilot Use Case 7: Virtual Screening

- Discussion:

Inputs for this pilot use case is protein structures that will be downloaded from EBI or IRB repositories and pharmacological compounds that will be extracted from Open PHACTS through its RESTful API. This use case can be extended to integrate the pathways approach to drug repurposing from Use case 6, enabling experimental testing of the hypothesis in relevant translation and safety models prior to any human studies.

In order to obtain an ensemble of structures from the initial one, the pipeline defined in pilot use case 2 (HT ensemble molecular simulations, see section 3.1.2) will be used, together with a tool to enhance this sampling (such as essential dynamics, not available in our list of tools). This final ensemble of structures will be mixed together with the pharmacological compounds to obtain a set of docking decoys using HADDOCK and/or SeaBed software tools. A final step of clustering and scoring will be run using the same docking tools. There's still no workflow manager defined to run the entire pipeline.

4 Initial user feedback from WP3

BioExcel WP3 (consultancy & user groups) conducted an initial survey to obtain a general view of user interests related to biomolecular research. The questionnaire was completed by a fairly small group of people who were known to the partners in the consortium, so care is needed when generalizing from the results (detailed description of the survey is presented in D3.1 "Selection and Establishment of User Groups").

In addition, we analyzed responses in two separate recent surveys with **HADDOCK** and **GROMACS** users. They were done prior to the start of **BioExcel** and even though the questions in them were not selected for the needs of the CoE, the results give useful insights of the users needs and practices.

Results from the three surveys bring up important points to be considered in the working plan for the next WP2 tasks. A summary is presented in the following sections.

4.1 Interoperability

Whilst the **BioExcel** survey did not explicitly ask about interoperability, this issue was raised in responses. In response to the question "*In which step of your workflow do you feel you could benefit from expert support?*" 37% of the respondents chose the option "Transfer of data between different software modules / different calculation steps".

There are various possible reasons why this could be difficult, such as incompatible file (or data transfer) formats (both in terms of syntax and semantics), difficulties with automating the steps and challenges associated with running different parts of the workflow in different places. The respondents who mentioned this issue are using a variety of different approaches to integrate the tools into their wider workflows. All but one of the respondents mentions that at least some steps are performed manually, either through SSH or by using a web frontend via a browser. Only one of the respondents who mentioned transfer of data is using a workflow system, and even this respondent mentions manual steps.

It would appear from this that there is scope for reducing the amount of manual interaction necessary and this suggests that there are technical gaps here with regard to system and process integration. This conclusion is supported by the **GROMACS** survey where 37% of users consider interoperability an important step; 18% of users are interested in improved tools for handling many simulations on heterogeneous resources, and approx. 20% would like to have access to an API for the core compute engine or the tools. Regarding interoperability with external codes, there is a big interest in improvements of support for visualization tools (e.g. VMD, PyMol), free energy calculations (PLUMED) and QMMM support (CP2K, Dalton, NWChem, Gaussian, Gamess etc.)

A recent **HADDOCK** user survey also invited respondents to comment on features that they would like to see implemented in future versions. The majority of these comments requested new (scientific) functionality and better analysis tools and very few of them pertain to interoperability. This is quite interesting when considered along side the fact that 79% of respondents use modelling software together with **HADDOCK**, so it would appear that users, on the whole, currently seem quite content with **HADDOCK's** interoperability. This is also supported by the variety of software that users report that they use with **HADDOCK** including MODELLER, Rosetta, I-TASSER, **GROMACS**, AutoDock, ElNemo and others. A small number of suggestions were made that relate to interoperability, namely: "Better file conversion to prepare ligands parameter files", "Should be format compatible", "I would like to see a standardized output for displaying docked structures.", "Easy incorporation of data of EM, MS, SA" and "standardization, a common API and integrated data handling would be highly appreciated". These responses suggest that in terms of interoperability, there are improvements that could be made with regard to file format compatibility.

List of selected responses:

- 37% of users consider **transfer of data between different software modules / different calculation steps** as an important step that will be benefit from expert support (*BioExcel*)

- 18% would like to see **improved tools for handling many simulations on heterogeneous resources (***GROMACS***)**

- 24% would like **an API for analysis tools and methods,** 16% **an API for mdrun functionality** *(GROMACS)*

- 30% would like **improved QMMM functionality** (*GROMACS*)

- 79% of users **combine HADDOCK it with other modeling software**.

4.2 Usability

Another possible area in which technical gaps can manifest themselves from a user point-of-view is in usability. In response to the question "*What do* *you consider to be the main challenges in adopting tools like Gromacs, Haddock, CPMD for your research and infrastructure?*" the following issues were mentioned: "Parameter tuning" (50% of respondents), "Configuration and commands" (27%) and "Compilation and installation" (23%). Most of these relate to possible improvements that could be made to the main codes themselves but it is possible that other technical solutions (such as web portals) could provide assistance, for example, with choice of appropriate configuration options.

Responses to the **HADDOCK** user survey which mention aspects of usability are also fairly few in number, although it is sometimes hard to differentiate between requests for additional functionality and requests to make existing functionality more accessible and usable. There are some requests for improved documentation, a "light version that is easier to use", and a comment that "input should be simplified" but on the whole it looks like the respondents are fairly happy with **HADDOCK's** usability. This is also supported by the fact that 81.9% of respondents gave a score of 4 or 5 out of 5 in response to the question, "*How satisfied are you with HADDOCK?*"

A third of **GROMACS** users have expressed interest in automation of drug-bonding calculations, and also 40% consider error checking in input files an important step for adoption by new users.

List of selected responses:

- 33% would like **improved tools for automating drug binding calculations** (*GROMACS*)

- 39% say that **tools for checking for errors in inputs files** are most useful to get new users trained (*GROMACS*)

- 50% consider **parameter tuning** to be a main challenge in adopting GROMACS/HADDOCK/CPMD (*BioExcel)*

- 27% consider **configuration and commands** to be a main challenge in adopting GROMACS/HADDOCK/CPMD (*BioExcel)*

- 23% consider **compilation and installation** to be a main challenge in adopting GROMACS/HADDOCK/CPMD (*BioExcel)*

4.3 Remotely accessible tools

The use of remotely accessible tools is related to the above issues, but also brings its own particular issues. In response to the question "*Do you use remotely accessible computational tools hosted by third-parties?*" the following results were obtained: "**CPMD**: 0%, **IRB/BSC**: 0%, **Gromacs** 46%, **Haddock** 21%". Since the number of respondents who are using **CPMD** is itself low, we cannot draw any real conclusions from the "0%" entries. It appears that the majority of users are using only local tools, but that there is a significant minority who are using online tools. What's more, in response to the question "*Would remote access to tools for MD sim or modeling be acceptable for your research?*", 68% of respondents answered yes. This suggests that there are reasons why people are *not* using remote tools even though it "would be acceptable". Unfortunately, from the responses here it is not possible to ascertain whether online services would be *preferable* or not.

Responses did, however, offer reasons why online services would not be acceptable. The most common reason was some aspect of privacy. Reliability, difficulty of file management, lack of control and lack of need were also mentioned.

GROMACS analysis tools are reportedly particularly useful to 85% of respondents, which presents an opportunity for offering some of their functionality for remote access.

In terms of responses to the **HADDOCK** user survey the results were even more pronounced: 92% of respondents use the web server and, furthermore, 56% of respondents use *only* the online version. For the 8% of respondents who do not use the web server, some people state that they are happy with the local version and have no need. As in the initial WP3 survey, privacy/data security is mentioned as a specific reason by several respondents (9 of the 50 respondents who do not use the web server). In several cases this appears to be a matter of policy rather than explicit mistrust of the **HADDOCK** server and the way it works, so in this case it is fairly unlikely that there is a technical gap that BioExcel can address. Other reasons for *not* using the web server include the relative performance of the web server version, flexibility in terms of certain workflows (e.g. "I prefer a local installation to streamline the process of optimizations of models.") and available functionality (e.g. "Did not allow ligand-protein systems"). These responses suggest that there are possible improvements that could be made to expose more of **HADDOCK's** functionality in the online version, but it is not immediately clear whether this is desirable from the point of view of the service provider (some functionality might, for example, use excessive compute resources on the server).

List of selected responses:

- 85% of users find **analysis tools** particularly useful (GROMACS) => integrate more of the tools with portals
- 46% of GROMACS users **utilize remotely accessible tools** hosted by third parties *(BioExcel)*
- 21% of HADDOCK users **utilize remotely accessible tools** hosted by third parties (*BioExcel)*
- 68% consider **remote access to tools for MD** acceptable for research; 16% said no due to privacy considerations (*BioExcel)*

5 Global observations

After in depth description and analysis of the set of pilot use cases and the initial feedback from WP3 communities, presented in the project, several common technological gaps were identified:

Workflow management: Almost all of the use cases exist as a set of tools/functionalities that are interconnected using scripting or just manually run step by step. Pipelines for the pilot use cases need to be automated. BioExcel will not try to generate new environments; instead those already used in the different areas will be leveraged. In particular, partners' expertise in environments such as Taverna, Galaxy or pyCOMPs (discussed in section 2.2.2) will be very valuable to implement automated versions of the analyzed workflows. First workflow prototypes will be generated in the upcoming tasks, starting from popular building blocks (discussed in section 5.1), covering functionalities that are common to the above use cases.

Interoperability: Pilot use cases workflows are built from a large variety of tools running in different infrastructures: Virtual machines, Docker containers, web portals, command-line, scripting languages, etc. (see section 2.2.1). Interconnection and interoperability between the set of tools and infrastructures should be addressed. Additionally, access to core bioinformatics databases is required for all of the use cases. Although such data is available (EBI, IRB and the appropriate interfaces exists, in the simulation world there is little tradition in using such type of interface (RESTful HTTP–based) in an automated way, as required by automated workflow enactment. This point is discussed in section 6.2 of the initial roadmap.

Missing functionalities: Although the set of available tools managed by BioExcel partners (Tables 2.2.*) covers most of the needs of selected use cases, there are still specific steps that are missing. Examples are:

- Tools for comparative modelling of protein structures.
- Automatic services to parameterize ligands or substrates for using them in Molecular Dynamics simulations and in Docking as well
- Mechanism to obtain an enhanced sampling of protein structures.
- Workflow manager for the complete pipeline.

Different requirements of user expertise: Although the available tools cover a large scope of research procedures, they were initially designed mainly for expert users. The analysis of the initial surveys reveals, however, a wider range of expertise. Questions raised like "**tools for checking for errors in inputs files**", "**transfer of data between different software modules"** indicates the need of guidance for less expert users. Some of the tools on Tables 2.2.3 and 2.2.4 have already a web-based interface with specific procedures for the newbies, but this is not the general case.

6 Initial roadmap proposal

The roadmap for the initial setup of BioExcel infrastructure will consist in the deployment of a set of common software blocks to perform most commonly demanded operations, as gathered from Use Case analysis. This will be a bottom-up building approach starting from the individual operation already available (see Section 2.2) to lead to "transversal workflow units", higher level operations that were considered general needs for the different use cases. Although the basic functionality is available, and stable, building such units will require to solve interoperability issues and deploy them in the selected software infrastructures, and eventually to set up workflow managers. To this end, the Cloud infrastructure available at the Barcelona Supercomputing Center (BSC) will come in hand (see section 6.1 for technical details). BSC's cloud will be used for initial deployment, verification, and testing, before tool made available at BioExcel's portal (section 6.3.1). Additionally, proof-of-concept complete worflows, like for instance, the generation of a MD ensemble for a series of known protein variants, or virtual-screening analysis, performed as extension of Open PHACTS queries. Lessons learned during this initial roadmap, will be applied to the integration of the remaining BioExcel tools.

6.1 BSC Cloud infrastructure

BSC (Barcelona Supercomputing Center) is the National Supercomputing Facility in Spain and was officially constituted in April 2005. BSC manages **MareNostrum**, one of the most powerful supercomputers in Europe. The mission of BSC is to investigate, develop and manage information technology in order to facilitate scientific progress. With this aim, special dedication has been taken to areas such as Computer Sciences, Life Sciences, Earth Sciences and Computational Applications in Science and Engineering.

BSC provides a cloud infrastructure to perform small-scale analysis. The main characteristics of the platform are:

1. A virtualization system, based in *OpenNebula*, to control the underlying hardware infrastructure. Applications are run in virtual machines that are instantiated dynamically following the requirements of the analysis workflow.

2. Workflows are defined by the use of COMPSs programming model. COMPSs is able to discover implicit parallelism in the pipelines, and hence, execute otherwise serial operations with an optimal use of a parallel environment. COMPSs workflows can be defined using Java, C++, or Python. COMPSs has been adapted to control the virtualization layer, making it transparent to the user, and also allowing to execute the same workflow in a series of environment, from single workstations, to HPC or grid/cloud facilities.

3. Applications where the use of COMPSs would not be advisable can be also executed in their native environment, exploiting already existing parallelism if available.

4. Complex applications are stored in the system as a collection of pre-packed virtual machines that include the application itself and the necessary software environment. Virtual machines developed here are fully compatible with most common cloud infrastructures, and EGI.

5. Access to the system is made through the Programming Model Enacting Service (PMES). PMES offers as a Basic Execution Service (BES) web service, accessible through WS clients (like Taverna), and also through a Java API.

6. A Web based tool (The Dashboard) allow for a full control of the infrastructure. The Dashboard is useful for small analysis and for development.

A Galaxy interface interacting with BES web service allows to integrate the infrastructure applications in Galaxy workflows.

6.2 Transversal workflow units

Following the initial analysis of the use cases (UCs) and transversal units, we have identified a series of operations that appear in several of the cases (Table 6.2). The initial work in Tasks 2.1 and 2.2 will be to generate integrated workflows covering such functionality and deploy them in the most appropriate environments. Components for such workflows will be adapted from the existing tools and orchestrated as interoperable building blocks, to be used in the chosen environment.

Table. 6.2 - Proposal of operations in the initial roadmap

- Remote Data Access (PDB, Uniprot, Ensembl) (UC2-4, UC7,PMuts, EDock)
- Homology modelling (UC4, PMuts)
- Ligand parameterization (UC4, UC7, EDock), using Antechamber, ACPYPE
- Basic docking (UC4, UC7, EDock)
- Simulation Setup (UC2-4, UC7,PMuts, EDock)
- MD Simulation (UC2-4, UC7,PMuts, EDock)

6.3 Deployment, Verification and Benchmark

6.3.1 A deployment portal architecture

The **BioExcel** project will use a portal to facilitate access to the software being supported within the project. The portal will also support deployment of the software (if available in a compatible repository) onto cloud platforms (if the cloud platform is supported). Where the software is already deployed the portal will provide direct access to the portal or the HPC platform – subject to any local access restrictions.

- **ELIXIR Authentication and Authorization Infrastructure (AAI)**: will be used to provide authentication to the portal, which will be built using an enterprise software platform. The ELIXIR AAI provides a source of identities, which can be endorsed with other attributes (e.g. ORCID, group information, home institution) that can be released to other service providers.

- **ELIXIR Tools and Services Registry**: The ELIXIR Excelerate project has established a tools and services registry (https://bio.tools/), which is being used across the European computational biology community to provide a browsable source of software and services. The ELIXIR registry will be used as a directory where **BioExcel** applications will be listed and properly tagged. The registry will host information about the software, the URL where instances of the software may be available for use, where test application data may be located, etc. In order to store applications (either as software or as virtual machine or container based appliances) we will use the EGI Applications Database (EGI AppDB - https://appdb.egi.eu/). It is the responsibility of the deployment portal to obtain the relevant information from the ELIXIR registry in order to be able to obtain the application, when needed, from the EGI AppDB, or other supported source.

- **EGI Applications Database**: The European Grid Infrastructure includes an application database that can host different flavors of virtual machines. In **BioExcel** we will make use of this infrastructure in order to host those workflow units that must be stored as such. Therefore, the portal will pull images from EGI AppDB in order to deploy them in the cloud infrastructure providers.

- **Cloud Service Providers**: One of the main objectives of the deployment portal is for the user to be able to seamlessly deploy workflow units into different cloud providers. We will start by using OpenStack based EMBL-EBI Embassy cloud and Amazon Web Services, but other providers will be considered as well.

  We will also monitor the developments and solutions offered by the INDIGO-Datacloud H2020 project (https://www.indigo-datacloud.eu), which develops an open source data and computing platform targeted at scientific communities, deployable on multiple hardware and provisioned over hybrid, private or public, e-infrastructures.

We foresee that a given user will be able to deploy into their choice of cloud provider provided they are authorized to do so. A user will either have to provide the appropriate credentials needed to access their selected cloud tenancy, or the user will need to be authorized to access a cloud tenancy shared across multiple users within the portal. Here again for the implementation of the authentication process, we will monitor and adopt, when suitable, solutions from the Authentication and Authorisation for Research and Collaboration (AARC, http://aarc-project.eu), which gathers partners representing all compute solutions we might build upon (HPC, HTC and Cloud).

The portal will orchestrate these systems in order to allow the deployment of **BioExcel** workflow units. There are four different components that make the portal able to perform this job:

- A set of **deployment tools**, based on technologies such as Terraform and Ansible. These are in charge of provisioning and starting the workflow units (virtual machines at first, Docker containers will be considered later on), as well as destroying them.

- A **RESTful web API** will provide access to the functionality available within the portal to define new workflow units, deploy them by using the right credentials, and verify their deployment status. The API will include the necessary databases to keep track of users, workflow units, and current deployments.

- A **rich GUI web application** using the previous RESTful web API offering a comprehensive and friendly way to deploy workflow units.

- A git server (i.e. GitHub) where workflow units will be defined as such, including the necessary Terraform and Ansible scripts, and additional data included in a manifest file for the portal to better know the application requirements.

6.3.2 Verification and Benchmarking

With the establishment of the core capability of the deployment portal, additional functionality relating to verifying a deployment on a particular cloud service provider and benchmarking the application on the cloud provider can begin.

A framework will be defined that will allow a deployed service to report on its availability to do work. These reports will be collected through the portal and reported back to the user. In the event of workflow not being successfully deployed, this can be reported back to the user, and corrective action taken.

Following a verified deployment on a particular platform a benchmarking workload will be triggered through the workflow. The results from the benchmarking workload will be used to firstly verify that the deployed workflow is operating correctly on a particular platform, and the timings will be used to benchmark a particular platform and configuration. The benchmarking results will be collected and made available for review. These benchmarking results can be used to help determine the most appropriate cloud platform and configuration for any application.

6.3.3 Roadmap

In order to establish the deployment portal and associated functionality, a series of milestones have been defined for the first project year.

Milestone 1: Get portal working, including the interaction with GitHub as a workflow unit definition repository, use http-basic for authentication, and the use of Terraform to deploy virtual machines on the EMBL-EBI Embassy Cloud and Amazon Web Services. Additionally, the portal will provide a status service to allow some initial form of deployment verification.

Milestone 2: Populating GitHub repositories and link them to ELIXIR Service and Tools Registry.

Milestone 3: Use the ELIXIR AAI as a source of identities that can be used to authenticate individuals to access the portal. Elixir AAI, as part of the AARC consortium (previously introduced), will promote integration between the various e-infrastructures and existing components based on the wishes of the research community (e.g. Elixir, EUDAT, DARIAH).

Milestone 4: Identify reference datasets, hosted at EMBL-EBI or other centre, that needs to be distributed to the relevant cloud providers to support a BioExcel workflow. The reference dataset distribution service is being developed by EMBL-EBI through the EUDAT 2020 project and will form part of the ELIXIR Compute Platform.

Milestone 5: Early adopter pilot (possibly June 2016), being deployed up to 1000 nodes. This will not be offered as a production service, but will be used to demonstrate the capability and scalability of the portal. Deployment through the portal of a Golden image from defined source (EGI AppDB).

Milestone 6: Updated early adopter pilot released for wider use within the project for early production use (End of Project Year 1)

Other milestones in later project years:

Milestone: Support for containers as first class citizen

Milestone: use Google Cloud Platform.
Milestone: Refined deployment verification framework
Milestone: Benchmarking framework with the ability to view benchmarking information by cloud service provider or application.

7 References

1. Hospital, A., et al., *MDWeb and MDMoby: an integrated web-based platform for molecular dynamics simulations.* Bioinformatics, 2012. **28**(9): p. 1278-9.
2. Hospital, A., et al., *NAFlex: a web server for the study of nucleic acid flexibility.* Nucleic Acids Res, 2013. **41**(Web Server issue): p. W47-55.
3. Fenollosa, C., et al., *SEABED: Small molEcule activity scanner weB servicE baseD.* Bioinformatics, 2015. **31**(5): p. 773-5.
4. van Zundert, G.C.P., et al., *The HADDOCK2.2 Web Server: User-Friendly Integrative Modeling of Biomolecular Complexes.* Journal of Molecular Biology, 2016. **428**(4): p. 720-725.
5. Berthold, M.R., et al., *KNIME: The Konstanz Information Miner*, in *Data Analysis, Machine Learning and Applications: Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V., Albert-Ludwigs-Universität Freiburg, March 7–9, 2007*, C. Preisach, et al., Editors. 2008, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 319-326.
6. Wolstencroft, K., et al., *The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud.* Nucleic Acids Res, 2013. **41**(Web Server issue): p. W557-61.
7. Goecks, J., et al., *Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.* Genome Biol, 2010. **11**(8): p. R86.