Grant agreement no. 823712

CompBioMed2

Research and Innovation Action
H2020-INFRAEDI-2018-1
Topic: Centres of Excellence in computing applications

D3.3 Analytics Requirements Analysis

Work Package: 3

Due date of deliverable: Month 08

Actual submission date: 29 May 2020

Start date of project: 01 October 2019 Duration: 48 months

Lead beneficiary for this deliverable: *UCL*
Contributors: *UOXF, SARA, UEDIN, LRZ, ANL, and UNIBO*

Disclaimer

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 **823712**"*

Table of Contents

1 Version Log 3
2 Contributors 3
3 Definition and Acronyms 4
4 Public Summary 5
5 Introduction 5
6 The role of machine learning and data analytics in CompBioMed 5
6.1 Molecular Medicine 9
6.2 Cardiovascular medicine 10
6.3 Neuromuscular-skeletal medicine 12
6.4 The confluence of HPC and HPDA on emerging exascale architectures 13
7 Conclusions 14
8 Bibliography 14

List of Tables and Figures

Figure 1: CompBioMed partners responses showing how time-consuming data analysis is. 6
Table 1: Summary on yes/no questions in the survey. The remaining fraction voted "no". 6
Figure 2: Doughnut charts on the distribution of data types in the three CompBioMed Biomedicine fields: molecular medicine, cardiovascular medicine and neuromuscular-skeletal medicine. 7
Figure 3: Histogram of the size of data analysed for each of the three CompBioMed's biomedical fields. The data was normalised for each of the three fields. 7
Figure 4: Histogram of the hardware used to carry out data analysis across the three fields in CompBioMed. Data was normalised for each of the three fields. 8
Table 2: List of software and programming languages used in CompBioMed for data analysis. The results are divided into the three exemplar biomedical fields. The number in parenthesis denotes the percentage of participants selecting the software/language within each of the three fields. 8
Table 3: List showing how analytics techniques could still be improved in the three biomedical fields. The number in parenthesis denotes the percentage of participants who mentioned these improvements. 9

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

1 Version Log

Version	Date	Released by	Nature of Change
V0.1	07/05/20	AP & MB	First Draft
V0.2	21/05/20	AP & MB	Second draft after comments from reviewers
V1.0	29/05/20	Emily Lumley	Final Draft, submitted to the EC

2 Contributors

Name	Institution	Role
Mateusz Bieniek (MB)	UCL	Co-Author
Andrew Potterton (AP)	UCL	Co-Author
Jon McCullough	UCL	Contributor
Vicente Grau	UOXF	Contributor
Antonino A. La Mattina	UNIBO	Contributor
Philip W Fowler	UOXF	Contributor
Austin R Clyde	ANL	Contributor
Peter Coveney	UCL	Reviewer
Emily Lumley	UCL	Reviewer
David Wifling	LRZ	Reviewer
Narges Zarrabi	SARA	Reviewer
Gavin Pringle	UEDIN	Reviewer
Marco Verdicchio	SARA	Reviewer

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

3 Definition and Acronyms

Acronyms	Definitions
ARC	Advanced Research Computing
CMR	Cardiac Magnetic Resonance
CoE	Centre of Excellence
CPU	Core Processor Unit
CUDA	Compute Unified Device Architecture
ECG	Electrocardiogram
ESMACS	Enhanced Sampling of Molecular Dynamics with Approximation of Continuum Solvent
FEM	Finite Element Method
GPU	Graphics Processor Unit
HPC	High performance computing
HPDA	High performance data analytics
HTMD	High-Throughput Molecular Dynamics
LRZ	Leibniz Supercomputing Centre
LV/RV	Left/Right Ventricular
MD	Molecular Dynamics
ML	Machine Learning
MMGBSA	Molecular Mechanisms Generalized Born Surface Area
MPI	Message Passing Interface
OCLF	Oak Ridge Leadership Computing Facility
PBSA	Poisson-Boltzman Surface Area
PC	Personal Computer
PID	Persistent Identifier
RAID	Redundant Array of Independent Disks
ROI	Region of Interest
SSH	Secure Shell
TACC	Texas Advanced Computing Center
TIES	Thermodynamic Integration with Enhanced Sampling

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

4 Public Summary

In this deliverable, we summarise findings concerning the current state of data analytics in the CompBioMed Centre of Excellence. We gained this insight through case studies and a survey answered by partner institutions. There were found to be differences in the requirements and current state of data analytics in our three biomedical exemplar domains: molecular medicine, cardiovascular medicine, and neuromuscular-skeletal medicine. Finally, an exemplar project in the molecular medicine field, that uses both machine learning and ultra-high-end supercomputers to carry out data analytics in combination with high performance computing, is presented.

5 Introduction

This deliverable "Analytics Requirements Analysis (D3.3)" focuses on understanding how CompBioMed partners carry out data analytics of biomedical data. This includes the computational resources that are used to run data analytics, with the aim of moving towards high performance data analytics (HPDA). HPDA is the use of high-performance computers (HPC) to carry out data analysis enabling high capacity and fast results. To enable the shift to HPDA, one needs to understand the programs used to analyse data and the type and size of the data analysed. In addition to HPDA, there is a focus on advanced analytics techniques, including machine learning (ML), in the report.

As CompBioMed is a user-driven CoE (Centre of Excellence), this deliverable is centred around responses from CompBioMed project partners. These responses will be used to determine the next steps. This preliminary exercise will be used to gauge our own partners needs and will subsequently be rolled out to our Associate Partners and other users.

6 The role of machine learning and data analytics in CompBioMed

Two methods were used to gain insight into CompBioMed's current data analytics ability and how it could be improved; first a survey, and second case studies.

For the survey, CompBioMed partners were asked to fill out a web form containing a series of questions concerning their data analysis and how this process could be improved. This survey enabled the assessment of the current data analysis approaches and helped to determine the steps to increase the use of HPDA. In total, 32 responses from all relevant partners were collected.

The first set of questions in the survey which are listed in Table 1 together with their results were used to acquire general information about analysis. The results show that analytics is of core importance to a large majority (71%) of the partners. At the same time, 29% of the participants indicated room for improvement in the analysis know-how of the team. The question about reproducibility also reveals room for improvement in a key area of science, with 21% of participants unsure as to whether their analysis can be reproduced with the software/tools that they utilised.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

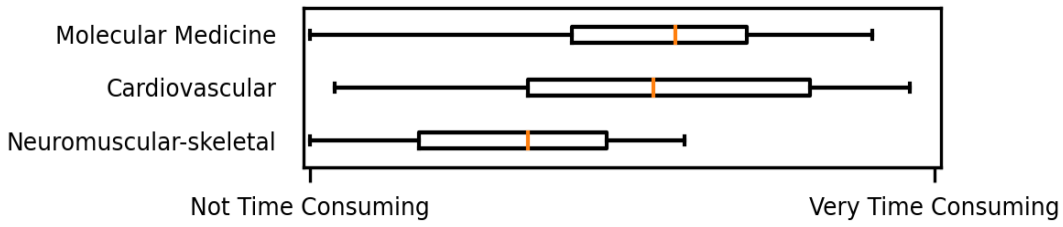

Figure 1: CompBioMed partners responses showing how time-consuming data analysis is.

The survey shows that, on average, analysis takes a moderate amount of the researcher's time (Figure 1). However, the variance in the responses is relatively large, particularly in the cardiovascular field. Whereas the neuromuscular-skeletal field appears to spend less time on analysis, it should be noted that only 3 responses were collected in that category. When asked the question about using machine learning, slightly over 50% stated that they use some form of machine learning. Furthermore, the majority of participants stated that they were not working with missing/noisy data.

Table 1: Summary on yes/no questions in the survey. The remaining fraction voted "no".

Question	Yes (%)	Unsure (%)
Are the analytics techniques of core importance to your project?	71	-
Do you have the know-how in the team to carry out analysis seamlessly?	71	-
Can another group reproduce your analysis with the software/analysis tools you use?	75	21
Do you use any machine learning methods in your analysis?	54	-
Do you have problems with missing/noisy data?	34	7
Is metadata available for your data?	45	17
Do you have/require PIDs (persistent identifiers) for the data?	17	34

The questionnaire contained two questions regarding metadata, specifically data annotation. Almost half of the participants stated that metadata is available for their data, whereas 17% were not sure. This suggests room for improvement in terms of how the data is annotated, which is key for sharing, finding, and reproducible data. However, despite the majority knowing that metadata are available for their data, only 17% stated that they have or require persistent identifiers, perhaps because this data does not form a part of the final publication.

The next questions were compiled into three fields: molecular medicine, cardiovascular medicine, and neuromuscular-skeletal medicine. This breakdown aims to distinguish and discuss the differences found across fields present in the CompBioMed CoE.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

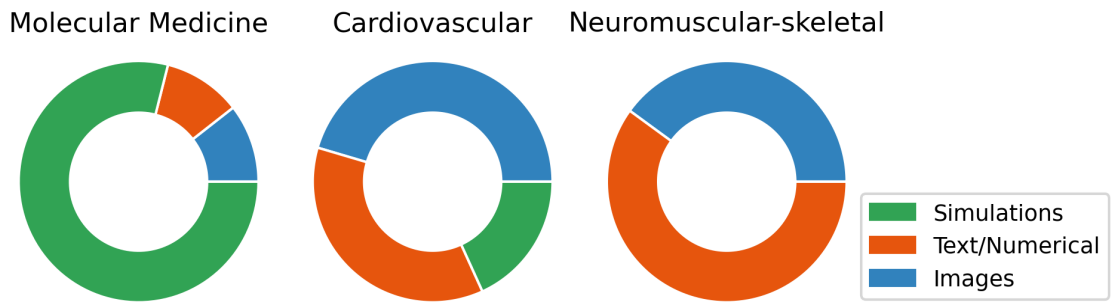

Figure 2: Doughnut charts on the distribution of data types in the three CompBioMed Biomedicine fields: molecular medicine, cardiovascular medicine and neuromuscular-skeletal medicine.

Three data types were singled out: simulations (data derived from simulations), text/numerical, and images (Figure 2). Simulations are primarily analysed in the molecular field and specifically refer to molecular dynamics simulations. In addition, simulations have a small prevalence in the cardiovascular field, where Text/Numerical and Images dominate. The latter two also dominate in the neuromuscular-skeletal field where simulations are even absent. Furthermore, the participants in the cardiovascular group stated that they work with ECG and 3D volumetric data.

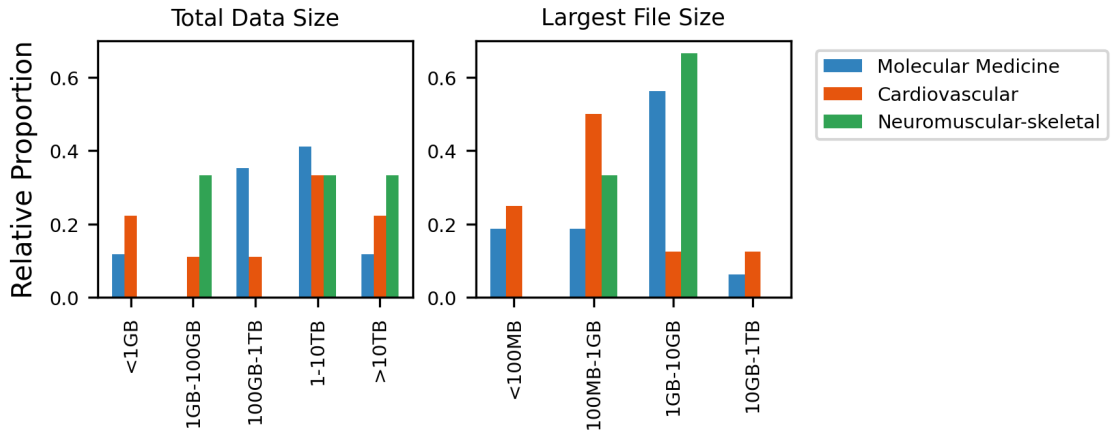

Figure 3: Histogram of the size of data analysed for each of the three CompBioMed's biomedical fields. The data was normalised for each of the three fields.

Most partners within the three fields work with data of large size (Figure 3). The majority in all three groups work with data larger than 100GB, with the 1-10TB size being the most frequent. Nonetheless, a significant number of participants work with data larger than 10TB. The size of the largest file varies significantly (Figure 3), with the largest, 10GB-1TB being rarely used. Most of the files are found in the range 100MB-10GB.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

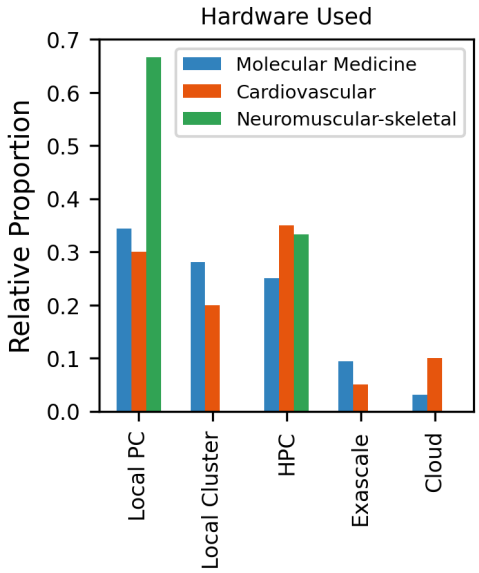

Figure 4: Histogram of the hardware used to carry out data analysis across the three fields in CompBioMed. Data was normalised for each of the three fields.

Despite the large data sets, more than 70% of respondents carry out their analysis on a local PC or a local cluster (Figure 4). Besides local resources, the next most commonly used resource for analysis is HPC. Machines in preparation for the exascale are being used by a small number of respondents to carry out machine learning, which is quite surprising that this sort of data analysis is carried out on this computing scale.

Table 2: List of software and programming languages used in CompBioMed for data analysis. The results are divided into the three exemplar biomedical fields. The number in parenthesis denotes the percentage of participants selecting the software/language within each of the three fields.

Molecular Medicine	**Cardiovascular**	**Neuromuscular-skeletal**
Python: scikit-learn, pandas, numpy, MDAnalysis, Seaborn, Jupyter notebooks (64%) MOE (18%) R (9%) KNIME (9%)	Python: Tensorflow, PyTorch (70%) C++ (10%) Matlab (10%) R (10%)	Python (50%) Matlab (50%)

The most commonly used tools for analysis are written in Python (Table 2). In the molecular medicine field, the majority of the mentioned python packages are intended for numerical analysis. In contrast, in the cardiovascular field, all selected python packages are intended for machine learning.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

Table 3: List showing how analytics techniques could still be improved in the three biomedical fields. The number in parenthesis denotes the percentage of participants who mentioned these improvements.

Molecular Medicine	Cardiovascular	Neuromuscular-skeletal
• Automation (72%) • Speed (14%) • Reproducibility (14%)	• Automation (40%) • Speed (30%) • Simplification of data (20%) • Memory issues (10%)	• Automation (100%)

Responses to questions regarding the scope for improvement of analytics techniques in these three biomedical fields show that automation is the most frequently requested improvement (Table 3). The next in turn is software speed. The two together point towards performance and efficiency demanding more extensive investigation and/or work on code optimisation.

Four case studies were collected from CompBioMed members from each of the three biomedicine fields. These researchers were asked three questions to guide their case study:

- *What data analytics do you currently carry out?*
- *What resources do you run your analytics software/tools and why?*
- *What do you want to improve about your analytics or how they are carried out?*

6.1 Molecular Medicine

Phillip W Fowler, Oxford University

What data analytics do you currently carry out?

We use computer modelling to predict *de novo* whether a mutation in the coding region of a bacterial gene confers resistance to an antibiotic. This is developing into a combination of machine learning (to quickly rule out mutations that have no effect) followed by intensive, high-throughput molecular dynamics simulations (GROMACS) to predict the effect of the mutation on the binding free energy of the antibiotic using alchemical free energy methods [1]. This case study focuses on the latter step, in particular our published work.[2]

In previous work, accurately and reproducibly assessing whether a mutation conferred resistance required 32 independent calculations of ΔΔG. Each of those required 8 alchemical free energies (ΔG) and each of those required 8-16 single core MD simulations.

Calculating each value of ΔG requires an input file from each of the 8-16 simulations to be processed using Python. There are a number of Python libraries[3] that support this task, but simply storing and parsing this large number of simulation files is a difficult and time-consuming task. I often use another Python library[4] that allows file discovery and filtering on tags within Python.

What resources do you run your analytics software/tools and why?

At present I am just about able to store and analyse all the MD files on my 12-core workstation with attached ~12 TB RAID array, but I am at the "hacky limit" and e.g. am using GNU parallel to accelerate compression and Python analysis jobs. I will have to either run analysis as a

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

separate job on the high-performance computing cluster, or more efficiently, make the analysis part of the "production simulation" HPC submission script.

What do you want to improve about your analytics or how they are carried out?
The above simply gives you the raw ΔG values. What is missing is a level of dynamism. For example, what one would like to do is set a maximum magnitude for the error of the binding free energy and have an algorithm dynamically decide which alchemical legs to repeat to reduce the overall estimated error. This would require (i) the computational job to be the *calculation* of a single alchemical ΔG (rather than just running the MD sims necessary) and (ii) the results of these analyses to be constantly fed into a process which can launch additional simulations until some pre-set criteria are satisfied. This is necessary if techniques like ours are to be adopted in the clinic, since they need to be as "hands off" as possible.

6.2 Cardiovascular medicine

Jon McCullough, UCL

What data analytics do you currently carry out?
HemeLB is a Lattice Boltzmann method based fluid flow solver that specialises in the study of flow through sparse geometries. In particular it has been optimised for the study of blood flow through vascular systems and has been used to examine flow patterns on a range of length scales. HemeLB is currently a CPU-based C++ code parallelised using MPI.

Accurately resolving flow through a vascular geometry routinely requires domains consisting of several million fluid sites. Extremely large or very highly resolved geometries may require over a billion data locations. The magnitude of this dataset means that analysis is often restricted to the study of flow velocity profiles and streamlines at specific regions of interest.

A typical workflow of HemeLB data analysis is as follows: 1) identify the location of geometric feature to be analysed and request appropriate data to be saved for that location (e.g. flow velocity at entry, mid-way and exit planes of a vessel); 2) run simulation; 3) convert compressed data file to human readable format and make this usable for analysis; 4) analyse output either with custom script or visualisation package (e.g. Paraview). Steps 3 (exclusively) and 4 are often completed with custom written analysis scripts to obtain the desired information. Paraview is a commonly used package in the field of computational fluid dynamics however HemeLB's native data output format is not optimally compatible with it (in part due to the data structures used to captures the irregular geometries regularly studied). This means that custom scripts are often written to generate a desired output.

What resources do you run your analytics software/tools and why?
Data conversion to human readable format is typically carried out (for me in the last 6-12 months) on HPC clusters Archer and SuperMUC-NG due to convenience of data location and presence of sufficient resources to perform the conversion. As noted previously, size of data can become very large, particularly if multiple time outputs are desired. Generation of analysis plots is typically performed on local machines in terms of visualisation capability.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

What do you want to improve about your analytics or how they are carried out?

The data analysis of HemeLB output could be improved in a number of ways. The first would be minimising the number of steps between the data output from the code (typically a compressed *.dat file) and being visualised. A second would be enabling a greater quantity of the output data to be effectively visualised (planes only capture a small component of data and full geometry output can require several GB of memory per output step to store). Developing effective analysis and visualisation techniques will be key to making best use of the large-scale simulation capabilities of HemeLB.

There is a mix of programming languages due to reasons of legacy, convenience and preference. The tool that converts compressed data files to human readable ones is written in C++ (as is the main HemeLB package). Most analysis scripts I write are coded in python, but there is a bash script in use as well. Streamlining this mix would be another potential area to improve our (my) current analytics approach.

Vicente Grau, Oxford University

What data analytics do you currently carry out?

Up to now, our work has mostly focused on the development of algorithms to transform Cardiac Magnetic Resonance (CMR) studies into meshes for personalised cardiac modelling. The main techniques this involves are:

- Segmentation of two-dimensional CMR images to extract Left and Right Ventricular (LV and RV) contours.
- Generation of three-dimensional meshes from two-dimensional contours, correcting for potential misalignment between slices.
- Calculation of relevant parameters from generated slices, both for quality control and to build population models. This includes the calculation of volume and shape metrics (e.g. end-systolic and end-diastolic volumes).
- Building deep learning models of the shape of the heart within the torso.

What resources do you run your analytics software/tools and why?

Our tools are mostly Python code developed using machine learning toolkits such as Keras/TensorFlow and Pytorch. We run these primarily on:

- Local computers with a standard GPU card, for prototyping and testing algorithms on small datasets.
- A local cluster with CPU nodes connected to four GPUs each, for training on larger datasets
- The University of Oxford Advanced Research Computing facilities (ARC), using GPUs.
- The main reason is availability and ease of data transfer.

What do you want to improve about your analytics or how they are carried out?

We have not yet worked on large amounts of data, but the plan is to scale up to the UK Biobank data volumes (~50,000 subjects.) With our current computational setup this would be impossible. Working with a large dataset on a remote supercomputer could be one of our main challenges.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

6.3 Neuromuscular-skeletal medicine

Antonino A. La Mattina, University of Bologna

What data analytics do you currently carry out?

In the musculoskeletal FEM simulations carried out, we are interested in calculating femur mechanical strains in dependence on different side-fall conditions, to evaluate its fracture risk. In particular, we are currently testing 28 different loading directions on each femur (currently 98 cases).

For each simulation carried out, we obtain around 2 GB of ANSYS result files, and extract from the entire model a region of interest (ROI) of around 6000 points by writing node coordinates and principal strains in a csv file (around 500 KB). We are thinking about using a different file format (possibly binary, such as HDF5) to efficiently store larger ROIs that include larger femur regions (up to a few million nodes).

To post-process the data we developed a simple Python script that leverages standard modules (numpy, scipy, pandas, sklearn) to calculate the distance matrix between the ROI points and average the strains over a sample volume (3 mm radius) around each node. Then we select the node with the maximum ratio between simulated mechanical strain and the bone critical strain and estimate the femur failure load in the simulated conditions. At the end of the 28 simulations of each femur, a bidimensional surface representing the femur critical load is obtained.

What resources do you run your analytics software/tools and why?

I am running FEM meshing and simulations and data analysis on a HPC cluster (currently ShARC, at Sheffield University; we have also gained access to Cartesius at SURFsara), but currently material mapping is performed locally on my office workstation because the software only runs on Windows (we are working on some kind of porting). The simulations are embarrassingly parallel and each single simulation does not require a huge amount of computational resources, so initial test runs are serially performed on my local PC before batch processing on HPC, where several simulations can be run simultaneously.

What do you want to improve about your analytics or how they are carried out?

The next research steps include running simulations over a 1000-femur cohort followed by Monte Carlo integration (around 10^6 sampling points) to probabilistically estimate the generalised fracture risk for every femur, based on the different loading condition probability. Furthermore, we plan to track femur mechanical property evolution over time, so that file storage (around 100 MB for each femur mesh input file at a certain time) will become a relevant issue in the future. The scaling of the Python scripts also has to be evaluated, and possibly the analysis script will be reimplemented in more performant languages, such as C or FORTRAN.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

6.4 The confluence of HPC and HPDA on emerging exascale architectures

Here we present a case study from Austin Clyde, who is part of Argonne National laboratory, a CompBioMed's international partner. He is using machine learning, molecular dynamics simulations on HPC, and HPDA, to try to identity compounds that potentially binding to SARS-CoV-2's viral proteins and human proteins that the virus uses as part of its life cycle.

Austin Clyde, Argonne National Laboratory

What data analytics do you currently carry out?

We utilize GPU-accelerated molecular dynamics simulations (HTMD/OpenMM) coupled with deep learning models to produce fast and accurate binding free energy approximations over large lead discovery databases. Our workflow takes a database of molecules and runs the process from 3D-conformer generation, to docking ligands to the protein, to computing fast approximations of the binding free energy (MMGBSA/PBSA), to rapid and state of the art relative free energy predictions using methods (ESMACS and TIES) from Peter Coveney's group at UCL. Between all these stages, we utilize deep learning to predict whether or not continuing to the next state will yield an interesting result or not. The data set is filtered down from a size of billions by an order of magnitude because each stage becomes computationally more expensive. At the end, we have a highly accurate set of binding free energy estimates on the most exciting leads to pursue.

What resources do you run your analytics software/tools and why?

Given the nature of deep learning and fast molecular dynamics simulations, we require NVIDIA based GPU hardware to run CUDA supported software packages effectively. We currently utilize in-house GPU-clusters at Argonne National Laboratory for development and debugging ranging in size from single node machines with 16-32 GPUs to smaller clusters with five nodes. We heavily use local resources as we are so able to use Secure Shell (SSH) for direct login to the machines and thus do not have to 'hassle' with submission scripts, queue wait times, and permissions for installing software. Given large amounts of data generated, working with machines on the same network also means little consideration for file transfer times.

However, given we are often deploying the workflows for lead discovery, data set sizes range up to an order of millions and billions. In these cases, we have to use the largest computer resources available. For large production runs, we employ the scientific workflow expertise/software from Shantenu Jha's group. We run on resources at Argonne Leadership Computing Facility (ALCF) including Theta, a CPU-only supercomputer, Oak Ridge Leadership Computing Facility (OLCF) including Summit, a GPU supercomputer, Texas Advanced Computing Center (TACC) Frontera (mixed CPU/GPU supercomputer) and Longhorn (GPU supercomputer), and Leibniz-Rechenzentrum Supercomputing Centre (LRZ) including Super-MUC-NG. None of these systems are in configurations that are fully desirable. The "dream" system for this type of workflow would be both x86 capable compute nodes and including NVIDIA GPU accelerators on each node. Theta, Super-MUC-NG and to a limited extent also

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*

TACC resources all satisfy the x86 requirements but lack a sufficient number of GPUs needed for our workloads. Summit is exceptionally powerful with regard to GPU resources but the POWER9 architecture implies that most of our software needs to be recompiled to run.

What do you want to improve about your analytics or how they are carried out?
As described in the previous paragraph, no system is currently ideal. The biggest impediment to our work is tooling required to run on a supercomputing system. This often implies running different steps of the workflow on different systems. Either locating an ideal system or designing an interface between supercomputing facilities would improve our throughput and work dramatically.

7 Conclusions

In this deliverable, we have analysed the current data analysis capabilities of CompBioMed's partners. We achieved this through two means: case studies, and a survey sent to all partners. Data analytics was found to be of high importance to the majority of members, with over half of our participants using machine learning in their workflows. Encouragingly, the second most utilised computing resource to carry out data analytics was HPC. To increase usage of HPC within data analytics further, supercomputers with a strong hybrid CPU/GPU will have to become more prevalent, the Swiss national supercomputer PizDaint is one of the only hybrid supercomputers in Europe. Having these hybrid machines reduces the need to transfer files between different supercomputing centres, many users from the analytics survey and case studies identified this as a reason why local machines were currently being used instead of HPC.

The high level of use of machine learning and HPC how members of CompBioMed are already performing advanced data analytics techniques and engaging HPDA, therefore no further internal training is recommended in this area. The consortium is therefore in a position where it can offer training in the form of webinars or workshops, so that those outside the consortium can benefit. One of these workshops could be on the topic of "Automation in Python" as this was the most frequently suggested area of development (see Table 3) and most used programming language (see Table 2).

8 Bibliography

1. Fowler PW, et al. (2018) Robust Prediction of Resistance to Trimethoprim in Staphylococcus aureus. *Cell Chem Biol* 25(3):339-349.e4.
2. Brankin AE, Fowler PW (2019) Predicting Resistance Is (Not) Futile. *ACS Cent Sci* 5(8):1312–1314.
3. https://alchemlyb.readthedocs.io/en/latest/index.html
4. http://datreant.org/

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No **823712**"*