Grant agreement no. 675451

CompBioMed

Research and Innovation Action
H2020-EINFRA-2015-1
Topic: Centres of Excellence for Computing Applications

D5.6 Report on Efficient HPC Usage by the Biomedicine Community

Work Package:	5	
Due date of deliverable:	Month 36	
Actual submission date:	September 06 2019	
Start date of project:	October 01 2016	Duration: 36 months

Lead beneficiary for this deliverable: *SARA*
Contributors: *UEDIN, USFD, UNIGE, BSC, UOXF*

Project co-funded by the European Commission within the H2020 Programme (2014-2020)		
Dissemination Level		
PU	Public	YES
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No ***675451****"*

Table of Contents

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

List of Tables and Figures

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No ***675451****"*

1 Version Log

Version	Date	Released by	Nature of Change
V0.1	5/05/2019	Marco Verdicchio (SARA)	First Draft
V0.2	1/06/2019	Marco Verdicchio (SARA)	Major content update. Added section 6 and 7
V0.3	26/07/2019	Marco Verdicchio (SARA)	Input from co-authors added in section 7
V0.4	2/08/2019	Marco Verdicchio (SARA)	Minor corrections.
V0.5	9/08/2019	Marco Verdicchio (SARA)	Minor corrections.
V0.6	15/08/2019	Marco Verdicchio (SARA)	Addressed reviewers’ comments.
V0.7	22/08/2019	Emily Lumley (UCL)	Minor corrections.
V0.8	02/09/2019	Marco Verdicchio (SARA)	Addressed reviewers’ comments.

*“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“*

2 Contributors

Name	Institution	Role
Marco Verdicchio	SURFsara	Author
Maxime Mogé	SURFsara	Co-Author
Jazmin Aguado Sierra	BSC	Co-Author
Robin Richardson	UCL	Co-Author
David Wright	UCL	Co-Author
Andrea Townsend-Nicholson	UCL	Co-Author
Hector Martinez-Navarro	OXF	Co-Author
Victor Azizi Tarksalooyeh	UVA	Co-Author
Andrew Narracott	USFD	Co-Author
Bettine van Willigen	LTG	Co-Author
Mariano Vázquez	BSC	Reviewer
Gavin Pringle	UEDIN	Reviewer
Emily Lumley	UCL	Reviewer
Peter Coveney	UCL	Reviewer

*“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“*

3 Definition and Acronyms

Acronyms	Definitions
AWS	Amazon Web Services
BAC	Binding Affinity Calculator
BSC	Barcelona Supercomputing Centre
CFD	Computational Fluid Dynamics
CoE	Centre of Excellence
CPU	Central Processing Unit
CSCS	Swiss National Supercomputing Centre
CT-scan	Computerised Tomography scan
DICOM images	Digital Imaging and Communications in Medicine
GPU	Graphics Processing Unit
HPC	High Performance Computing
I/O	Input/Output
IaaS	Infrastructure as a Service
IB-LBM	Immersed Boundary lattice Bolzmann Method
LB	Lattice Boltzmann
LRZ	Leibniz Rechenzentrum
LTG	LifeTec Group
MD	Molecular Dynamics
MM	Molecular Mechanics
MPI	Message Passing Interface
MS	Microsoft
NCSA	National Centre of Supercomputing Applications
PaaS	Platform as a Service
pFIRE	Parallel Framework for Image Registration
QM	Quantum Mechanics
S-CAM	Sheffield Cardiac Arrhythmia Model
SaaS	Software as a Service
SSC	Student selected Component

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

STL file	STereoLithography file
UCL	University College London
UEBAS	Unified European Application Benchmark Suite
UNIGE	University of Geneva
UvA	University of Amsterdam
VTK	Visualization ToolKit
WP	Work Package
XML	eXtensible Markup Language

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

4 Executive summary

From porting of codes on High Performance Computing (HPC) resources to the development of modelling techniques and the optimisation of existing codes, the CompBioMed Centre of Excellence (CoE) has extensively supported the biomedical community to access HPC systems and efficiently use specialised software and computational tools.

This report details the work done by consortium partners with the support of HPC and Cloud resources. We present a summary of the main HPC systems employed during CompBioMed and the main access channels used to obtain an allocation on these resources. The analysis then focuses on the usage and development on four of the largest codes developed within the CompBioMed partners: Alya, HemeLB, HemoCell and BAC, as well as on other research activities within the three CompBioMed biomedical domains (Cardiovascular, Molecularly-based, Neuro-musculoskeletal medicine) associated with the use of HPC resources.

This document clearly shows the uptake of biomedical applications on various e-infrastructures and the use of community applications on HPC systems facilitated by this project.

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

5 Introduction

Despite the use of High Performance Computing (HPC) systems being routine in physical science and engineering nowadays, users experience in engaging with such resources is still progressing slowly and the expertise required for an efficient use of such systems, are usually confined to computational specialised research groups or HPC centres.

One of the main goals of the CompBioMed Centre of Excellence (CoE), is to advance the progress of biomedicine through the use of high-fidelity computational modelling and simulations. Since the beginning of the project, CompBioMed has provided support to biomedical researchers wishing to run their applications on HPC resources, advising on suitable parallel architectures and algorithm design and mapping application requirements to the characteristics of current computational and storage architectures. The efficient use of HPC resources has been at the centre of the CompBioMed project and we have continuously worked to identify the needs of our user community and promote the use of biomedical HPC codes and workflows.

In this deliverable we present an overview of the collaborative activities within the CompBioMed community focusing on the usage of HPC resources. The document focuses on three main topics:

- HPC systems accessed by the CompBioMed partners;
- Simulations code and tools used on HPC infrastructures;
- Efficient use of HPC resources by community codes.

In Section 6 we present a summary of the main resources accessed by the core partners of the project, and the access programmes used to obtain the allocations on these HPC systems.

Section 7 focuses on the services that have been used in the project, describing the technical characteristics of the main community applications and how they have been used and tested on some of the largest supercomputers in the world. In this Section we also analyse the impact of the use of HPC resources for development of efficient computational services and the production of high-level biomedical research.

This analysis aims to demonstrate the uptake of biomedicine applications on various HPCs and describes the use of community applications on HPC systems facilitated by our CoE.

This work is the result of the activities carried out in CompBioMed WP5 within the following two tasks.

Task 5.3: Efficient Infrastructure Usage by Community Codes
Leader: SARA; Partners: UEDIN, BSC, USFD
In task 5.3 we will act in 'responsive mode' to the biomedical research community outside the core project partnership, and with reference to the analysis conducted in task 5.1, to provide support to biomedical researchers wishing to run their applications on HPC resources. Support will take the form of advice on parallel architectures and algorithm design, to mapping application requirements in the biomedical community to characteristics of current computational and storage architectures and providing best practise recommendations for efficiently mapping biomedical applications to production resources, and using them. Where relevant, and in collaboration with task 6.3, we will also assess the benefits of porting

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

applications to new architectures, such as GPUs and Xeon Phi, and assess the need to establish fault tolerance and energy efficiency. We will involve a wide range of resources, including capability computing, capacity computing, cloud and Hadoop platforms.

Task 5.4: Increasing Uptake of Existing e-infrastructures and Alignment with International Projects
Leader: SARA; Partners: UEDIN, UPF, EVO
Here we will work with the scientific user community of this CoE to identify usage and application requirements which could be taken to the next level of computing resources within Europe. We will provide support for biomedical applications to obtain access to PRACE Tier-0 and Tier-1 systems, relying on results from Task 5.3 and focusing especially on applications which have previously been unable to operate at these scales. We will initiate and maintain relations with European infrastructures such as PRACE, EGI, EUDAT & EU-T0, through the partners of this CoE, with the purpose to assess and improve the applicability of these European infrastructure projects to Biomedical applications and, in cooperation with task 5.2, to establish access mechanisms and application modes that fulfil the requirements of the community.

This work is also the result of work done in collaboration with our CoE's WP2 and WP6, to both strengthen and diversify the presence of biomedical applications in the HPC landscape. The project, indeed, produced two other deliverables which complement the information presented here, namely:

- D2.4 Report on the Impact of Modelling and Simulation within Biomedical Research as Enabled by CompBioMed [1]

- D6.6 Final Report on the End-User Solutions [1]

These two deliverables, produced by WP2 and WP6 respectively, extend the description of the CompBioMed services described here, instead focusing on the end-users engagement and the scientific impact of the research performed within the project.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

6 Community HPC access

Access to the right resources can still be problematic for users, especially if not experienced in using HPC systems. During CompBioMed we have worked extensively to help community end-users gain access to HPC resources appropriate for their current work. On the one side we assist our users to adopt and deploy biomedical software and tools on HPC systems and promote the use of efficient compute services. On the other hand, we help consortium partners to access directly the HPC resources offered through the project by CompBioMed Core Partners, namely BSC, EPCC (UEDIN), and SURFsara.

In this Section we describe the type of HPC allocations (Section 6.1) and the HPC system accessed (Section 6.2) and employed by CompBioMed partners, both Core and Associate Partners, since the start of this CoE.

6.1 CompBioMed High Performance Computing Allocations

Since the beginning of our CoE, CompBioMed has been granting allocations on several large scale HPC resources to support the work of the project. Within the project we have offered, to the consortium partners, both Core and Associate Partners, the opportunity to access some of the systems managed by CompBioMed organisations (BSC, EPCC (UEDIN), and SURFsara) and obtain direct support from the HPC experts at each site. This has been crucial for partners to efficiently use the resources and employ the queue systems on board the supercomputers. The access to these resources is managed centrally, through the CompBioMed HPC allocations service [2], where anyone wishing to make use of an allocation on one or more of the HPC machines is required to apply to with a brief description of the code used, the reasons for accessing a supercomputer and their experience with HPC systems. The service has provided allocations for a total of about 10M core/hours, which have been essential for the development and optimisation of CompBioMed services. A technical description of the systems accessed can be found in Section 6.2.

Throughout the duration of the project, researchers within CompBioMed were also able to obtain access to HPC resources through national (NWO[NL] [3], EPSRC[UK] [4], RES[ES] [5], etc.), European (PRACE [6]) and extra-European (DOE INCITE [US] [7]) allocation schemes. PRACE, within the 2019 Project access call, has also provided a community access scheme, which reserved the 0.5% of the total allocation available specifically to Centres of Excellence (CoE).

6.2 Systems accessed by the project’s partners

In this Subsection we present a description of the HPC systems exploited within CompBioMed. The different systems are grouped based the type of access channel CompBioMed users have to obtain an allocation.

6.2.1 CompBioMed’s Core Partners’ HPC systems

These systems are accessible through the CompBioMed HPC allocations service and are managed by Core Partners of the project. CompBioMed users, in addition to access to the resources, obtain specialised support and training on how to port and use their applications on the target system.

*“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“*

6.2.1.1 EPCC (ARCHER, Cirrus)

EPCC, the supercomputing centre of the University of Edinburgh, provides a wide variety of services to both industry and academia including: HPC application design, development and re-engineering; HPC application performance optimisation; distributed computing consultancy and solutions; HPC facilities access; project management for software development; and data integration and data mining consultancy. Within the HPC systems hosted and managed by EPCC, CompBioMed users have access to two HPC platforms, namely ARCHER, a Tier-1 platform, and Cirrus, a Tier-2 platform.

System name	ARCHER	
Website	http://www.archer.ac.uk/	
Documentation	http://archer.ac.uk/documentation/	
Manufacturer	Cray	
Standard Compute nodes	**Configuration:**	2 x Intel Xeon E5-2697 v2 CPUs
	Cores/CPU:	12 (hyperthreading on)
	Memory:	64 GB
	Aggregate:	4544
High-memory Compute nodes	**Configuration:**	2 x Intel Xeon E5-2697 v2
	Memory:	128 GB
	Aggregate:	376
Interconnect	Aries interconnect	

System name	Cirrus	
Website	http://www.cirrus.ac.uk/	
Documentation	https://cirrus.readthedocs.io/en/master/	
Manufacturer	HPE	
Standard Compute nodes	**Configuration:**	2 x Intel Xeon E5-2695 CPUs
	Cores/CPU:	18 (hyperthreading on)
	Memory:	64GB
	Aggregate:	280
GPU Compute nodes	**Configuration:**	2 x Intel Xeon Gold 6148 CPUs 4 x NVIDIA Tesla V100-PCIE-16GB GPU accelerators
	Cores/CPU:	20 (hyperthreading on)
	Memory:	384GB
	Aggregate:	2
Interconnect	Single InfiniBand fabric	

6.2.1.2 BSC (Marenostrum IV)

BSC is the Spanish national supercomputing facility. Its mission is to research, develop and manage information technologies in order to facilitate scientific progress by combining expertise

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

covering a wide range of HPC expertise. BSC hosts Marenostrum IV, one of the Tier-0 systems accessed by the CompBioMed community. The system is composed of two “blocks” of compute nodes, General purpose and Emerging technologies. The latter is not included in this report, as it is not part of the CompBioMed HPC resources portfolio.

System name	Marenostrum IV	
Website	https://www.bsc.es/marenostrum/marenostrum	
Documentation	https://www.bsc.es/user-support/mn4.php	
Manufacturer	Lenovo	
Standard Compute nodes	**Configuration:**	2 x Intel Xeon Platinum 8160 CPUs
	Cores/CPU:	24
	Memory:	96 GB
	Aggregate:	3220
High-memory Compute nodes	**Configuration:**	2 x Intel Xeon Platinum 8160 CPUs
	Cores/CPU:	24
	Memory:	184
	Aggregate:	216
Interconnect	Intel Omni-path	

6.2.1.3 SURFsara (Cartesius, Lisa)

SURFsara is the national supercomputing and e-Science support centre in the Netherlands. SURFsara provides expertise and services in the areas of High Performance Computing, e-Science & Cloud Services, Data Services, Network support, and Visualisation. SURFsara hosts the national e-infrastructure services, i.e. the Dutch national supercomputer service Cartesius and the national compute cluster Lisa, both accessible to CompBioMed users.

System name	Cartesius	
Website	https://www.surf.nl/en/dutch-national-supercomputer-cartesius	
Documentation	https://userinfo.surfsara.nl/systems/cartesius	
Manufacturer	BULL/ATOS	
Haswell Compute nodes	**Configuration:**	2 x Intel Xeon E5-2690 v3 CPUs
	Cores/CPU:	12
	Memory:	64 GB
	Aggregate:	1080
Ivy Compute nodes	**Configuration:**	2 x Intel Xeon E5-2695 v2 CPUs
	Cores/CPU:	12
	Memory:	64 GB
	Aggregate:	540
Broadwell	**Configuration:**	2 x Intel Xeon E5-2697A v4 CPUs

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“

Compute nodes	**Cores/CPU:**	16
	Memory:	64GB
	Aggregate:	177
GPU Compute nodes	**Configuration:**	2 x Intel Xeon E5-2450 v2 CPUs 2 x NVIDIA Tesla K40m GPU accelerators
	Cores/CPU:	8
	Memory:	96 GB
	Aggregate:	64
High-memory Compute nodes	**Configuration:**	2 x Intel E5-4650 CPUs
	Cores/CPU:	16
	Memory:	256 GB
	Aggregate:	32
Interconnect	InfiniBand FDR	

System name	Lisa	
Website	https://www.surf.nl/en/lisa-compute-cluster-extra-processing-power-for-research	
Documentation	https://userinfo.surfsara.nl/systems/lisa	
Gold Compute nodes	**Configuration:**	1 x Intel Xeon Gold 6130 CPUs
	Cores/CPU:	16
	Memory:	96 GB
	Aggregate:	192
Silver Compute nodes	**Configuration:**	2 x Intel Xeon Silver 4110 CPUs
	Cores/CPU:	16
	Memory:	96 GB
	Aggregate:	96
GPU Compute nodes	**Configuration:**	2 x Intel Xeon Bronze 3104 CPUs 4 x GeForce 1080Ti GPU accelerators
	Cores/CPU:	6
	Memory:	256 GB
	Aggregate:	23
Interconnect	40 Gigabit ethernet connection	

6.2.2 Local HPC systems at partners' sites

Here we discuss systems that are usually managed locally at the university or research institute and are mainly accessed for development and prototyping of new services. Usually these systems guarantee quick access but are limited in compute power and services (e.g. GPUs, High-

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

memory nodes, etc.). On these systems access to the resources may be exclusive to users affiliated to the host institute.

6.2.2.1 ShARC

ShARC is the University of Sheffield's latest central HPC cluster.
The system consists of 124 standard compute nodes and 4 high-memory nodes, with 64GB and 256GB of memory respectively. Each node has two CPUs and each CPU has 8 cores, giving a total of 1,536 cores. The ShARC services also offers access to 8 nodes equipped with GPU accelerators.

Website: https://www.sheffield.ac.uk/cics/research/hpc/sharc
CompBioMed partner(s) accessing the system: USFD

6.2.2.2 Baobab

Baobab is an HPC cluster available to all researchers of the University of Geneva. The system is equipped with about 2,100 cores with different compute nodes equipped with memories of 64GB, 58GB or 512GB of RAM.

Website: https://plone.unige.ch/distic/pub/hpc/baobab_en
CompBioMed partner(s) accessing the system: UNIGE

6.2.2.3 Genji

Genji is a private cluster owned by BULL/ATOS. Composed of dual socket system equipped with 2 Intel Xeon Gold 6148 CPUs, with a total of 40 cores. It has been used for the benchmark and profiling of CompBioMed applications.

CompBioMed partner(s) accessing the system: BULL/ATOS

6.2.2.4 Research Computing Platforms (UCL)

The Research Computing Platform provide access to three computing clusters: Legion, Myriad, and Grace. All UCL researchers are eligible to use these platforms on a fair share basis and at no cost.

Website: https://wiki.rc.ucl.ac.uk/wiki/RC_Systems
CompBioMed partner(s) accessing the system: UCL

6.2.3 External HPC systems

HPC systems managed by institutes outside of the CompBioMed Core Partners and accessed through national or international compute calls (e.g.: PRACE, DOE INCITE, etc.). Table 1 below, lists the main external HPC systems accessed within the CompBioMed community and the project partner(s) who used the allocation.

*“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“*

Table 1. List of external HPC systems accessed by CompBioMed partners.

Name	Location	Site	CBM partner using it	Website
Piz Daint	CH	CSCS	UNIGE, BSC	https://www.cscs.ch/computers/piz-daint/
SuperMUC SuperMUC-NG	DE	LRZ	UCL, JAN, BSC, UvA, OXF	https://www.lrz.de/services/compute/supermuc/ https://doku.lrz.de/display/PUBLIC/SuperMUC-NG
JADE	UK	STFC	UCL	https://www.jade.ac.uk/
Prometheus	PL	CYFRONET	UCL, UvA	http://www.cyfronet.krakow.pl/computers/15226,artykul,prometheus.html
Summit	US	ORNL	UCL, JAN	https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/
Titan	US	ORNL	UCL, JAN	https://www.olcf.ornl.gov/olcf-resources/compute-systems/titan/
Blue Waters	US	NCSA	UCL, BSC	https://bluewaters.ncsa.illinois.edu/blue-waters-overview
Mira	US	ANL	BSC	https://www.alcf.anl.gov/mira
Anton	US	PSC	UCL	https://www.psc.edu/resources/computing/anton

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

7 CompBioMed HPC usage

In this section we present an overview of the usage of HPC infrastructures within the project. The analysis includes descriptions of the technical characteristics of the flagship CompBioMed HPC applications, which are able to scale to thousands of cores, as well as the use made by consortium partners of HPC systems for the development of services or new biomedical applications.

7.1 CompBioMed HPC applications towards exascale

Between the services developed and maintained by the CompBioMed partners, selected computational workflows are actively developed and optimised for the emerging exascale systems. Alya, HemeLB, HemoCell, Palabos and BAC are among the main HPC applications developed within CompBioMed and are routinely used on some of the largest HPC systems in the world. These codes showed excellent parallel performances and they have been the focus of many software optimisation and porting activities. All five codes described here have been used for the production of the CompBioMed's "Virtual Humans" film [8].

Here we provide a summary of the main characteristics of these codes and their use on HPC machines. The following sections describe technical requirements to deploy and use the software on HPC machines as well as an analysis of their performances on large systems and the optimisation work done within CompBioMed or in collaboration with other CoEs and H2020 projects.

7.1.1 Alya

7.1.1.1 Code description

Alya is a multi-scale, multi-physics simulation code developed by the team co-lead by Mariano Vazquez and Guillaume Houzeaux at the Barcelona Supercomputing Centre (BSC). The simulations involve the solution of Partial Differential Equations in an unstructured mesh using Finite Elements methods. In the context of CompBioMed the code has been used for coupled cardiac fluid-electro-mechanical simulations, from ion-channel kinetics up to organ level; and simulations of the respiratory system, particularly focusing on particle deposition. Alya is available for use to researchers at several HPC systems; for clinical and industrial users, BSC recommends users access it as a service, due to the complexity involved with setting up simulations. To this purpose BSC has launched a spin-off (ELEM Biotech [9]) that will provide commercial software-as-a-service to medical device, pharmaceutical and biomedical industries using Alya.

7.1.1.2 Technical specification and requirements

The code is written in modern Fortran and C, and its parallelisation is based on mesh partitioning, with MPI as the message passing library for inter-node and tasks level parallelism. In order to improve the efficiency on multi-core shared memory systems, some heavy weight loops are parallelized using OpenMP [10]. Both MPI and OpenMP layers can be used at the same time in a hybrid scheme. GPU acceleration is available through the use of OpenACC [11] directives or CUDA API [12], offloading some specific parts of the code, such as some matrix assembly loops or some types of solvers. Alya is also capable of actively using multi-code coupling on top of MPI / OpenMP parallelisation.

To build Alya from source, the following software is needed:

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

- GNU make [13]
- Fortran and C compilers
- MPI
- METIS [14] (optional)
- HDF5 [15](optional)
- VTK [16](optional)
- CUDA (optional)

The code is highly portable and compatible with most available compilers including GCC, Intel, Cray and IBM XL compilers. There are no specific requirements on type and version of the compiler used, but OpenMP support and vectorisation capabilities can improve drastically the performances of the code. Alya is compatible with variety of MPI implementations, including Intel MPI and OpenMPI, as well as bespoke libraries for specific hardware such as the Cray-MPI library. The external library METIS is mainly used for domain decomposition at the MPI level. A compatible version of the library is shipped with the source to reduce dependencies and facilitate installation procedure avoiding version conflicts. Input data is generally composed of text files with a set format. Input files can be converted into binary inputs within Alya. Output files are post processed using tools within Alya to generate standard format files, like the widely used so-called ensight format that can be visualized with software visualization tools as the open-source viewer Paraview [17].

7.1.1.3 Alya on High Performance Computing systems

Alya is currently installed and available to research users on Marenostrum IV (BSC), Cartesius (SURFsara), and ARCHER (EPCC, UEDIN). Alya has been specifically optimised for the efficient use of supercomputing resources. Combining the use of multiple level of parallelism (MPI, OpenMP, OpenACC, etc.) and efficient dynamic load balance techniques, the code is able to exploit both the full power current chips and the large number of compute elements on modern supercomputers. The code is one of the two CFD codes of the Unified European Applications Benchmark Suite (UEBAS) as well as the Accelerator benchmark suite of PRACE. Alya has been tested by the BSC team, on a large number of Tier-0 and Tier-1 HPC systems. Figure 1 and Figure 2, extracted from the work of Vasquez et al. [18], show Alya's performance on the Blue Waters (NCSA) system, for two different biomedical problems (incompressible flow in the respiratory system and non-linear solid mechanics coupled with electro-physiology of the human heart) with different mesh resolutions. The results show an almost linear scalability up to a hundred thousand cores for the problem with larger mesh size and excellent scalability, with a 0.8 efficiency maintained up to 30K and 60K cores for respiratory and cardiac problem respectively.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

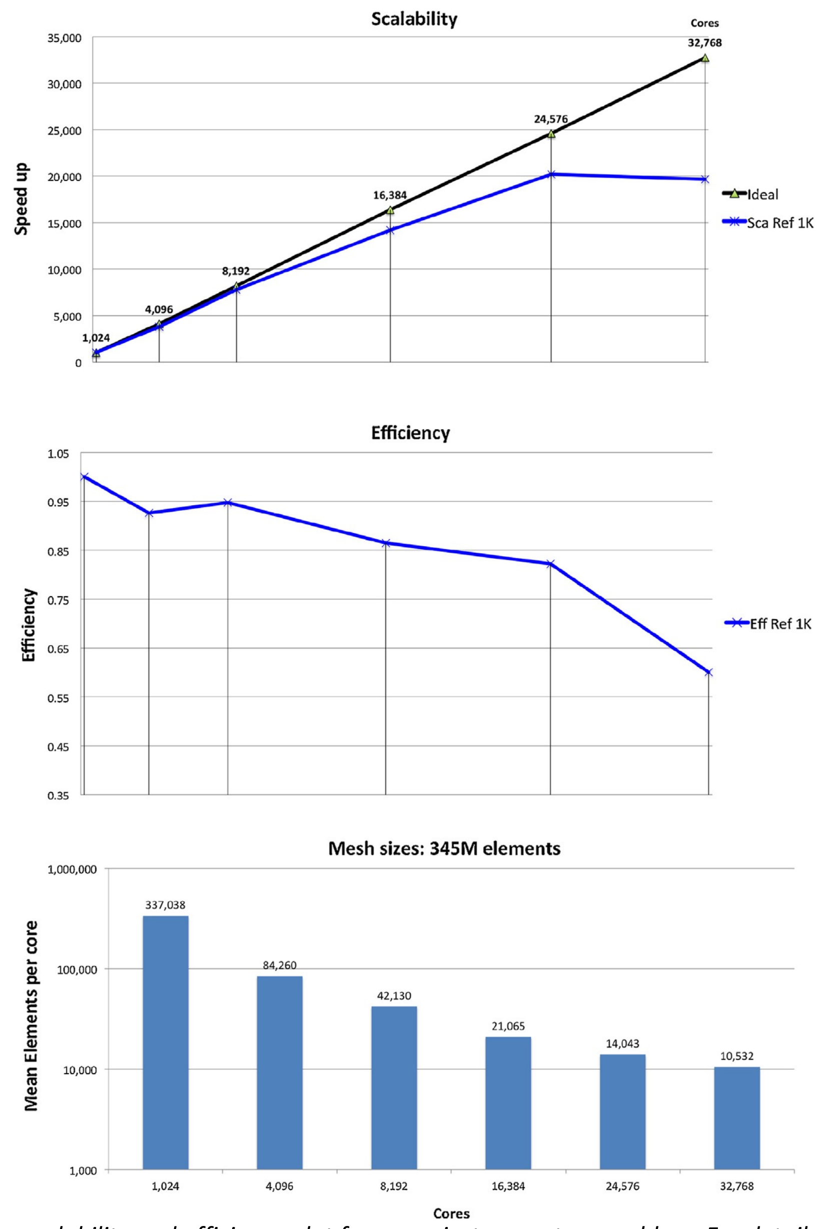


Figure 1. Strong scalability and efficiency plot for a respiratory system problem. For details see Vazquez et al. **[18]**.

Within the activities of CompBioMed, Alya has been installed on the two Tier-1 partners' systems Cartesius (SURFsara) and ARCHER (EPCC, UEDIN) and the code has been used within the CompBioMed HPC allocations service by researchers from Barcelona Supercomputing Centre and University of Oxford. The compute time provided was crucial to create highly computationally demanding simulations of human haemodynamics within the heart and to study the fluid structure interaction problem in active tissue contraction [18], [19], [20], [21]. A summary of the typical usage of the code on HPC system for CompBioMed users can be found in Table 2.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No ***675451****"*

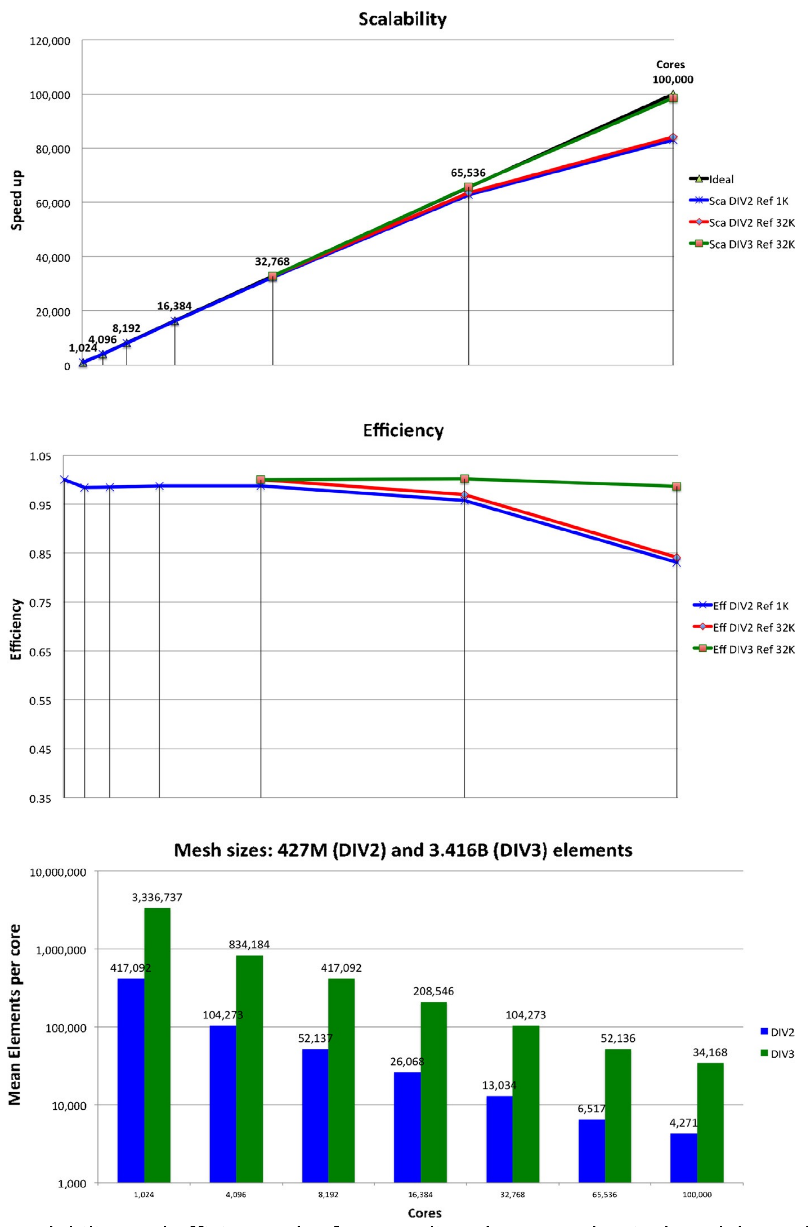


Figure 2. Strong scalability and efficiency plot for a cardiac electromechanical model. For details see Vazquez et al. **[18]**.

On Cartesius, Alya has been compiled by BSC and SURFsara personnel with different optimization flags until the maximum performance was obtained. After that a scalability test with a 100M (40M Solid, 60M Fluid) elements mesh was done. Finally, the bi-ventricular and full heart simulations where computed.

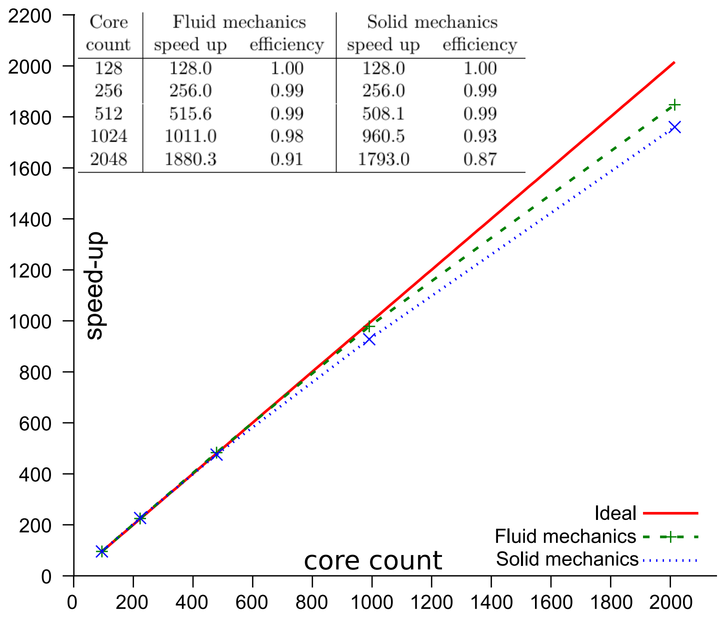
Table 2. Alya typical HPC usage within the CompBioMed community.

Mode of operation	One single extreme parallel run for each problem required.	
Type of parallelism	MPI, OpenMP	
Number of cores per run	**Typical run**	200-500
	Large run	>1,000
	Typical run	0

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

Number of GPUs per run	**Large run**	0
Input data	**Format**	Own format, HDF5, VTK
	Coming from	Mesh generators tools (e.g. gmsh, GiD, ANSA)
	Disk use	100 MB – 1GB
Output data	**Format**	Own format, HDF5, VTK, Ensight
	Used by	Data analysis and visualisation tools (e.g. Paraview, Visit)
	Disk use	> 1GB

Figure 3 shows the strong scaling analysis obtained on Cartesius (SURFsara). The results have been obtained by measuring the speed-up of the uncoupled fluid and solid mechanics simulations for an increasing number of processing elements. For optimised installation an efficiency above 87% was obtained for both models up to 2,000 cores.



Core count	Fluid mechanics speed up	Fluid mechanics efficiency	Solid mechanics speed up	Solid mechanics efficiency
128	128.0	1.00	128.0	1.00
256	256.0	0.99	256.0	0.99
512	515.6	0.99	508.1	0.99
1024	1011.0	0.98	960.5	0.93
2048	1880.3	0.91	1793.0	0.87

Figure 3. Scalability of Alya on Cartesius system, for an uncoupled problem. Solid mesh: 40M elements. Fluid mesh: 60M elements.

Alya is one of the simulation codes used within the 16th Project Access Call project "In silico drug trials in the beating ischaemic human heart" [22], awarded to CompBioMed partners University of Oxford and Barcelona Supercomputing Centre for 54.6 million core hours on MareNostrum.

7.1.2 HemeLB

7.1.2.1 Code description

HemeLB developed by the team of Prof Peter Coveney at University College London (UCL), is a software pipeline that simulates blood flow in vessels in the human brain, in support of clinical neurosurgery. HemeLB is specifically designed to efficiently handle sparse topologies, supports real-time visualization and remote steering of the simulation and can handle fully resolved red

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

blood cells. It runs on petascale platforms, alone and coupled to other codes. The pipeline takes as input an STL file of the surface geometry of the patient, generally obtained via segmentation of DICOM images [23] from a CT-scan, and HemeLB, the parallel lattice-Boltzmann CFD solver, then simulates the fluid flow within that geometry, using the given velocity-time profiles for each inlet. Once complete, the simulation output is analysed using the hemeXtract utility, which can produce images of cross-sectional flow, or 3D shots of wall shear stress distribution in the geometry using ParaView visualisation software. The UCL team also provide consulting to biomedical companies and clinical users for the use of the software.

7.1.2.2 Technical specification and requirements

HemeLB is an open source massively parallel lattice-Boltzmann (LB) simulation framework. The main lattice-Boltzman solver is written in C++ and its parallelisation is implemented via MPI. The HemeLB application relies on several external libraries for tasks as XML processing, domain decomposition, unit testing, and real time visualisation.

In order to compile HemeLB from source, the following software is needed:

- C and C++ compilers
- MPI implementation >= 3.0
- CMake buildiing tool
- Zlib
- Boost
- ParMETIS
- TinyXML
- CPPUnit
- CTemplate

HemeLB is compatible with most versions of GCC and Intel compilers and most modern MPI implementations. TinyXML, ParMETIS, CPPUnit, CTemplate, Boost and Zlib are provided with HemeLB sources and can be built during the HemeLB build.

The HemeLB computational pipeline take as input an STL file of the surface geometry of the patient, usually obtained through segmentation of DICOM image files from CT-scans and uses the fully parallelised Palabos' voxelizer to prepare the geometry for the LB solver. The approach is relatively rapid and with almost no user intervention needed. This is an advantage over traditional CFD methods, which rely on unstructured mesh generation procedures to obtain the discrete representation of the geometry; complex geometries tend to require high levels of user intervention and considerable CPU time to ensure mesh quality. The output of HemeLB can be analysed using the post-processing tool hemeXtract, which is distributed with the code. This utility produces images of different properties (i.e. cross-sectional flow or 3D image of the wall shear stress distribution) which can then be visualised with Paraview.

7.1.2.3 HemeLB on High Performance Computing systems

HemeLB is specifically designed to efficiently handle sparse topologies, supports real-time visualization and remote steering of the simulation and can handle fully resolved red blood cells. The code is highly scalable, already running on up to 200,000 cores on several multi-petaflop machines. The code has been installed and tested on ARCHER (EPCC, UEDIN), Cartesius (SURFsara), SuperMUC (LRZ), Prometheus (PSNC) and Blue Waters (NCSA), and it is a flagship code of the UKCOMES project [24]. HemeLB has undergone continuous development within the

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

project, and CompBioMed Core Partner UCL has worked to improve performance (e.g. memory usage and load balance) and execution times for large systems with billions of lattice sites.

Within a work in collaboration with the PoP CoE [25] and the EU funded project Compat [26], UCL has analysed the strong scaling behaviour of HemeLB for the simulation of a blood flow in the Circle of Willis cerebral arterial circle. The code has been tested on ARCHER and NCSA Blue Waters supercomputers, showing excellent performances up to hundreds of thousands of cores.

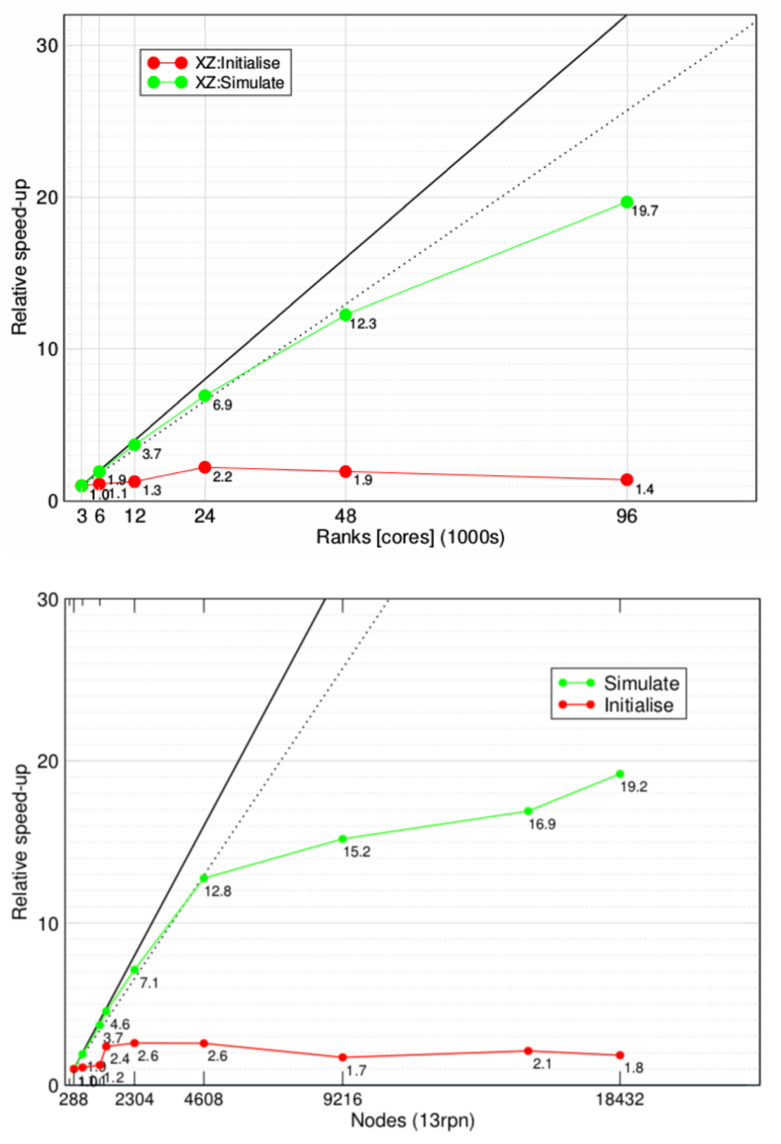


Figure 4. Strong scaling of HemeLB up to 96,000 cores on EPCC ARCHER (top) and up to 239,615 cores on NCSA Blue Waters (bottom). The plots show both initialisation (red line) and simulate phase (green line). Two different initial data datasets of 7.7 x 10^8 and 1.5x10^9 lattice sites on ARCHER and Blue Waters, respectively. Extracted from **[27]** *and* **[28]**.

Figure 4 contains the results on studies performed on the two systems: HemeLB, on ARCHER (EPCC, UEDIN) (top panel), showed excellent scalability, with about 20-fold speed-up at 96,000 cores and 80 % parallel efficiency up to 48,000 cores with respect to the reference configuration

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No ***675451****"*

run on the system (3,000 cores). On Blue Waters (NCSA) (bottom panel), scalability has been investigated from 288 up to 18,432 compute nodes. In order to fit within the memory per node on the system, only 13 of the 16 cores available per node have been used, for a total of about 240,000 MPI ranks per single simulation. The results for the simulation phase, show an 80% efficiency and speed up by a factor of 13 with 59,904 cores compared to the used baseline for this system (3744 cores). A maximum relative speed up of 19.2 was achieved with 239,615, but in this case efficiency was much lower. For this work the authors have collaborated with SCALASCA [29] developers in order to use this profiling tool at core counts over 30,000. The study, has been useful to identify bottlenecks in the use of current MPI-2 and MPI-3 implementations which, using only 32-bit communications, are inadequate when running on hundreds of thousands of cores. For this reason, HemeLB has become a "use case" for the MPI Forum in the release of MPI-4, which contains clean 64-bit communications.

UCL is currently working on coupled full human blood flow simulation which is planned to run on SuperMUC-NG (LRZ) using up to 160,000 cores which will provide a new insight into the scalability of the code, towards exascale machine. More details about this research, can be found in the WP2 deliverable D2.4 [1].

Table 3. HemeLB typical HPC usage within the CompBioMed community.

Mode of operation	One single extreme parallel run for each problem required.	
Type of parallelism	MPI	
Number of cores per run	**Typical run**	1,000-10,000
	Large run	>50,000
Number of GPUs per run	**Typical run**	0
	Large run	0
Input data	**Format**	STL (for surface geometry), XML (config file), GMY (HemeLB own format)
	Coming from	The XML and GMY are generated from the STL using the HemeLB setup tool
	Disk use	100Mb – 2GB
Output data	**Format**	XTR (HemeLB own format)
	Used by	hemeXtract will process these files into format for data analysis and visualisation tools (e.g. Paraview, Visit)
	Disk use	500MB – 50GB

7.1.3 HemoCell

7.1.3.1 Code description

HemoCell, developed by the team of Prof. Alfons Hoekstra at the University of Amsterdam (UvA), is a framework for simulating transport properties of dense suspensions of deformable cells, such as blood. The code is based on the combined Immersed boundary-lattice Boltzmann method (IB-LBM) and is built on top of the open source C++ lattice Boltzmann solver Palabos. HemoCell is currently used in a wide range of application, from modelling aggregation of platelets and thrombus formation, via white blood cell margination, to studying details of the infectious pathways in malaria.

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

Palabos is a multi-physics, open-source lattice Boltzmann code which has been developed by CompBioMed Core Partner University of Geneve (UNIGE). The code runs on many platforms, from laptop to HPC machines. It has been used for several biomedical applications, such as thrombus formation in cerebral aneurysms, vertebroplasty, transport of fully resolved red blood cells and platelets in complex geometries, and haemodynamics in stented coronary arteries. It can also be deployed for the study of neuro-musculoskeletal problems. It is now available via the scientific software company NUMECA.

7.1.3.2 Technical specification and requirements

Both HemoCell and Palabos are written in C++ and they make use of an on-demand compilation process, wherein the codes need to be compiled for each specific end-user application and then the produced executables can be re-used in future, until a new compilation is needed due to a modification of the code or compilation options. Detailed instructions on how to compile HemoCell are available at [30].

The following software and libraries are needed to compile and run HemoCell:

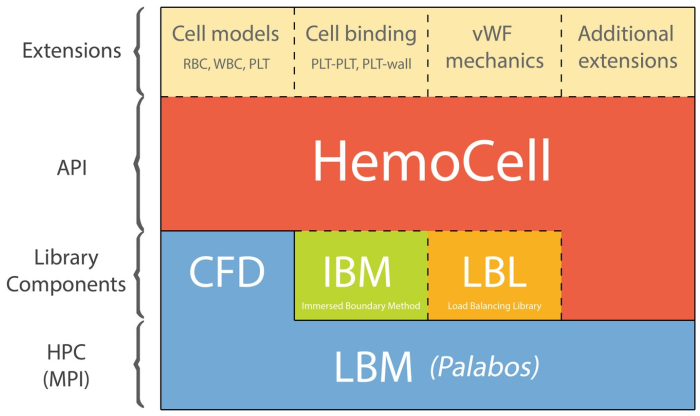
- Openmpi (≥ 1.10.2) or Intel Mpi (≥ 17.0.5)
- GCC (≥ 5.2.0) or Intel C++ compiler(≥ 17.0.5)
- CMake ≥ 3.7.2
- HDF5 ≥ 1.8.16
- GNU “patch” utility ≥ 2.7.5
- h5py ≥ 2.6.0-1
- PARMETIS 4.0.3 (Optional)
- Palabos 2.0
- Python

HemoCell uses Palabos to manage initial domain decomposition and to voxelize geometries. The outputs, after post processing with in-house developed tools, are then visualised through data visualisation software such as Paraview or Visit.

7.1.3.3 HemoCell on High Performance Computing systems

HemoCell is computationally capable of handling a large domain size with a high number of cells (10^4-10^6 cells). The code has been used and optimised to run on SURFsara systems Cartesius, Lisa, SuperMUC (LRZ) and Marenostrum IV (BSC). The code showed good performances on all the systems with good weak and strong scaling performances up to ~4,000 cores [31].
The framework uses data parallelism to distribute the workload over many compute elements, with Palabos used for the fluid phase simulation, and taking care of the boundary communications between cores, and HemoCell taking care of the cell-based flow and the communications between the two fluid and cell-based parts. The HDF5 library is used for handling I/O.

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“



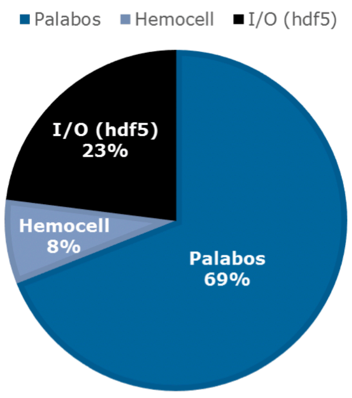


Figure 5. Overview of the Palabos and HemoCell libraries structure (right) and profile chart (left). Extracted from **[32]** *and D2.3* **[1]**.

The development and porting of the HemoCell framework is between some of the central activities carried out within WP2 and WP5 for the support of exemplar applications and extension to novel architectures. CompBioMed Core Partner BULL/ATOS (BULL) has worked in collaboration with the developers of the code, to port and optimise a dense cellular suspension flow application, based on HemoCell, to Intel Skylake microarchitecture (see: CompBioMed report D2.3 [1] for more details). This work provided insight of the main bottlenecks in the code parallelisation strategy as well as strong and weak scaling behaviour up to about 2,000 cores. The results suggested a degrading of performances, as the core count increases, due to an uneven distribution of the workload among the different MPI ranks and the relevance of data structures within the code to efficiently exploit core level vectorisation capabilities of modern CPUs.

UvA, a CompBioMed Core Partner, has been continuously working to improve load balance within the code and increase communication efficiency of the codebase. In a recent study UvA have showed how the reorganisation of internal data structures and algorithms as well as the use of optimized communications procedures, bring a noticeable improvement of the base performances as well as of the strong scaling behaviour (dividing the same domain into smaller atomic blocks).

Recently, SURFsara and UvA have been working on evaluation of best practices for the optimised porting and efficient installation of HemoCell on Cartesius (SURFsara) and ARCHER (EPCC, UEDIN) supercomputers. HemoCell-2.0 was built on the two systems, with Intel and GNU compilers, using architecture-specific optimisation flags, and the CMake configuration file was tuned to facilitate optimised installation of the software on supercomputers for all users. We are currently investigating the interoperability of the code with the Cray compiler. At the moment of writing, compilation of HemoCell with this type of compiler fails, as part of the source code seems not to conform to the C++11 standard according to the Cray compiler. Benchmarking on ARCHER and Cartesius showed a discrepancy in the performances of HemoCell compiled with the GNU compiler and with the Intel compiler. In our benchmarks (input taken from the HemoCell examples: 'hematocrit_33') on 3 nodes / 64 cores), the runtime with the Intel compiler is significantly longer than with the GNU compiler: The runtime with the Intel compiler is 19% longer than with the GNU compiler on Cartesius (SURFsara), and 20% longer on ARCHER (EPCC, UEDIN). This behaviour is under investigation at the moment.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

We also interfaced the code with Reframe [33], a new framework, developed by CSCS supercomputing centre, for running regression tests for HPC systems. The idea is to use the framework to automate the testing and benchmarking of HemoCell on Cartesius, separating the logic of a regression test from the low-level details, which pertain to the system configuration and setup.

Table 4. HemoCell typical HPC usage within the CompBioMed community.

Mode of operation	Generation of initial conditions happen on single nodes using OpenMP parallelism. The actual simulation is run on multiple nodes. A typical investigation consists of about a dozen simulations	
Type of parallelism	MPI for the simulation	
Number of cores per run	**Typical run**	100-500
	Large run	5,000 – 1,000
Number of GPUs per run	**Typical run**	0
	Large run	0
Input data	**Format**	Geometry triangulation (STL files), Initial cell positions and orientations (plain text files), simulation parameters and material model descriptors (xml files).
	Coming from	Initial condition generator which is part of the framework
	Disk use	100MB - 1GB
Output data	**Format**	HDF5 compressed arrays of the flow field, cells, forces.
	Used by	Post processing tools, Paraview, VisIt
	Disk use	>1GB

7.1.4 BAC

7.1.4.1 Code description

Binding Affinity Calculator (BAC), developed in part by UCL and partly by EnsembleMD [34], is a workflow for the automated building, execution and analysis of a diverse range of Molecular Dynamics (MD) based free energy calculations. Designed to coordinate large numbers of simulations, it currently runs on multi-petaflops and emerging exascale environments. The workflow tool runs and analyses simulations designed to assess how well drugs bind to their target proteins and the impact of changes to those proteins. The code uses ensemble simulations to robust, accurate and precise free energy computations from both alchemical and end-point analysis methodologies. BAC is a fairly complex tool to use, so at the moment the development team at UCL have made it available as part of consulting services or for research collaborations. EnsembleMD provides user-friendly interfaces to related binding affinity calculation services also available through the on-line store of associate partner DNAnexus (limited access).

7.1.4.2 Technical specification and requirements

BAC is a collection of scripts, mainly written in Python, which wrap around common MD codes to facilitate free energy calculations. Except for a recent version of the Python interpreter, there are no external requirements to install and run BAC, however the workflow execution strongly depend on the MD software package employed and its performance on the selected hardware.

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“

The main Molecular Dynamics (MD) codes used within BAC are:

- **NAMD (>2.9)** [35] is a parallel MD code designed for high-performance simulations of large biomolecular systems. The code is written in C and is based on Charm++ parallel objects. NAMD uses the Charm++ native communications layer and the program "charmrun" to launch the executable processes. NAMD scales to hundreds of cores for typical simulations and beyond 500,000 cores for the largest simulations. The code is distributed as binary, however compilation from source is necessary if we want to tune performances and use MPI (suggested for use on HPC systems).

  NAMD requirements:

  - C compiler
  - Charm++ (distributed with the code)
  - GNU utility make
  - Tcl
  - FFTW library
  - MPI library
  - CUDA library (for GPU enabled versions)

- **GROMACS (>5.1.x)** [36] written in C and C++, is a versatile package to perform MD simulations, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

  GROMACS requirements:

  - C, C++ compiler (require full C++11 support)
  - CMake
  - Tcl
  - FFTW library or Intel MKL
  - MPI library (supports the MPI 1.3 standard)
  - CUDA library (for GPU enabled versions)

- **OpenMM** [37] is a high performance toolkit for molecular simulation. It is a collection of libraries written in the Python programming language which can easily be chained together to create Python programs that run simulations. Extensive language bindings for Python, C, C++, and even Fortran are available. The code is open source and actively maintained on Github, licensed under MIT and LGPL. As an alternative to source code compilation, OpenMM is available also through the "conda" package manager.

  Requirements for building OpenMM from source:

  - A C++ compiler
  - CMake
  - CUDA library (for GPU enabled versions)

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

or

- conda
- CUDA library (for GPU enabled versions)

A summary of the typical HPC usage done within this research area can be found in Table 11.

CompBioMed's Associate Partner Rutgers has developed, in collaboration with UCL, HTBAC [38]. This framework is based on the protocols defined and automated within BAC but uses high level python object abstractions for defining simulations, physical systems and ensemble-based free energy protocols for large scale free energy binding calculations. HTBAC abstraction uses underlying building blocks middleware, based on RADICAL Pilot [39], to create and execute multiple sets of replicas, a.k.a. ensemble members, on supercomputing cyberinfrastructures.

7.1.4.3 BAC on HPC Systems

BAC computational method is based on classical MD simulations where the unit of parallelism involved is a few hundred cores, but runs hundreds to thousands of these at a time in order to perform "high throughput" calculations in support of drug discovery and personalised medicine (for advising on drug treatment for patients based on their genomic profile). In order to guarantee reproducibility of predictions, ensembles of replica MD calculations are performed for each method. Combining this approach with the necessity to screen multiple compounds, makes BAC a perfect example of replica computing pattern application, where a large number of copies of simulations are needed to produce statistically robust results. All replicas simulations, which may be parallel runs themselves, execute independently of each other, allowing BAC to reach excellent scalability also at high number of cores. It is a computational workflow which is data intensive and data driven in the sense of needing a large amount of data to initialise and produce tens or more of terabytes of output per run.

In the case of replica compute applications such as BAC, weak scaling analysis provide useful information to demonstrate the ability to solve larger systems as the resources available increase. Figure 6 shows the result of a weak scaling analysis, performed by UCL on the Blue Waters (NCSA) supercomputer. The study consisted in the screening of sixteen drug candidates concurrently using thousands of multi-stage pipelines. The results demonstrated the almost perfect scaling of the code up to 33,280 cores under the conditions tested. The overhead measured, consisted in the adaptive sampling of the method used, and small overhead from the HTBAC framework. The latter depends on the number of protocols generated and increase linearly, however, as showed in the results presented, is always negligible compared to the total time of execution. Similar scaling has been demonstrated on other platforms such as Titan (ORNL) and for different protocols.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

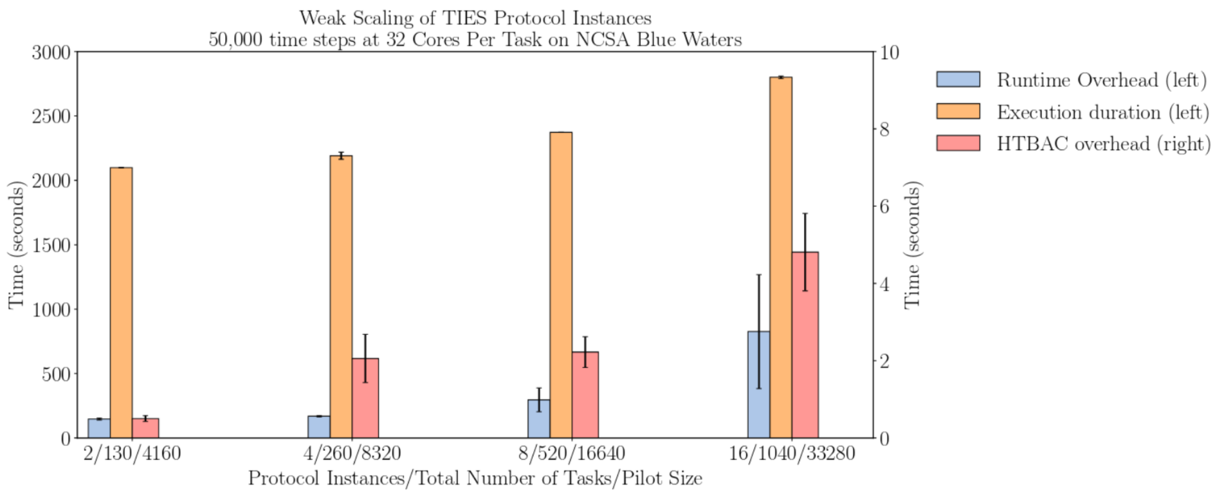


Figure 6. Weak scaling properties of HTBAC on NCSA Blue Waters for 16 candidate drugs using the TIES protocol.

UCL has ported and optimised the workflow on some of the largest multi-petascale supercomputers in the world: SuperMUC and SuperMUC-NG (LRZ), Blue Waters (NCSA) and Titan (ORNL).

BAC is one of the codes used by CompBioMed partners UCL, Janssen R&D and Rutgers within the INSPIRE DOE INCITE project, and has been awarded a major allocation on two systems, namely Titan (ORNL, 80 million core hours) in 2018 and on Summit (ORNL, 25,000 node hours), which were #1 in the top500 list at the times of the allocations.

In 2017 CompBioMed partners UCL, Janssen R&D and LRZ have been awarded a two-year project on SuperMUC (LRZ) for 28 million core hours over 2 years. The planned work concentrated on the development and application of the BAC workflow. On the same system, in June 2016, UCL ran a giant instance of BAC across phase 1 and phase 2 of the system for a total of about 240,000 cores.

UCL work to execute and manage large and complex campaigns of ligand binding calculations (using our domain specific workflow middleware, HTBAC) has won the 11th IEEE International Scalable Computing Challenge (SCALE 2018) at the IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid) 2018 held in Washington DC. The prize rewards real-world problem solving using computing that scales to very large core counts.

7.2 HPC Usage within the Biomedical Community

Besides the already described HPC codes (see Section 7.1), the CompBioMed software stack is composed of a heterogeneous and extended set of applications and computational services which run on HPC infrastructures. CompBioMed has facilitated the deployment and development of these services providing access to resources and technical support to the researchers. This section provides a summary of the different use our community has made of the available HPC resources. This section is divided in three parts:

- Cardiovascular research;
- Molecularly based research;
- Neuro-musculoskeletal research.

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

And focuses on the work done using HPC resources within the three research exemplars explored in CompBioMed, for the development of computational services and the advancement of computational biomedical research.

7.2.1 Cardiovascular research

7.2.1.1 LifeTec Group (LTG)

Using the CompBioMed HPC allocations service, LTG accessed an allocation of 100,000 core/hours on Cartesius (SURFsara) HPC system for the development of AngioSupport [40]. The application, a 1D tool for the simulation of coronary intervention, allows clinicians to perform a bypass surgery or to place a stent virtually and provide clinical decision support as a Software as a Service (SaaS). In particular the allocation was used to perform a sensitivity analysis on the 1D wave propagation model of the coronary arteries to identify the important parameters that can be coupled to specific patient data.

The code is based on the Variance Based Sensitivity Analysis of Saltelli et al. [41] and uses Python programming language to perform the analysis. In collaboration with SURFsara support team, the code was interfaced with the Slurm workload manager [42] to automate the procedure and study a large number of parameters concurrently. In particular, we used job array support and jobs dependency functionalities in Slurm, to submit and manage collections of jobs and run postprocessing analysis on the cluster within an automated workflow. The collaboration with HPC experts at the site has been beneficial for the integration of the code with HPC resources.

Table 5. LTG – HPC usage for cardiovascular research

Mode of operation	Embarrassingly parallel run.		
Software used	Python		
Type of Parallelism	serial		
Systems used	Cartesius (SURFsara)		
Hardware used	CPU		
Job size	**Small run**	10 nodes (24 cores / node)	
	Large run	20 nodes (24 cores / node)	
I/O	**Input**	Text files	1-100 MB
	Output	Text files	1-100 MB

7.2.1.2 University of Sheffield (USFD)

CompBioMed partner USFD has accessed HPC resources for the study of two biomedical cardiovascular problems:

A. A cardiac electrophysiology study of the vulnerability of arrhythmias in ventricles with diffusive scars;

B. Simulations of cardiovascular biomarkers across a virtual population.

For the first study USFD has accessed the local HPC cluster ShARC, where they used an in-house developed Finite Element Method based code (Sheffield Cardiac Arrhythmia Model (S-CAM)) to perform analysis of 20 samples in 4 different time scales. The code, written in C, uses MATLAB

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“

[43] for post processing and analysis of the results. Simulations are embarrassingly parallel where several samples for different time scales can be simulated concurrently.

In the second study a patient-generic 1D model of the cardiovascular system (openBF [44], USFD) was used to generate several models, representative of a range of individuals. OpenBF is an open-source 1D blood flow solver based on a finite-volume numerical scheme, written in Julia. [45] It uses Python and bash to manage execution and files handling.

The virtual population consisted in 50k+ individuals which could be simulated only thanks to the extreme parallelism offered by the HPC platform. A Latin Hypercube algorithm was used to ensure a homogeneous coverage of the input space with three mesh density values to ensure mesh independence, leading to a total of about 15,000 simulations. The single-core simulations, which will require more than 200 days if executed sequentially, were run with an embarrassingly parallel approach over a multi-core Virtual Machine (HPC cloud, SURFsara), a tier-1 (Cartesius, SURFsara) and a tier-3 HPC system (ShARC, USFD), allowing us to reduce the total compute time linearly with the number of cores available.

Table 6. USFD – HPC usage for cardiovascular research.

Mode of operation	Embarrassingly parallel run		
Software used	MATLAB, OpenBF, Python, S-CAM		
Type of Parallelism	serial		
Systems used	Cartesius (SURFsara), ShARC (USFD), HPC Cloud (SURFsara)		
Hardware used	CPU		
Job size	**Small run**	1 node (10 cores / node)	
	Large run	100 nodes (24 cores / node)	
I/O	**Input**	Text files	1-100 MB
	Output	Text files	1-100 MB

7.2.1.3 University of Geneva (UNIGE)

Within CompBioMed, UNIGE is using HPC resources for the development of a software for the simulation of a fully resolved 3D blood flow. The code uses Lattice Boltzmann (LB) methods for the CFD, Finite Element methods for the deformable bodies and immersed boundary methods for the fluid-structure interactions. It is written in C++ and uses Palabos (See Section 7.1.2), Eigen, CUDA libraries. Meshlab [46] is used for the generation of the 3D mesh with Python used for other pre/post processing operations.

For this work UNIGE has accessed the Baobab (UNIGE), Cartesius (SURFsara) and Piz Daint (CSCS) HPC systems, with a total resources allocation of more than 1M core/hours. Currently UNIGE is working on performance optimisation of the code and to make the developed framework a general-purpose tool for simulation of blood flow.

Table 7. UNIGE – HPC usage for cardiovascular research.

Mode of operation	Massively parallel run
Software used	Palabos, Eigen, Meshlab, Python
Type of Parallelism	MPI, CUDA
Systems used	Cartesius (SURFsara), Baobab (UNIGE), Piz Daint (CSCS)

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

Hardware used	CPU, GPU		
Job size	**Small run**	5-10 nodes (24 – 12 cores, 1 GPU / node)	
	Large run	100 nodes (24 – 12 cores, 1 GPU / node)	
I/O	**Input**	csv, xml files	100MB – 1GB
	Output	csv, xml files	>1GB

7.2.1.4 University of Oxford (UOXF)

UOXF is one of CompBioMed's Core Partners working in the field of computational cardiovascular medicine. Their use of HPC resources have been focused on the development of multiscale human models of cardiac activity, to improve drug development and the study of the impact of cardiac diseases which is not possible to investigate in clinical trials due to ethical and technical limitations.

Apart from a massive use of the aforementioned code Alya, the Oxford group uses CHASTE [47], a highly parallel cardiac electrophysiology code written in C++ and Python. Developed at UOXF, the code uses external libraries for handling I/O (XCERES [48], CellML [49], HDF5), the CVODE [50] library for linear algebra operations and it uses MPI and OpenMP for parallel execution on supercomputers. Paraview and MATLAB have been used for the post-processing and visualisation of the output files.

Oxford has used several millions of core/hours on ARCHER (EPCC), Marenostrum IV (BSC) and SuperMUC (LRZ) HPC systems. The access to these resources was obtained through the CompBioMed HPC allocations and through PRACE Project Access programme.

Table 8. OXF – HPC usage for cardiovascular research.

Mode of operation	Massively parallel run		
Software used	Alya, CHASTE, Paraview, MATLAB		
Type of Parallelism	Hybrid MPI/OpenMP		
Systems used	ARCHER (EPCC), Marenostrum 4 (BSC)		
Hardware used	CPU		
Job size	**Small run**	20-50 nodes (24 cores/node)	
	Large run	20-50 nodes (24 cores/node)	
I/O	**Input**	Mesh files	> 1GB
	Output	HDF5, VTK files	> 1GB

7.2.1.5 University of Edinburgh (UEDIN)

UEDIN, has worked on the development of PolNet [51], an open-source tool for the study of blood flow at the single-cell level in realistic microvascular networks. PolNet runs on a single workstation, which, for much of the analysis, is an appropriate platform. However, estimating the forces due to the blood flow requires running a CFD simulation which, on a single workstation, can take several days to complete. PolNet uses HemeLB for this work, allowing the code to take full advantage of HPC systems thus reducing the wall clock time for a simulation to tens of minutes.

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

Table 9. UEDIN – HPC usage for cardiovascular research.

Mode of operation	Massively parallel run		
Software used	PolNet, HemeLB, Hoff		
Type of Parallelism	MPI		
Systems used	ARCHER (EPCC), Cirrus (EPCC), Lisa (EPCC).		
Hardware used	CPU		
Job size	**Small run**	1-5 nodes (24 cores/node)	
	Large run	>50 nodes (36 - 24 cores/node)	
I/O	**Input**	Mesh files	> 1GB – 100MB
	Output	Custom binary format	> 1GB – 100MB

In order to facilitate the offload of the CFD simulation to HPC system, EPCC has designed the HemeLB High Performance Offload service (Hoff [52]). The services, developed with the support of the CompBioMed HPC allocations, has been used to allow the execution of PolNet on Cirrus (EPCC) and Lisa (SURFsara), with the plan to extend its functionalities to include commercial cloud providers. Detailed description of the Hoff and its implementation can be in found CompBioMed deliverable D5.5 [1].

7.2.2 Molecularly-based research

7.2.2.1 Janssen Pharmaceutica NV (JAN)

JAN is a CompBioMed Core Partner from industry which works on the development and implementation of advanced molecular simulations for the accurate prediction of binding free energy for small molecule drugs to protein targets. Through the CompBioMed HPC allocations service JAN obtained an allocation of 1.8M core/hours on Cartesius (SURFsara). JAN evaluated the use of non-commercial software for predicting the free energy of a series of compounds, and build a protocol which uses open-source software for MD and free energy perturbation simulations [53]. The main simulation code used was GROMACS (version 2016.1) configured to run on GPU, and the results compared to previous simulations performed with Schrodinger's FEP+ [54] commercial software. Slurm support for job array and job dependencies, has been used to orchestrate and streamline the workflow and to optimise the MD simulations execution on multiple compute nodes.

In addition, during the project Janssen has been a partner in two successful bids for HPC resources led by UCL: 32M core hours on SuperMUC (LRZ) in 2017-19, and on DoE Titan (ORNL) in an INCITE award worth around 100M core hours between 2018-2019.

Table 10. JAN – HPC usage for molecularly-based research.

Mode of operation	Multiple concurrent single node parallel simulations		
Software used	GROMACS		
Type of Parallelism	Hybrid MPI/OpenMP, CUDA		
Systems used	Cartesius (SURFsara)		
Hardware used	CPU, GPU		
Job size	**Small run**	1 node (8 cores/node)	
	Large run	>50 nodes (16 cores/node)	
I/O	**Input**	Gromacs specific files	> 1GB

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

	Output	Gromacs specific files	> 1GB

7.2.2.2 University College London (UCL)

Within CompBioMed, UCL researchers have accessed several partners' HPC systems (i.e. Cartesius (SURFsara), ARCHER (EPCC), UCL Research Computing Platforms) to study the properties of biomolecular systems using MD, quantum mechanics (QM) and molecular mechanics (MM) simulation methods. UCL has employed several software benchmarking performances and testing functionalities on the different HPC systems.

UCL has also started a collaboration with CompBioMed Core Partner Evotec UK Ltd. (EVO) using the Cartesius (SURFsara) HPC system for the study of residence time of a drug with target proteins and the quantum-mechanical study of protein-small molecule and residue-residue interactions [55], [56]. These resources have been accessed through the CompBioMed HPC allocation service.

Table 11. UCL – HPC usage for molecularly-based research.

Molecular dynamics simulations			
Mode of operation	Multiple concurrent multi node parallel simulations		
Software used	GROMACS, NAMD, AmberTools,		
Type of Parallelism	MPI, Hybrid MPI/OpenMP, CUDA		
Systems used	Cartesius (SURFsara), ARCHER (EPCC), Research Computing Platforms (UCL)		
Hardware used	CPU, GPU		
Job size	**Small run**	1 node (8 cores/node, 2 GPU/node)	
	Large run	>50 nodes (24 cores/node)	
I/O	**Input**	application specific files	1GB - 100MB
	Output	application specific files	> 1GB
Quantum mechanics simulations			
Mode of operation	Multiple concurrent single node parallel simulations		
Software used	Gamess US, NWChem		
Type of Parallelism	MPI, Hybrid MPI/OpenMP		
Systems used	ARCHER (EPCC), Research Computing Platforms (UCL)		
Hardware used	CPU		
Job size	**Small run**	1 node (8 cores/node)	
	Large run	10 nodes (24 cores/node)	
I/O	**Input**	Application specific files	1MB- 100MB
	Output	Application specific files	1GB – 100MB
Molecular mechanics simulations			
Mode of operation	Multiple concurrent single node parallel simulations		
Software used	DL_POLY, ChemShell		
Type of Parallelism	MPI, Hybrid MPI/OpenMP		
Systems used	ARCHER (EPCC)		
Hardware used	CPU		
Job size	**Small run**	1 node (8 cores/node)	

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

	Large run	10 nodes (24 cores/node)	
I/O	**Input**	Application specific files	<100MB
	Output	Application specific files	1GB – 100MB

In addition to research work, UCL has also used HPC resources within the Student Selected Component (SSC) of UCL's Medical School Curriculum. This course provides medical students with the opportunity to use state of the art laboratory and computational resources to complete a metagenomics project in molecular medicine. The training programme, created by UCL and EPCC, provided an introduction to use of HPC systems, and the use the software Qiime [57] for processing and analysis of DNA sequencing data. The course made use of the Cirrus (EPCC) and Cartesius (SURFsara) HPC systems; access to these resources was provided through the CompBioMed HPC allocation service.

7.2.2.3 Universitat Pompeu Fabra (UPF) and Acellera (ACE)

CompBioMed Core Partners UPF and ACE, have used the CompBioMed HPC allocation service, to access ARCHER (EPCC). The work involved the use of HPC resources for the development of automated force field parameterization tools for computer-aided drug discovery simulations. The main software used to perform the work was Psi4 [58], an open-source electronic structure code written in C++ and Python. The code run efficiently on shared memory machines using the multithread parallelism of the mathematical libraries configured as backend (e.g. BLAS, LAPACK, Intel MKL). HTMD [59] and Python have been used for the preparation of the simulations input and the analysis of the results.

Table 12. UPF and ACE – HPC usage for molecularly-based research.

Mode of operation	Multiple concurrent single node parallel simulations		
Software used	Psi4, HTMD, Python		
Type of Parallelism	Multithreading		
Systems used	ARCHER (EPCC)		
Hardware used	CPU		
Job size	**Small run**	1 node (12 cores/node)	
	Large run	5 nodes (12 cores/node)	
I/O	**Input**	Text files	1 - 100MB
	Output	Text files	100MB – 1GB

7.2.3 Neuro-musculoskeletal research

7.2.3.1 University of Sheffield (USFD)

USFD has made use of HPC resource, also for the development of pFIRE(The parallel Framework for Image REgistration) [60], an image registration code especially designed to handle large images.

For this reason, the code requires high-memory nodes equipped with multicore CPUs that can simply not be accommodated on local clusters. The code is written in C++ and uses the library PETSc to parallelise the workload across multiple nodes, Boost libraries for general utilities and HDF5 for I/O support. In addition, the code can be compiled with DCMTK or OpenImageIO libraries for support respectively to DICOM and general image formats (e.g. png, tiff). More

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

information on how to use and configure pFIRE can be found in the user documentation provided at [61]. USFD has used Python and MATLAB for the pre-/post- processing of output and input files.

Allocations on ARCHER (EPCC), Cirrus (EPCC) and ShARC (USFD) have been used through the CompBioMed HPC allocation service for the development and testing of the code.

Table 13. USFD – HPC usage for neuro-musculoskeletal research.

Mode of operation	Embarrassingly parallel run		
Software used	pFIRE, PETSc, Python, MATLAB		
Type of Parallelism	MPI, OpenMP		
Systems used	ARCHER (EPCC), Cirrus (EPCC), ShARC (USFD)		
Hardware used	CPU		
Job size	**Small run**	1 node (24 cores / node)	
	Large run	>50 nodes (36 cores / node)	
I/O	**Input**	Dicom or png image	>1GB – 100MB
	Output	displacement map (xdmf)	>1GB – 100MB

Within the activities of WP6, Sheffield has also developed the CT2S hospital workflow, for the analysis of bone strength based on CT images. The aim is to provide access to HPC workflows directly to typical clinical environment without exposing the complexity of the interaction with the HPC resources and software. The workflow uses ShARC (USFD) as backend computational resource to perform the analysis. Details of the workflow functionalities as well as its implementations are described in the CompBioMed deliverable D6.6 [1].

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“

8 CompBioMed Cloud Computing Usage

High Performance Computing resources, characterized by highly parallel and memory efficient applications that run on many cores, occupy only the lowest layers of the vertically integrated application stacks that the CompBioMed CoE developed. CompBioMed users require not only high compute power and efficient data storage systems, but also flexible and secure access to resources, especially within industrial and clinical set ups. Cloud Computing infrastructure provides the quality of service essential to use this type of applications, and consortium partners have worked extensively to develop tools and services capable of exploiting the full power of these resources.

In this section we provide a summary of the activities performed within CompBioMed which involved the use of Cloud computing resources. The section describes the work CompBioMed partners have done within the three types of service models provided within cloud environment: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

8.1 Infrastructure as a Service in CompBioMed

The Infrastructure as a Service (IaaS) model offers the capability to provide processing, storage, networks, and other fundamental computing resources where the user is able to deploy and run arbitrary software. Within this model, users do not manage the underlying infrastructure but enable on-demand access to resources, without the overhead of their ownership, offering the required services through the use of one or more virtual machines.

In the following a description of the IaaS provided and managed by CompBioMed core and associate partners is presented:

SURFsara HPC cloud
SURFsara offers the HPC Cloud computing facility [62] as an Infrastructure as a Service (IaaS) platform. SURFsara has implemented the Cloud service relying on the OpenNebula [63] software. Users interact with the HPC Cloud via the web interface that OpenNebula offers. SURFsara together with University of Sheffield, has produced the CompBioMed webinar, Introduction to Cloud Computing for the VPH [64], on the access and use of their IaaS cloud platform.

Microsoft Azure
Microsoft Research joined CompBioMed as associate partner in October 2017. They perform research within Microsoft but also collaborate in and supports research around the world well beyond the company itself. This partner has facilitated access to their Microsoft Azure [65] infrastructure and provided support to CompBioMed partners running on this platform.

8.2 Platform as a Service in CompBioMed

The Platform as a Service model relies on infrastructures made available by others where the work of dealing with the underlying resources and giving users a specialized environment (including for example additional security layers or efficient resource management tools), is abstracted by the provider. Two Associate Partners of CompBioMed provide this type of service:

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"

DNAnexus

DNAnexus Platform provides a data agnostic platform which allows one to store, manage, analyse and share data, with support to any type of data and analysis software. Most of the current applications are in the genomics space but do include a few in the Computational Chemistry space. DNAnexus services have been used by CompBioMed partner UCL for the deployment of their computational workflows using both AWS and Microsoft Azure infrastructures.

Alces Flight

Alces Flight is part of the Alces software group, and their aim is to build and deliver high performance compute clusters for customers utilising a wide range of open-source and custom-developed software. Within CompBioMed, Alces Flight has organised a workshop [66] to work on the deployment of bioinformatic software on AWS cloud services using their capabilities. This has enabled a quick package install guide to be produced for future users. Alces Flight has also taken part in the organisation of several workshops organised by CompBioMed within ISC [67], [68], and SC [69] conferences.

8.3 Software as a Service in CompBioMed

At the other end of the service model spectrum, Software as a Service (SaaS) offers the customer the capability to use services pre-deployed by the provider running on cloud infrastructure without the need to control directly any components of the stack, with the possible exception of limited user specific application configuration settings. Within the consortium we have continuously promoted the development of SaaS services through the use of cloud infrastructures [70] and containerisation technologies [71]. In this subsection, we present three examples of the work done by CompBioMed partners within this context:

Alya, containers in HPC

BSC worked on a comparative study of the performance and implementation issues of using their software Alya (see section 7.1.1) with different containers technologies: Singularity [72], Docker [73] and Shifter [74]). The study provided a performance comparison of the three container technologies, conducted on a cardiac mechanics fluid-structure simulation problem, which has shown that Singularity and Shifter can provide excellent service up to 112 MPI ranks. The results of this work have been presented at the CompBioMed Containerisation Meeting [71], and reported in deliverable D2.3 [1].

Molecular Dynamics in the Cloud

University College London (UCL) has worked on the deployment of the BAC (see section 7.1.4) workflow not only on very large HPC systems but also on Cloud computing infrastructures. The service has been ported to DNAnexus platform and runs on both MS Azure, and AWS resources. UCL has also worked to increase the portability of BAC interfacing some of the building block of the workflows with Docker container technology. This work is currently under development, and it will be continued and extended in CompBioMed 2.

Portable software for embarrassingly parallel analysis

University of Sheffield (USFD) and SURFsara (SARA) core partners have collaborated to optimise the execution of the openBF cardiovascular application (see section 7.2.1.2) for the generation of 1D network for a virtual population. This problem required the study of the virtual population through a Monte Carlo analysis, therefore requiring the submission of many independent simulations. In order to improve the performances of the full workflow, the code has been

“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“

ported to Singularity container technology. With this approach, the individual (single-core) simulations, were run with an embarrassingly parallel approach over a multi-core Virtual Machine (HPC cloud, SURFsara, NL), a tier-1 (Cartesius, SURFsara, NL) and a tier-3 HPC system (ShARC, Sheffield, UK), allowing reduction of the total compute time linearly with the number of cores available.

9 Conclusions

The overview presented here shows clearly the impact that the use of HPC resources is having on the production of high-level biomedical research and on the development of efficient computational services. We presented in detail the work, done within the CoE, to provide the community with favourable access routes to HPC systems, and specialised support for code development and integration with the HPC software stack present on the available resources.

The codes developed and used by CompBioMed partners have shown excellent performance on large and powerful supercomputers, reaching scale of parallelism which make them suitable candidates to run the next generation exascale compute systems. The use of HPC resources was also useful for the development and execution of biomedical workflows which have previously been unable to operate at these scales. Since the beginning of CompBioMed, users had accessed a heterogeneous pool of HPC systems, showing how the community not only requires large and powerful supercomputers, but also flexible and easy to access HPC in general, across the globe.

Much of this work would not have been possible without the help of CompBioMed for both the technical support for running codes efficiently on HPC systems, and in obtaining compute allocations on HPC systems required to perform the work

In the future, CompBioMed, will continue to promote more powerful, reliable, robust and flexible simulations by providing services to the community to help port and scale codes to appropriate HPC systems. We will intensify our service incubator activities (see D4.4 and D4.3 [1]), working together with code developers and infrastructure providers, to extend best-practices and guidelines for a correct infrastructure usage, as well as, to work in synergy for the co-design of future HPC machines.

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

10 Bibliography

[1] [Online]. Available: https://www.compbiomed.eu/media-social/publications/deliverables/.

[2] [Online]. Available: https://www.compbiomed.eu/high-performance-computer-allocations/.

[3] [Online]. Available: https://www.nwo.nl/en/.

[4] [Online]. Available: https://epsrc.ukri.org/research/facilities/hpc/.

[5] [Online]. Available: https://www.res.es/en/access-to-res.

[6] [Online]. Available: http://www.prace-ri.eu/call-announcements/.

[7] [Online]. Available: http://www.doeleadershipcomputing.org/.

[8] [Online]. Available: https://www.youtube.com/watch?v=1FvRSJ9W734.

[9] [Online]. Available: http://www.elem.bio/.

[10] [Online]. Available: https://www.openmp.org/.

[11] [Online]. Available: https://www.openacc.org/.

[12] [Online]. Available: https://docs.nvidia.com/cuda/cuda-runtime-api/index.html.

[13] [Online]. Available: https://www.gnu.org/.

[14] [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/metis/overview.

[15] [Online]. Available: https://www.hdfgroup.org/solutions/hdf5/.

[16] [Online]. Available: https://vtk.org/.

[17] [Online]. Available: https://www.paraview.org/.

[18] G. H. S. K. A. A. J. A.-S. R. A. D. M. H. C. F. C. H. O. A. T. E. D. B. J. M. C. M. V. Mariano Vázquez, "Alya: Multiphysics engineering simulation toward exascale.," vol. 14, no. 15-27, 2016.

[19] P. B. L. O. I. T. I. P. H. G. V. M. B. C. A.-S. J. Sacco F., "Left ventricular trabeculations decrease the wall shear stress and increase the intra-ventricular pressure drop in CFD simulations," 2018.

[20] P. B. L. O. I. T. I. P. H. G. V. M. B. C. A.-S. J. Sacco F., "Evaluating the roles of detailed endocardial structures on right ventricular haemodynamics by means of CFD simulations," 2018.

[21] J. A. M. Z. R. D. S. G. R. A. J. C. C. E. C. M. V. Alfonso Santiago, "Fully coupled fluid-electro-mechanical model of the human heart for supercomputers," 2018.

[22] [Online]. Available: http://www.prace-ri.eu/callproject/2017174226/.

[23] [Online]. Available: https://www.dicomstandard.org/.

[24] [Online]. Available: http://www.ukcomes.org/.

[25] [Online]. Available: https://pop-coe.eu/.

[26] [Online]. Available: https://cordis.europa.eu/project/rcn/197534/factsheet/en.

[27] [Online]. Available: https://pop-coe.eu/sites/default/files/pop_files/pop-ar-hemelb-b.pdf.

[28] [Online]. Available: https://www.ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5be0e99a1&appId=PPGMS.

[29] [Online]. Available: https://www.scalasca.org/.

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

[30] [Online]. Available: https://hemocell.eu/user_guide/QuickStart.html.

[31] L. Mountrakis, E. Lorenz, O. &. A. S. Malaspinas, B. Chopard and A. Hoekstra, "Parallel performance of an IB-LBM suspension simulation framework.," 2015.

[32] Z. G. H. A. Tarksalooyeh V.A., "Optimizing Parallel Performance of the Cell Based Blood Flow Simulation Software HemoCell.," 2019.

[33] [Online]. Available: https://github.com/eth-cscs/reframe.

[34] [Online]. Available: http://www.ensemblemd.com.

[35] [Online]. Available: http://www.ks.uiuc.edu/Research/namd/.

[36] [Online]. Available: www.gromacs.org.

[37] [Online]. Available: http://www.openmm.org.

[38] [Online]. Available: https://github.com/radical-cybertools/htbac.

[39] [Online]. Available: https://radical-cybertools.github.io/.

[40] [Online]. Available: https://www.lifetecgroup.com/cases-papers-products-services/case-angiosupport.

[41] A. R. M. A. T. C. F. C. J. G. D. S. M. a. T. S. Saltelli, Global Sensitivity Analysis: The Primer, John Wiley & Sons.

[42] [Online]. Available: https://slurm.schedmd.com/documentation.html.

[43] [Online]. Available: https://www.mathworks.com/products/matlab.html.

[44] [Online]. Available: https://github.com/INSIGNEO/openBF.

[45] [Online]. Available: https://julialang.org/.

[46] [Online]. Available: http://www.meshlab.net/.

[47] [Online]. Available: https://www.cs.ox.ac.uk/chaste/.

[48] [Online]. Available: https://xerces.apache.org/xerces-c/.

[49] [Online]. Available: https://www.cellml.org/.

[50] [Online]. Available: https://computing.llnl.gov/projects/sundials/cvode.

[51] [Online]. Available: https://github.com/mobernabeu/polnet.

[52] [Online]. Available: https://github.com/EPCCed/hemelb-hoff.

[53] N. C.-M. M. J.-R. H. v. V. a. G. T. Laura Pérez-Benito, "Predicting Activity Cliffs with Free-Energy Perturbation," 2019.

[54] [Online]. Available: https://www.schrodinger.com/fep.

[55] H. A. T.-N. A. Potterton A., "ynergistic Use of GPCR Modeling and SDM Experiments to Understand Ligand Binding," 2019.

[56] F. S. H. M. W. Y. S. M. J. B. A. H. P. V. C. a. A. T.-N. Andrew Potterton, "Ensemble-Based Steered Molecular Dynamics Predicts Relative Residence Time of A2A Receptor Binders," 2019.

[57] [Online]. Available: http://www.quiime.org.

[58] [Online]. Available: http://www.psicode.org/.

[59] [Online]. Available: https://www.htmd.org/.

[60] [Online]. Available: https://github.com/INSIGNEO/pFIRE.

[61] [Online]. Available: https://insigneo.github.io/pFIRE/docs.html.

[62] [Online]. Available: https://doc.hpccloud.surfsara.nl/.

[63] [Online]. Available: http://opennebula.org/.

*"This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **675451**"*

[64] [Online]. Available: https://www.compbiomed.eu/compbiomed-webinar-2/.

[65] [Online]. Available: https://azure.microsoft.com/en-us/.

[66] [Online]. Available: https://www.compbiomed.eu/alcesflight-and-compbiomed-work-together-to-install-galaxy-on-aws/.

[67] [Online]. Available: https://2018.isc-program.com/presentation/?id=bof129&sess=sess163.

[68] [Online]. Available: https://2019.isc-program.com/presentation/?id=bof145&sess=sess235.

[69] [Online]. Available: https://sc18.supercomputing.org/presentation/?id=bof136&sess=sess392.

[70] [Online]. Available: http://www.compbiomed.eu/events/cloud-high-performance-computing-in-biomedicine/.

[71] [Online]. Available: https://www.compbiomed.eu/events-2/compbiomed-containerisation-meeting/.

[72] [Online]. Available: https://sylabs.io/.

[73] [Online]. Available: https://www.docker.com/.

[74] [Online]. Available: https://docs.nersc.gov/programming/shifter/overview/.

*“This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No **675451**“*