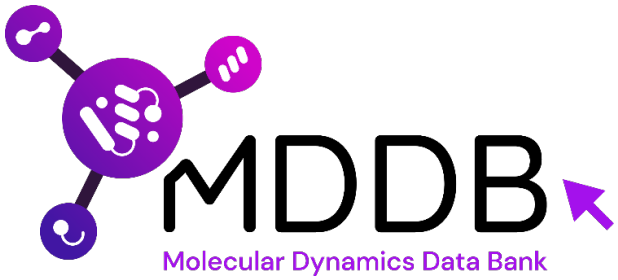

Project Acronym: MDDB

Project title: Molecular Dynamics Data Bank. The European Repository for Biosimulation Data

Call: HORIZON-INFRA-2022-DEV-01

Topic: HORIZON-INFRA-2022-DEV-01-01- Research Infrastructure Concept Development

Project Number: 101094651

Project Coordinator: Institute for Research in Biomedicine (IRB Barcelona)

Project start date: 01/03/2023

Project end date: 28/02/2026

Deliverable 2.1: Report on the evaluation of requirements and technical alternatives

Work Package: WP2 – Technical infrastructure

Lead beneficiary: BSC CNS

Dissemination level: PUBLIC

Due date: 10/03/2024

Actual submission date: 11/03/2024

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Document history

Version	Contributor(s)	Partner	Date	Comments
0.1	Josep Ll. Gelpi	BCN CNS	26/02/2024	First draft
0.2	D. Beltran, A. Hospital	IRB	29/02/2024	Revision
1.0			11/03/2024	Final version

Table of contents

Executive summary

During the initial period of the MDDB project, an analysis of the past and present simulation repositories has been performed. Also, conversations among partners involved in such repositories have led to the definition of the initial technical framework that will support MDDB operations. Activities on WP2 will be largely connected to WP1 with regards to data formats, and to WP3 to keep a continuous observation of the requirements and the evolution of technical solutions. This document outlines the overall technical strategy, fixes some required terminology and examines the available technical alternatives, based on solutions being applied to this field and others in the life sciences domain. The overall plan will be assayed by the implementation during this year of several new repositories, gathering the necessary feedback to eventually tune the original design.

1 INTRODUCTION

Molecular dynamics (MD) simulations have experienced a significant breakthrough through the use of large-scale supercomputers reaching the Exascale regime, the use of hardware accelerators and the general improvement of simulation software. MD is no longer an isolated technology available only to highly specialized researchers but is now attaining simulation lengths that are compatible with biological scale processes, and hence must be challenged with them. Experimental 3D structures, generally available at the Protein Data Bank[1] lack a dynamics perspective while macromolecular dynamics has proven to be a key concept in the understanding of the structure-function relationships in macromolecules[2]. Adding to this, the general availability of 3D structures for proteins due to AlphaFold[3] is generating an increasing interest in the use of structural data by the more general biological researcher which only further stresses the need for the inclusion of dynamic aspects in the equation. Molecular dynamics simulations, as in present state, is the best option to derive such dynamics properties for macromolecules. MDDB addresses a missing component in this scenario, the general availability of simulation data for macromolecular systems in the way as PDB and AlphaFold provide 3D structural data. This deliverable reports on the conceptual design and the technical alternatives being examined to address the building of the technical infrastructure for MDDB. The handling of simulation data implies a major challenge in comparison with existing structure databases. In MD, a single simulation experiment done at the state-of-the-art level, generates as much data as the whole PDB[4]. The total amount of expected data in MDDB would exceed the capacity of any single institution. For this reason, MDDB is conceived as a federated project. Data will be maintained in a distributed manner while the necessary technical solutions to allow MDDB to work as a unified repository will be adopted. The following report is structured to cover the following general requirements:

- **Data storage and mobilization**. Addressing two main objectives: 1) Long term data storage, with priority to minimize storage requirements, and 2) Data storage systems providing fast data mobilization, allowing for fast data browse and analysis.
- **Data compression and transfer**: Although envisioned as a federated system, MDDB is likely to require a heavy data transmission, moving data to long term storage, and specially including data submission, among others. Minimizing the time required for data transmission among centers, or the time required for data submitters to add new contents to MDDB is a crucial aspect to cover.
- **Analysis and visualization**: Trajectory data has little value *per se* unless such data is analyzed and provides answers to scientific questions. A systematic analysis addressing Quality Control (QC) is required to both help data providers to evaluate their simulations' quality, and for MDDB to keep the level of quality of the simulations being included in the repository. More

[1] https://www.ebi.ac.uk/pdbe/

[2] Hospital A, Goni JR, Orozco M, Gelpi JL (2015) *Molecular Dynamics Simulations: Advances and Applications*. Adv Appl Bioinform Chem 8: 37–47

[3] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A, Meyer C, Kohl SAA, Ballard AJ, Cowie A, Romera-Paredes B, Nikolov S, Jain R, Adler J, …Hassabis D (2021) *Highly Accurate Protein Structure Prediction with AlphaFold*. Nature 596(7873): 583–89

[4] Hospital A, Battistini F, Soliva R, Gelpí JL, Orozco M (2020) *Surviving the Deluge of Biosimulation Data*. WIREs Computational Molecular Science 10(3): e1449

specialized analysis and also user-driven analysis including easy visualization of simulation trajectories are needed for users to fully grasp the information that can be obtained from simulation experiments. Additionally, analysis results are the main outcome of the field regarding to export significant information to the biological data ecosystem in a way that can be shown together with current biological annotations.

- **Interfaces to HTC/HPC**: Simulation data is mainly produced at large computational facilities. Providing user-friendly interfaces for simulation producers will ease the usage of such facilities, normally restricted to command-line access, and traditionally managed from simple user-produced ad-hoc scripts.

The following sections will elaborate on the details of addressing these general requirements within MDDB.

2 CONCEPTUAL DESIGN AND REQUIREMENTS

MDDB is globally conceived as a federated repository (Fig. 1) including a central management node and a series of data nodes with different levels of involvement.

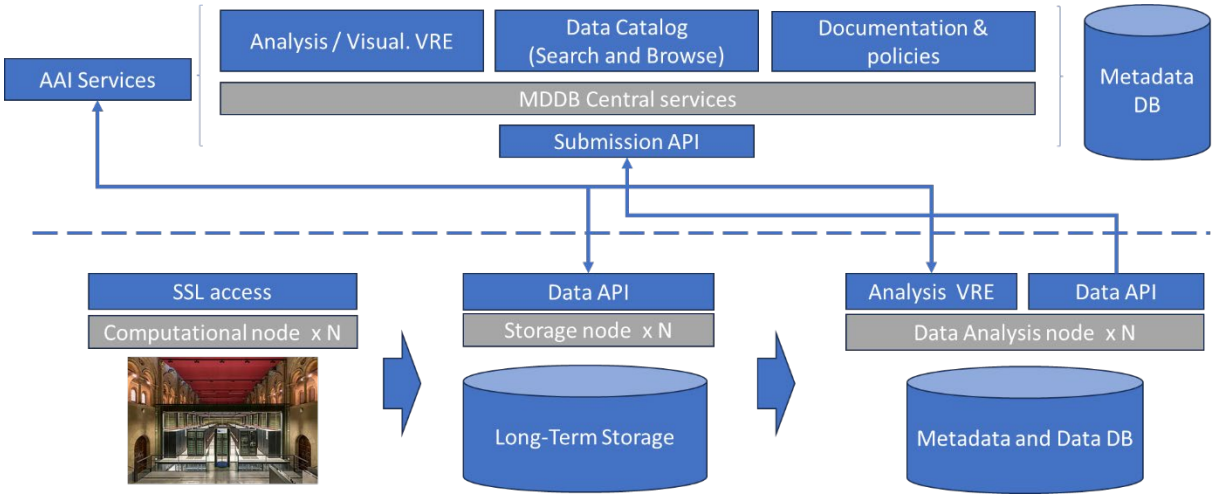

Fig. 1: Overall schema of MDDB infrastructure. Data will be generated at computational nodes, transferred to long-term storage and then analyzed and exposed in data analysis nodes. The federation level will be provided by MDDB central services.

2.1 MDDB central management node

Central management node should host core services as follows:

- the main MDDB web portal, including the reference documentation about the MDDB policies and procedures, description of the federation and links to data and computational nodes,
- a federated authentication/authorization system compatible with accepted standards,
- a global (meta)data catalog, powered by a common metadata database, with the necessary APIs for data identification (browse and search facilities),
- an API/portal for metadata submission,
- a module for management of access rights when necessary,
- a software repository including the reference implementation of MDDB toolkit.

MDDB central management should not require large computational resources.

2.2 MDDB nodes typology

MDDB associated nodes will span a series of modalities shown here as a list of prototypical scenarios. Actual sites may belong to one of these categories or share features of more than one, according to the level of engagement with MDDB. This report regards only the technical characteristics with no insight in the legal or policy related issues regarding the involvement with MDDB that would be developed elsewhere.

2.2.1 Computational nodes (likely HPC centers)

Computational nodes, most probably high capacity HPC centers, are the most relevant facilities to produce simulation data. Their main characteristic is the availability of significant computational resources, including traditional or accelerated hardware, most usually provided as large-scale clusters. Examples include the most powerful EuroHPC supercomputers[5] like LUMI, Marenostrum 5 or Leonardo, but also a significant number of clusters provided by smaller centers or even belonging to single research groups. A common characteristic of this type of node, especially large-scale ones, is that execution time is allocated for a limited time period and no long-term storage is provided as part of these allocations. In addition, the access to such facilities is highly restricted allowing in the simplest case only terminal access via SSH, but in other cases even VPNs should be established. Data management from users' perspective is largely based on POSIX file systems, although the underlying hardware may differ.

MDDB will expect from computational nodes to deploy:

- simulation software and helper (command-line) applications, adapted to the specific characteristics of available hardware,
- deployment of MDDB software stack, regarding setup, execution, and analysis of simulation experiments to assure that simulation and analysis results fulfill the agreed levels of quality and provenance recording.

2.2.2 Long-term data storage nodes

Long-term data storage nodes will provide the necessary resources for long-term storage. It is expected that they will be associated with computational centers, which will ease data deposition by minimizing the needs for data transmission. The main characteristics of this kind of node would be large-scale storage resources but low data mobilization capabilities. A usual layout would be based on hierarchical (HSM) systems with most of the storage provided by larger latency systems (e.g. tapes) combined with a disk-based buffer system. This layout is not designed for data mobilization. Instead, large, highly compressed files are expected, and access to them would imply a significant latency. Taking as example, the Data Service provided by the Spanish Supercomputing Network (RES)[6] data storage projects up to 5 years can be granted, with a capacity from 200 TB to up to 1 PB based on a HSM system. Data is hosted by specific facilities co-located with HPC centers belonging to the RES. Projects are provided with a single virtual machine (VM) for data management.

MDDB will rely on this kind of facility for long-term storage of raw trajectory data and eventually associated analysis data. MDDB will identify the stored data with the necessary unique identifiers and metadata, to make it accessible in the main catalog.

[5] https://eurohpc-ju.europa.eu/supercomputers/our-supercomputers_en
[6] https://www.res.es/en

MDDB will require from long-term storage facilities:

- long-term storage for large simulation datasets,
- a data API to both add and access data in the repository and synchronize metadata with the central data catalog,
- appropriate bandwidth for data transfer, and possibly specialized data transfer software.

2.2.3 Data storage/analysis nodes.

Data storage and analysis (project) nodes would be the most relevant components of MDDB federation. Opposite to long-term storage sites, data stored here would not be complete raw trajectories, but for instance, trajectories without solvent, representative curated datasets, relevant meta-trajectories (processed trajectories), and will be stored together with analysis results (QC and also specific analysis). Data stored here should be linked to the appropriate entries on long-term storage. Nodes will normally implement an independent data portal with interactive access, and eventually an analysis virtual environment, where users may execute their own analysis pipelines. A reasonable setup will be an infrastructure centered in a database management system allowing for fast data mobilization, and implementing a data access API, complemented with an execution cluster for data analysis. This type of node will largely cover already existing project data portals. MDDB portal will derive users to the project's portal from the main data/metadata catalog.

MDDB will expect from such project nodes, independently on the underpinning infrastructure and the structure of the portal:

- to deploy a MDDB-compliant data access API,
- provide a minimum set of analysis results (QC),
- synchronize metadata with the central data catalog,
- provide documentation about data access policies and manage access according to them.

2.2.4 Levels for federation

The above node types are described as abstract components of the federation, although they will likely be grouped in more complex infrastructures. Taking as example the bioexcel-cv19 portal[7], raw trajectory data is provided by data contributors to a long-term storage facility supported by RES and hosted at the Barcelona Supercomputing Center where it is maintained. Data is then temporarily transferred to BSC HPC facilities where trajectories are processed, solvent is removed, and a set of analysis performed. Processed trajectories and analysis are then transferred to a noSQL database systems based on MongoDB[8], accessible through a data API[9] that in turn is used by the bioexcel-cv19 data portal. This infrastructure ensemble, albeit complex, would correspond to a single project node.

MDDB envisions several levels of data federation. Large-scale nodes, possibly HPC centers, will provide with a single infrastructure the three main kinds of nodes, computational, long-term and analysis for a series of projects. These projects will not need to have other common features that have been produced in the center, with the clear advantage of avoiding data transmission, and possibly take advantage of the already existing infrastructures. Thematic nodes will provide the infrastructure to host together projects sharing a common scientific goal, e.g. drug targets for a specific kind of disease,

[7] https://bioexcel-cv19.bsc.es/
[8] https://www.mongodb.com/
[9] https://bioexcel-cv19.bsc.es/api/rest/docs/

membrane-based simulations, and nucleic acids simulations. Most MDDB pilot cases (see D3.1) lie in this category. Projects at this level of federation will share a common data portal, probably with specific types of analyses, not necessarily available at the main MDDB portal. Also, at a lower scale, individual projects, maintained by single research groups may be part of MDDB, provided that metadata is available at the central catalog, and the necessary availability is assured. Table 1 summarizes MDDB envisioned nodes, and the expected interaction with the repository.

Type of Node	Description	MDDB requires	MDDB provides
Computational	HPC center or equivalent	Simulation software MDDB software stack for trajectory analysis and QC	Complete workflow for basic analysis and metadata extraction
Long-term storage	Storage for raw trajectory data	Data insertion and retrieval API	Reference implementation of Data API
Storage/analysis	"Project node" providing a data portal and possibly visualization and analysis environment	Data API compliant with MDDB schema for data and metadata, provision of metadata for population of MDDB central catalog	Reference implementation for database management, data API, and data portal

Table 1: Summary of MDDB prototypical nodes and associated requirements.

2.3 Technical requirements

The above analysis implies the evaluation of a series of technical components necessary to build the MDDB ecosystem. Table 2 summarizes the most relevant ones.

3 SOLUTION ALTERNATIVES

3.1 Data storage and mobilization

Data storage is the main issue when designing a simulation repository. Two levels of storage can be envisioned:

- **Storage of raw trajectory data**. Data obtained from the simulation without any processing, other than possible concatenation of partial runs. Raw trajectories would be largely stored but processed only on a few occasions. Long-term storage nodes would be the expected receivers. Maximum compression is required to minimize storage size, and efficient data transfer solutions are desirable.
- **Storage of processed trajectory data**. Data obtained from the initial processing of raw trajectories and main source for analysis or visualization. This would be held at data analysis nodes associated with fast mobilization facilities. It is expected that data would be very often accessed, either as a whole or as fragments, for downloading or visualization but also as input of analysis. Since access to processed trajectory data is usually done through a dedicated data portal, analysis results should be stored along the trajectory.

Component	Description
Data Storage and mobilization	
Database management system	Handle simulation metadata, processed trajectory data and analysis results
ETL software for data ingestion	Software to process trajectory data and analysis and format them for database loading
Data access API	Interface to the storage system. Data API should provide metadata, analysis results and access to the stored trajectories, being able to provide fragments of such trajectories according to specific time slices or atom sets for easier analysis, and meta-trajectories
Data compression and transfer	
Compression format	Format and required software for trajectory data packaging and compression, including the necessary metadata for simulation reproduction/extension
Protocols/software for fast data transfer	Software solutions for fast data transfer among nodes
Analysis and visualization	
Analysis toolkit	Software toolkit for QC analysis of structures and trajectories, adapted to run in HPC environments.
Specialized analysis toolkit	Software toolkit to perform analysis adapted to the specific nature of the simulations at project
Visualization software	Visualization software deployed in data portals for 3D structures/trajectories and analysis results
Virtual Research Environment	Virtual environment combining data access, analysis, and visualization modules
Interfaces to HPC/HTC	
Software deployment on HPC	Procedures for easy software deployment on HPC environments
User interfaces	User interfaces to set and monitor HPC executions and data transfer

Table 2: Summary of software components required by MDDB ecosystem.

Available examples of simulation data stores are mainly in the second level. The first wide initiative to store MD trajectories and use them to extract general characteristics of protein dynamics should be attributed to V. Daggett who developed the Dynameomics database and server in 2010[10]. In the latest version of this database accessible to external users[11], Dynameomics comprised 30 ns (stored every ps) trajectories of around 800 soluble proteins selected based on their structural diversity and enriched in enzymes and thermophilic proteins. Typical simulation Dynameomics' trajectories and analyses

[10] Van der Kamp MW, Schaeffer RD, Jonsson AL, Scouras AD, Simms AM, Toofanny RD, Benson NC, Anderson PC, Merkley ED, Rysavy S, Bromley D, Beck DAC, Daggett V (2010) *Dynameomics: A Comprehensive Database of Protein Dynamics*. Structure 18(4): 423–35

[11] http://www.dynameomics.org/

were stored in a combination of a relational database management system and an online analytical processing (OLAP) multidimensional system (built on Microsoft SQL Server)[12]. The OLAP implementation offers descriptively rich data retrieval and modeling thanks to the multidimensional expressions query language (mdx). This construction is opposite to retrieve and analyze relationships, as each dimension implies an independent index. All Cartesian coordinates for all the atoms are stored in OLAP cubes, decreasing the storage required by a factor of 5 in comparison to a typical SQL table. At the same period, the first generation of all-purpose MD databases was exemplified by MoDEL (Molecular Dynamics Extended Library)[13]. The first version of the database included a list of 1,595 monomeric, nonhomologous proteins, covering Cluster-90 PDB families. The length of the simulations ranged from 10 to 100 ns, with a total amount of data reaching 20 TB of storage space. MoDEL used a dual approach to store data: a MySQL RDBMS model to store simulation metadata and analyses, and a disk-based raw data repository. The relational database was designed for an efficient retrieval of simulation analyses, defined by a combination of a particular simulation, the structure fragment of interest, and the portion of the trajectory to be analyzed. This scheme made it possible to store a wide variety of results: from a simple collection of trajectory snapshots to a specific combination of analyses done over several parts of the trajectory or restricted to a specific domain. All the simulations and analysis computed in the MoDEL collections were made available to the scientific community from a freely accessible web server[14]. Trajectories were available for download in a compressed PCZ format[15]. In 2016, a second-generation MD database BigNASim (Big data Nucleic Acids Simulations)[16] was built. BigNASim database stored MD trajectories and analyses of nucleic acids simulations. BigNASim's initial release contained around 120 simulations, with more than 200 µs of total simulation time and occupied more than 100 TB of disk storage. An updated version of this DB is being included as an MDDB pilot (see D3.1). The database is based on the combination of two NoSQL engines, Cassandra[17] for storing trajectories and MongoDB[18] to store analysis results and simulation metadata. These two types of databases offer different characteristics that suit very well for the two main sets of data generated by the project: trajectories and analyses. On one hand, Cassandra is a column-oriented database, especially efficient when data can be represented in key-value pairs with simple structures. The simplicity of trajectory data structure, a uniform series of Cartesian coordinates that should be retrieved in well-known groups of data, makes it ideal to be handled by the Cassandra engine. On the other hand, MongoDB is a flexible document-oriented database. MongoDB may store from single values, to 2D or 3D data, or even full-length trajectory videos within a single document. The described structure allows global queries to retrieve meta-simulation analysis or the possibility to download meta-trajectories built using parts of different simulations. These two features are especially

[12] https://learn.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing

[13] Meyer T, D'Abramo M, Hospital A, Rueda M, Ferrer-Costa C, Perez A, Carrillo O, Camps J, Fenollosa C, Repchevsky D, Gelpí JL, Orozco M (2010) *MoDEL (Molecular Dynamics Extended Library): A Database of Atomistic Molecular Dynamics Trajectories*. Structure 18(11): 1399–1409

[14] https://mmb.irbbarcelona.org/MoDEL

[15] Meyer T, Ferrer-Costa C, Pérez A, Rueda M, Bidon-Chanal A, Luque FJ, Laughton CA, Orozco M (2006) *Essential Dynamics: A Tool for Efficient Trajectory Compression and Management*. J Chem Theory and Comput 2(2):251-8

[16] Hospital A, Andrio P, Cugnasco C, Codo L, Becerra Y, Dans PD, Battistini F, Torres J, Goni R, Orozco M, Gelpi JL (2016). *BIGNASim: a NoSQL database structure and analysis portal for nucleic acids simulation data*. Nucleic Acids Res 44: D272-278

[17] https://cassandra.apache.org

[18] https://mongodb.com

interesting for nucleic acids as they allow the user to obtain sequence dependent structural parameters by combining many trajectories. Both databases are linked to a web server[19], which offers the whole set of simulations and their associated analyses in a graphical interface, as well as the ability to download individual trajectories and combined meta-trajectories. The third-generation MD databases, whose development is parallel to MDDB, are not generic, but specific-purpose databases, and they include BioExcel-cv19, ModTox and MoDEL_CNS (see D3.1). The storage methods used in such databases is an evolution of BigNASim's, replacing Cassandra by MongoDB's grid File System (GridFS) for storage and retrieval of large files. Although Cassandra was adequate for trajectory data due to its simple structure, GridFS stores large files in chunks of defined size, allowing similar direct access capabilities and allowing also storing other types of data. Interestingly, GridFS can also provide complete trajectories in the file-oriented way that traditional analysis packages require. Those databases are complemented by a web server interface presenting the trajectories, analyses, and protein properties.

Other databases have been developed and have made the MD simulations accessible and interoperable. The GPCRmd[20] is a comprehensive database and web platform[21] focused on MD simulations of GPCRs and their analyses. MemProtMD[22], also a MDDB pilot, contains MD simulations of around 3,500 membrane proteins embedded in lipid bilayers, and its associated web platform[23] allows the user to find analysis of the simulations, and the resources to set up coarse-grained and atomistic simulations. Cyclo-lib[24] is a database focused only on cyclodextrins, and it contains atomistic MD simulations and trajectory analyses of almost 100 different natural cyclic oligosaccharides. One last example is TMB-iBIOMES[25] a database collecting over 20 ms of all atom MD simulations for over 500 different realizations of the nucleosome. Those databases are not only a collection of MD data, but also web platforms where the user can navigate, visualize analyses, and produce and retrieve data.

3.2 Data compression and transfer

In the previous section, data storage was analyzed for the case where fast data mobilization was required. In this case, data compression was not the issue. Actually, database managers used were responsible for decreasing storage size while allowing quick mobilization. A different scenario is the case of long-term storage, or when data must be transferred among nodes, or from the data providers to the infrastructure. In this case, the prime criterion should be to decrease the size of the data. This has also been one of the concerns of the design of MD data formats (see D1.1). Solutions commented here are largely following the solutions adopted by MD codes themselves.

In principle, there are two strategies for recording MD raw data:

[19] https://mmb.irbbarcelona.org/BigNASim

[20] Rodríguez-Espigares I, Torrens-Fontanals M, Tiemann JKS, Aranda-García D, Ramírez-Anguita JM, Stepniewski TM, Worp N, Varela-Rial A, Morales-Pastor A, Medel-Lacruz B, Pándy-Szekeres G, Mayol E, Giorgino T, Carlsson J, Deupi X, Filipek S, Filizola M, Gómez-Tamayo JC, Gonzalez A, …Selent J (2020) *GPCRmd Uncovers the Dynamics of the 3D-GPCRome*. Nat Methods 17(8): 777–87

[21] https://gpcrmd.org/

[22] Newport TD, Sansom MSP, Stansfeld PJ (2019) *The MemProtMD Database: A Resource for Membrane-Embedded Protein Structures and Their Lipid Interactions*. Nucleic Acids Res 47 (D1): D390–397

[23] http://memprotmd.bioch.ox.ac.uk/

[24] http://cyclo-lib.mduse.com/

[25] Sun R, Li Z, Bishop TC (2019) *TMB-iBIOMES: An iBIOMES-Lite Database of Nucleosome Trajectories and Meta-Analysis*. ChemRxiv doi:10.26434/chemrxiv.7793939.v1.

- Regular spacing (most common): Regular storage of snapshots after a certain number of integration steps (for atomistic simulations typically every 1–10 ps). This approach allows a simple compression of the data (x10^3-10^4), but at the expense of missing fast movements, especially those related to solvent relaxation, which are not taken into consideration, apart from when the user is interested in low-frequency vibrations, or fast movements of water molecules.
- Adaptive schemes: Restart files, instead of trajectories, are stored after a certain number of steps (typically in the multi-nanosecond range). If some interesting phenomena has happened between restart file (i) and (i + t), the simulation is launched again starting at point (i) and new data are collected at shorter intervals. This approach is useful when the user is interested in unlikely events and requires trajectories to be completely reproducible using the restart files, and it is typically used with ultra-long simulations using very fast CPUs that do not have rapid access to the disk (e. g. MD-specific Anton computer). Although this approach is more powerful it implies a change in the usual protocols as the recovery of the trajectory requires running further simulations.

Additionally, a hybrid schema can be followed. It would correspond to standard trajectory recording, perhaps with a longer time step, together with the addition of restart files, recorded much more often than in normal protocols. This would allow them to respawn fragments of trajectories to obtain more detail of the events. In any case, the procedure followed just has to be clearly indicated in the metadata to allow understanding of the stored data.

Regarding the molecular nature of the data, storage of the data is typically done using one of these three approaches:

1. storage of all coordinates, including solvent molecules for the entire simulation (raw trajectories),
2. storage of the dry trajectory, that is, the coordinates along the simulation time of the macromolecule and ligand(s), deleting the solvent, and usually ions (processed trajectories),
3. intermediate storage solution, for example, maintaining all solvent molecules located at less than a given distance from the macromolecule.

It is to be noted that in the last two cases, keeping the original data is necessary to allow future analysis requiring solvent. Alternatively, restart files containing solvent information, allowing to reproduce solvent behavior, could be kept for reusability/reproducibility purposes, since keeping only dry trajectories many interesting aspects of macromolecular dynamics cannot be understood, as they are coupled with the solvent.

Irrespective of the actual contents of the data, compression strategies can be used to reduce the disk space occupied by the simulations. Such strategies can involve:

- lossless compression, where simulation data can be recovered without any alteration,
- methods using data characteristics to reduce size at the expense of losing some of the least relevant data.

Within the first category, traditional Lempel-zip based methods (like gzip) can be applied and are the only possible alternative to text-based trajectory formats like Amber's MDCRD[26]. Amber's

[26] https://ambermd.org/FileFormats.php

recommended data format NetCDF[27] includes zlib-based lossless compression. The most popular trajectory format in GROMACS, XTC[28] uses a compression strategy based on reducing the numerical precision to 9–10 bits instead of 32, while full-precision trajectories could still be saved as TRR format. The recent GROMACS file format, TNG, uses a container-based strategy, which can apply different compression schemes to different parts of the trajectory. The default compression schemes used in TNG are lossy methods, sacrificing precision (in a similar way as XTC) to reduce data size, taking advantage of both space and time proximity. TNG might also accept Zlib as a compression engine, but with lower efficiency.

Alternative strategies for the storage of macromolecular dynamics are based on transformation of the original coordinates, for example, by principal component analysis (PCA)[15]. The original trajectory is projected on the essential deformation space and only the eigenvectors and the projections of every snapshot are kept. If all eigenvectors are considered, the method is exact, as the set of eigenvectors obtained by PCA defines a complete space, but no compression is achieved. Discarding less relevant eigenvectors, large compressions of the dry trajectory is obtained, but at the expense of some distortions in the internal geometry and the loss of information on hydrogen atoms. If the user is interested in the general structural movements of the macromolecule, PCA compression is a useful alternative. PCA compression was the default strategy for trajectory download in the MoDEL database, as it provided very small files easy to transfer. Kumar et al.[29] used PCA strategy, but with time-based windows, combining PCA and discrete cosine transform (DTC), finalized by a zip compression, which allowed them to reach a compression ratio up to 13 with an acceptable information loss. In any case, the loss of information on the solvent environment implicit to PCA compression might be not acceptable in many cases, especially where fine details are required, or where solvent is crucial. Omeltchenko et al.[30] designed a different compression approach, based on the principles of data locality, using octree indexing and sorting atoms accordingly on the resulting space-filling curve. By storing differences of successive atomic coordinates and using an adaptive, variable-length encoding to handle exceptional values, the size is reduced by an order-of-magnitude still maintaining a user-controlled margin of error. Despite the work done with lossy trajectory formats, there is still a need for highly compressed lossless formats.

Compression efforts are not only relevant for storage. One of the less evolved components in modern computers is the network. GEANT, the institution providing the scientific network across Europe has a 100 Gbps maximum capacity, that in theory would support up to 12 GB/s, but connections with North America, for instance, drop to one order of magnitude below, and very few research institutions offer more than a few Gbps of global bandwidth. A typical university site may offer up to 100 Mbps of useful bandwidth, which implies that the transfer of a 100 GB trajectory will require more than 2 hr. With this scenario, the ideal procedure consists in keeping data at their production site, providing on-site analysis software, perhaps within a virtual research environment (VRE) (see below), and transferring only analysis results. This implies that: (1) the data repositories should provide high computational

[27] https://www.unidata.ucar.edu/software/netcdf/

[28] https://manual.gromacs.org/current/reference-manual/file-formats.html

[29] Kumar A, Zhu X, Tu YC, Pandit S (2013) *Compression in Molecular Simulation Datasets*. In: Sun C, Fang F, Zhou ZH, Yang W, Liu ZY (eds) *Intelligence Science and Big Data Engineering* (pp. 22–29). Berlin, Heidelberg: Springer, Berlin, Heidelberg

[30] Omeltchenko A, Campbell TJ, Kalia RK, Liu X, Nakano A, Vashishta P (2000) *Scalable I/O of Large-Scale Molecular Dynamics Simulations: A Data-Compression Algorithm*. Comput Phys Commun 131(1): 78–85

power and the highest available bandwidth, (2) databases should allow programmatic access to the stored information, and (3) databases should facilitate integrated analysis. This model has been tested in the genomics field with moderate success, and in fact, genomics has largely opted for centralized data storage and retrieval in a reduced set of trusted repositories (EBI[31], NCBI[32]), powered with highly efficient networks and computational facilities. In the case of simulations, no equivalent solution exists. Although the equipment performing simulations is powerful, it is not designed to allow efficient external communications. A clear example is supercomputing centers where communication to the outside is blocked, and users cannot keep storage after the assigned project ends. From a technical point of view, some alternatives to the traditional FTP/HTTP-based transmission have been tested. GridFTP[33] usually orchestrated through Globus[34] uses several TCP threads in parallel to get the maximum use of the available bandwidth, allowing for safe partial transmissions. A common solution adopted in the genomics world is Aspera[35] which combines partial transmission with the use of UDP to speed up the transfer. Peer-to-peer protocols such as GeneTorrent[36] have been used in large projects like PCAWG[37,38]. A clear improvement over traditional data transmission would be the use of streaming, which would allow the analysis before a complete download. MDSrv[39] is a popular example of a streaming server created to help remote visualization (see below). However, proper benefit of streaming would require the adaptation of most analysis software, which was originally designed to read from disk files. Direct access to selected parts of the trajectory, available in formats like TNG, NetCDF, and also in database implementations like Bioexcel-cv19 and MoDEL-CNS would also facilitate remote analysis, as only a small part of the data should be transferred.

3.3 Data analysis and visualization

Molecular dynamics simulation can only be useful when a proper analysis of the trajectory data is performed. Analysis tools are a crucial component of the MDDB ecosystem. Analysis results are very relevant for possible MDDB users interested in taking benefit of the insight that simulation gives to the study of macromolecules without doing simulations themselves and also, the main link for the integration of MDDB with other biological data repositories like PDBe-KB[40] (see D3.1). Analysis can be structured in several categories (Table 3 summarizes, as example, the analysis available in the bioexcel-cv19[41]):

[31] https://www.ebi.ac.uk/
[32] https://www.ncbi.nlm.nih.gov/
[33] https://opensciencegrid.org/
[34] https://www.globus.org/data-transfer
[35] https://asperasoft.com
[36] http://annaisystems.com/
[37] Aaltonen LA, Abascal F, Abeshouse A, Aburatani H, Adams DJ, Agrawal N, Ahn KS, Ahn SM, Aikata H, Akbani R, Akdemir KC, Al-Ahmadie H, Al-Sedairy ST, Al-Shahrour F, Alawi M, Albert M, Aldape K, Alexandrov LB, Ally A, …von Mering C (2020) *Pan-Cancer Analysis of Whole Genomes*. Nature 578(7793): 82–93
[38] Yakneen S, Waszak SM, PCAWG Technical Working Group, Gertz M, Korbel JO, PCAWG Consortium (2020) *Butler Enables Rapid Cloud-Based Analysis of Thousands of Human Genomes*. Nat Biotechnol 38(3): 288–92
[39] Kampfrath M, Staritzbichler R, Pérez Hernández G, Rose AS, Tiemann JKS, Scheuermann G, Wiegreffe D, Hildebrand PW (2022) *MDsrv: Visual Sharing and Analysis of Molecular Dynamics Simulations*. Nucleic Acids Res 50(W1): W483–489
[40] https://www.ebi.ac.uk/pdbe/pdbe-kb/
[41] https://bioexcel-cv19.bsc.es/#/help

- **Data processing.** Operations required to process raw trajectories to obtain "processed trajectories". Includes imaging, removal of CoM, removal of solvent.
- **Quality control.** Done to evaluate the quality of the trajectory. This group of analysis will be a required component in the process of MDDB submission.
- **Generally applicable analysis**. Types of analyses that do not depend on the molecular type like correlation matrices, PCA, solvent accessible surface, hydrogen bonding conservation, binding energies.
- **Specialized analysis.** Tools specific to types of molecules like helical parameters in the case of NA simulations, or specific to the kind of simulation, like Potential of Main Force or weight histogram analysis for replica exchange simulations.
- **AI Based analysis.** Analysis based on deep-learning techniques, singular as they may require specialized hardware (e.g. GPUs).
- **User defined analysis.** Analysis not covered by standard packages and requiring user provided codes.
- **Visualization of structures and trajectories.** Required component for data portals, allowing users to assess details of the simulations.

MDDB will be not just a repository but should offer an analysis layer together with the trajectories. This will provide the final user with already digested insight on the simulations' interpretation and how the simulation experiment helps to understand the biological questions. Following the previous classification, we could envision different analyses provided at different levels in the MDDB ecosystems. Data processing and QC analysis are a natural component of the simulation itself and should be naturally organized as a workflow after the simulation trajectory is obtained. Their results will be naturally packed together with the raw trajectories and added to the long-term storage to assure a reference for reproducibility and provenance. Whether this data could be included in the trajectory files themselves or kept separated is a matter of discussion (see D1.1) and MDDB recommendations in that direction will be issued in due time. General-purpose analysis will not be as required, and will be largely provided *a posteriori,* although it is likely that will be also included in the initial analysis workflow, as they are comparatively cheap in terms of computational cost. Specialized analyses and user-defined ones are likely to be performed at data analysis nodes where the specific tools may be deployed on purpose. Visualization will also be offered at this stage, as part of an interactive data portal.

MDDB will provide a reference implementation of a general analysis software stack using standard tools and libraries, and as part of the submission policies, recommendations on QC acceptable thresholds.

3.3.1 Analyses and analysis tools

Table 3 shows a list of the available analysis at bioexcel-cv19 platform classified according to these criteria. Note that most of the analysis is performed with software already provided together with the simulation packages. Indeed, most MD users tend to use the same package for setup, simulation and analysis. Although GROMACS, for instance, opted to offer a complete suite of analysis, other packages rely on tools that are able to manage data from a wider set of data formats. For instance, the ambertools suite[42] (provided free of use by the Amber package) includes CPPTraj and pyTraj for general

[42] https://ambermd.org/AmberTools.php

analysis. Independent libraries like MDAnalysis[43] provide a widely compatible set of analysis while BioExcel Building Blocks[44] (BioBB) provides an umbrella offer of trajectory analyses[45] using both own implementations and the mentioned analysis packages.

Although general-purpose analyses are of general application, specialized analyses are also relevant for MD users. Indeed, bioexcel-cv19 portal[46] (Table 3) provides specific analysis results to put the stored simulations in the context of known mutations and epitopes of SARS-Cov-2. Also, a great part of the offered analysis, albeit of general use, were tailored to account to the interactions of SARS-Cov-2 Spike and Human ACE2 protein, the complex most usually simulated for COVID19. Another clear example is the simulation of Nucleic Acids. One of the simulation databases to be extended as a pilot project (see D3.1), BigNASim[16], is the only available simulation database for this kind of molecule. BigNASim took great advantage of NAFlex[47], a developed analysis and portal combining specific nucleic acids simulation analysis like helical parameters, NMR observables, and Hydrogen Bonds. Specific DNA oriented analyses are available to visualize together with simulation data.

Analysis	Description	Software	Type
Root Mean Square deviation (RMSd)	RMSd against the first frame and against the average structure RMSd analyzes over different protein atom selections: all protein atoms (protein), heavy atoms (protein-h), backbone and alpha carbon (c-alpha).	Gromacs[48]	QC
Template modeling (TM) score	TM-score against the first frame and against the average structure, over alpha carbon atoms.	tmscoring[49] python module.	QC
RMSd per residue	The RMSD between each residue in its original conformation (first frame)	PyTraj[50]	QC
RMSd pairwise	RMSd pairwise analysis computes RMSD between 200 pairs of frames (CA-only) along the trajectory. The analysis is done for the whole structure and for the interface residues in each interaction.	PyTraj	QC
Radius of gyration	Radius of gyration of a molecule and the radii of gyration about the x-, y- and z-axes, as a function of time. The atoms are explicitly mass weighted.	Gromacs	QC

[43] https://www.mdanalysis.org/

[44] https://mmb.irbbarcelona.org/biobb

[45] https://github.com/bioexcel/biobb_analysis, https://github.com/bioexcel/biobb_dna, https://github.com/bioexcel/biobb_flexdyn, https://github.com/bioexcel/biobb_flexserv

[46] Beltrán D, Hospital A, Gelpí JL, Orozco M (2024) *A New Paradigm for Molecular Dynamics Databases: The COVID-19 Database, the Legacy of a Titanic Community Effort*. Nucleic Acids Res 52(D1): D393–403

[47] Hospital A, Faustino I, Collepardo-Guevara R, Gonzalez C, Gelpi JL, Orozco M (2013) *NAFlex: A Web Server for the Study of Nucleic Acid Flexibility*. Nucleic Acids Res 41(Web Server issue): W47-55

[48] https://manual.gromacs.org/current/reference-manual/analysis/analysis.html

[49] https://pypi.org/project/tmscoring/

[50] https://amber-md.github.io/pytraj/latest/index.html

Principal Component Analysis (PCA)	PCA calculates and diagonalizes the (mass-weighted) covariance matrix to obtain a set of orthogonal eigenvectors and its associated eigenvalues. Then a set of conformations are projected into the eigenvectors 2D space (e.g. Principal Component 1 and Principal Component 2) for the relevant eigenvectors	Scikitlearn + MDTraj	General
Solvent accessible surface (SAS)	SAS computes the area of contact between the solvent and each residue, along time.	Gromacs	General
Distance per residue at interfaces	The distance per residue analysis computes distance between each pair of residues in each interaction interface. Distance average and standard deviation are also calculated.	PyTraj	General
Electrostatic potential surface (visualization)	Surface colored according to the atomic charges.	NGL[51]	General
Hydrogen bonds	Hydrogen bonds between different interacting components using standard geometry-based detection.	PyTraj	General
Interaction energies	Electrostatic and Van der Waals interaction energies between interacting components of a complex. Energy averages per residue.	CMIP[52]	General
Pockets	Pockets analysis searches for cavities in the structure along the trajectory.	MDPocket[53]	General
Mutations (SARS-Cov-2 specific)	Entropies for each residue were obtained from the Nextstrain web page, which relies on the GISAID database.	Nextstrain[54]	Specific
Epitopes (SARS-Cov-2 specific)	Extracted from a systematic search on antibody-containing SARS-Cov-2 PDB entries.	In house	Specific

Table 3: Analysis available at bioexcel-cv19 Platforms

3.3.2 Visualization and Virtual Research environments

Several visualization engines have been used on different repositories so far, largely following the evolution of this kind of software. MoDEL[13] used the now obsolete JMol engine[55], this was replaced by its Javascript based version (JSMol) in BigNASim[16], which proved a limited support for trajectory data. Most recent servers like bioexcel-cv19 and the BioExcel Building Block workflows (BioBB-Wfs)[56] servers

[51] https://nglviewer.org/
[52] https://mmb.irbbarcelona.org/gitlab/gelpi/CMIP
[53] http://fpocket.sourceforge.net.
[54] https://nextstrain.org/sars-cov-2/
[55] https://jmol.sourceforge.net
[56] Bayarri G, Andrio P, Hospital A, Orozco M, Gelpí JL (2022) *BioExcel Building Blocks Workflows (BioBB-Wfs), an Integrated Web-Based Platform for Biomolecular Simulations*. Nucleic Acids Res 50(W1): W99–107.

rely on NGL[57] in combination with MDSrv[39], for visualization of structures and trajectories. MDDB will follow this trend and will draw inspiration from most modern web-based, lightweight, open-source visualization tools like Mol*[58]. Their combination with MDSrv offers a web-based tool designed to enhance collaborative research by providing non-experts with easy and quick online access to molecular dynamics (MD).

The combination of efficient visualizers, analysis tools and a fast mobilization storage will allow to build not only a traditional data portal, but a Virtual Research Environment providing:

- A remote/local way to visualize trajectories.
- A way to run analyses.
- An overview of metadata highlighted on 3D structure (from MDDB and from data integration)
- Ability to explore the results in one place.

This kind of complete functionality has not been provided so far by any simulation repository, although the availability of simulation related tools in workbenches like Galaxy[59] and the BioBB's library has been ported to Galaxy and to openVRE[60].

To achieve these objectives and provide MDDB users with a comprehensive visualization and analysis platform, several key components will be necessary:

- Modular molecular viewer: implement/use a flexible molecular viewer such as molstar, UnityMol3D[61] or VTX[62] that allows users to visualize MD trajectories. This viewer should support file formats used by MDDB and provide a scripting environment for running custom analyses directly within the viewer, and support for viewing PDF or image files for supplementary information and analysis visualizations.
- Remote work infrastructure: establish a robust infrastructure capable of supporting remote work, ensuring seamless access to the visualization and analysis platform from any location, allowing original data to remain in place. This infrastructure should prioritize security, scalability, and reliability to accommodate varying user needs.

By incorporating these elements into the MDDB platform, users will have a centralized hub for visualizing trajectories, conducting analyses, accessing metadata from MDDB and integrated data sources, and exploring results—all within a single, user-friendly interface. This integrated approach enhances efficiency, collaboration, and productivity for researchers working with MD simulations.

3.4 Interfaces to HPC/HTC infrastructures

The traditional use of large-scale infrastructures in MD is largely restricted to command line scripts chaining together modules of a given simulation software. Also, access to HPC facilities is restricted to specific network protocols, most usually SSH terminals, and the management of executions is based on a batch queuing system. The procedures of a HPC center cannot be altered. MDDB will need to adapt to such requirements, although a series of helper components can be leveraged.

[57] https://nglviewer.org/
[58] https://molstar.org/
[59] Bray SA, Lucas X, Kumar A, Grüning BA (2020) *The ChemicalToolbox: Reproducible, User-Friendly Cheminformatics Analysis on the Galaxy Platform*. J Cheminform 12(1): 40
[60] https://github.com/inab/openVRE
[61] https://sourceforge.net/projects/unitymol/files/
[62] https://vtx.drugdesign.fr/

Regarding execution, the BioExcel Building Blocks library offer uniform packaging system that makes compatible with most common workflow management systems, while their compatibility features allows the seamless building of complex workflows[63]. The combination of such workflows with execution managers like PyCOMPSs[64] has allowed us to execute, for instance, a series of simulations for a wild-type protein and a series of sequence variants showing good scalability up to 65,536 cores in a single calculation[65]. A production application of this approach has allowed us to evaluate highly accurate binding energies between SARS-Cov-2 mutants and human ACE2[66].

Regarding access, the possibility of managing HPC executions from a friendlier environment, possibly with a GUI, has been largely demanded by less-expert users. The Fenix[67] initiative was the most advanced one to provide a uniform and friendly access interface to HPC supercomputers. It was used in the ICEI project[68], part of the Human Brain initiative. However, Fenix requires an active participation of HPC managers, which makes a general implementation difficult. Other large-scale infrastructure (HTC) like EGI, do allow for friendly interfaces, as they are largely cloud-based, however, their infrastructure, highly distributed, makes them less efficient for large-scale simulations. The most practical approach is to build a transparent interface allowing the management of HPC queueing systems by simple impersonation of the HPC users through the accept connection channels. An initiative in this respect is the biobb_remote[69] module included in the BioBB library. Such module is a python interface allowing managing HPC simulation jobs programmatically, through a ssl communication channel, compatible with most HPC access protocols. It is being used in the BioBB-Wfs server[70] to automatically derive longer executions of a simulation project to HPC computers, provided that user has the necessary access rights. Biobb_remote is being implemented also in the openVRE platform[71] .

Regarding installation and deployment, different existing technologies are helping in these processes. Conda[72] is a package and environment management system that quickly installs, runs, and updates packages and their dependencies. Conda easily creates, saves, loads, and switches between environments on a computer. Initially created for Python programs, it is now able to package and distribute software for any language. Nix[73] is a package management and system configuration tool to make reproducible, declarative and reliable systems. In a very similar way to Conda packaging tool, an isolated environment with the needed dependencies is created, shareable and portable across machines. Although having more than 80,000 available packages, it is mainly focused on Unix systems

[63] https://mmb.irbbarcelona.org/biobb/workflows

[64] https://pypi.org/project/pycompss/

[65] Ejarque J, Andrio P, Hospital A, Conejero J, Lezzi D, Gelpi JL, Badia RM (2022) *The BioExcel Methodology for Developing Dynamic, Scalable, Reliable and Portable Computational Biomolecular Workflows*. 2022 IEEE 18th International Conference on E-Science (e-Science) 357–66

[66] Wieczór M, Genna V, Aranda J, Badia RM, Gelpí JL, Gapsys V, de Groot BL, Lindahl E, Municoy M, Hospital A, Orozco M (2023) *Pre-Exascale HPC Approaches for Molecular Dynamics Simulations. Covid-19 Research: A Use Case*. WIREs Computational Molecular Science 13(1): e1622

[67] https://www.fenix-ri.eu/

[68] https://www.fz-juelich.de/en/ias/jsc/projects/icei

[69] https://github.com/bioexcel/biobb_remote

[70] https://mmb.irbbarcelona.org/biobb-wfs/

[71] https://github.com/inab/openVRE

[72] https://docs.conda.io/en/latest/

[73] https://nixos.org/

(Linux, MacOS-Darwin). Conda and Nix, despite having a wide adoption in the scientific community, are not suitable for HPC systems, mainly due to a lack of focus on performance (packages are usually pre-built generic binaries). Different packaging tools are available for easing the installation of software/environments in HPC systems. EasyBuild[74] is a software build and installation framework that allows you to manage (scientific) software on HPC systems in an efficient way. It is widely used and one of the preferred tools for HPC centers. It is currently being used in the first deployment of software for some of the new EuroHPC supercomputers (e.g. Lumi, Meluxina). Spack[75] is another HPC-focused package manager for acquiring, building, and managing HPC applications as well as all their dependencies. It is especially useful for building and maintaining installations of many different versions of the same software. Like other similar frameworks, it is associated with a repository of both source-code and binary packages. The Spack system is also able to create customizable environment modulefiles for each built package. It is quickly gaining track and is already available in some of the new EuroHPC supercomputers (e.g. Lumi, Meluxina), in popular cloud providers (e.g. AWS, Azure), and is the chosen packaging technology under the new EBRAINS[76] collaborative infrastructure. Software container technology is also expanding. A large number of container solutions are now available (e.g. Vagrant, Containerd, Ubuntu Linux daemon (LXD), ZeroVM, Podman, Apptainer), most of them relying on the Open Container Initiative (OCI). In spite of all these alternatives, Docker and Singularity containers remain the most relevant container technologies thanks to its large community and ease of use.

MDDB will explore the indicated strategies to incorporate them into the provided software stack, on one side, and to the analysis platforms to facilitate analysis requiring large computational resources.

4 CONCLUSIONS AND ROADMAP

Building the technical infrastructure of MDDB is a rather complex task. A large number of components, as seen, have to be built and combined to provide an integrated experience for the different user types and the different nodes participating in the federation. Fortunately, this does not start from scratch and a significant experience gained in several previous projects, either in the management of simulation repository or in the management of an established repository like PDBe, can be leveraged.

MDDB will largely follow the technical evolution referred to in the above sections. The most complex type of node, the data storage/analysis one will largely build on the experience of the most recent repository bioexcel-cv19 for data storage and mobilization, but also for the generation of trajectory processing and analysis software. Experiences in the management of large performance workflows gained from the BioExcel Center of Excellence project[77] and its software library, the BioExcel Building Blocks[78], will help in generating the necessary software stack. Data management regarding data and metadata schemas, and compression formats will be developed in the context of MDDB WP1, agreed and made available to the community. Data analysis and visualization tools will be tested with the collection of pilot use cases presented in WP3. Conversation with HPC centers will help to determine their availability as simulation data producers, and as long-term storage facilities. In this respect, MDDB

[74] https://github.com/easybuilders/easybuild
[75] https://spack.readthedocs.io/en/latest/
[76] https://www.ebrains.eu/
[77] https://bioexcel.eu
[78] https://mmb-irbbarcelona.org/biobb

will provide the necessary software stack and access interfaces to help their integration in the MDDB ecosystem.

The immediate roadmap for the development of a prototype implementation of MDDB technical infrastructure will be as follows:

Q1 2024. Establishment of relevant working groups, including discussions about analysis software stack (metadata, file formats, data integration, data quality) and technical/hardware requirements.

Q2 2024. Accurate analysis of existing bioexcel-cv19 software stack, and adaptation to a federated strategy. Deployment of mddb-abc, and mddb-pk pilot databases and data portals.

Q3 2024 Test installation of a data analysis node (candidate Oxford University facilities)

Q1 2025. Test deployment of a VRE connected to the master MDDB database.

Q2 2025. Update on the evaluation of requirements and technical alternatives using the feedback obtained from the pilot use cases.

Q3 2025. Potential test installation of new data analysis nodes in 2-3 additional locations.

Q1 2026. Whitepaper on the MDDB federated infrastructure.

5 ANNEXES

5.1 GLOSSARY OF TERMS

Data

Project:	Collection of simulation trajectories with a common scientific purpose
System:	Macromolecular system being simulated/analyzed, including all non-solvent components
System modifications:	Modifications made to the initial systems to fulfill project requirements, e.g. sequence variants, added/removed ligands
Solvent:	Solvent, ions and components added during simulation setup
Simulation conditions:	Set of conditions used on a given simulation (e.g. temperature, thermostats)
Topology:	Complete molecular structure (with modifications), force-field parameters, restrains, as used in a simulation (e.g. a GROMACS TPR file or an AMBER FRCMOD file)
Trajectory:	Result of an individual simulation run, possibly part of a project or replica set (e.g. a GROMACS XTC file)
Raw Trajectory:	Complete trajectory as obtained from the simulation run
Processed Trajectory:	Trajectory data after manipulations like removal of solvent, PBC imaging, etc
Trajectory format:	Documented format used to hold trajectory data & topology (e.g. NetCDF, XTC, TRR)
Trajectory replica set:	Collection of replicated trajectories, sharing a common topology, but differing on coordinates due to different initial coordinates, to different random seed or simple stochastic divergence
Trajectory fragment:	Part of a trajectory made due to a specific purpose, can be a time-slice of the original trajectory, a sub-trajectory of part of the system
Meta-trajectory:	Ensemble of consistent trajectory fragments, e.g. protein active sites from different simulated trajectories, dinucleotides simulated with different flanking sequences, etc

Data Hierarchy

Project

- |- Local identifier
- |- MDDB identifier
- |- Trajectory replicaset
  - |- Topology
  - |- Trajectory 1
  - |- ….
  - |- Trajectory n
    - | - QC analysis
    - | - Analysis x (N)
  - |- Replicaset meta-analysis x (N)
- |- Meta-trajectories

Analysis:	Any calculation made on trajectory data
QC Analysis:	Analysis performed to establish simulation quality. Implies an acceptance threshold.
General purpose analysis:	Type of analysis that can be done irrespective of the molecular type
Specific analysis:	Type of analysis restricted to a given molecular type or simulation strategy
Meta-Analysis:	Consistent group of analyses made on several trajectories.
Local identifier:	Unique Identifier assigned to a project/trajectory by their original producer
MDDB identifier:	Unique and permanent identifier assigned to a project/trajectory on submission to MDDB

Software

Simulation software:	Software suite providing the means for MD setup/simulation/analysis (e.g. GROMACS)
Modeling software:	Software suites providing structure manipulation, format conversion, etc
Helper software:	Software components that help users for configuration and execution of simulation software (e.g. BioBBs)
Analysis software:	Software providing trajectory analysis (e.g. a RMSd calculator)
Visualization software:	Software meant to provide a graphical interface to 3D structures and simulation (e.g. Mol*, NGL)

Virtual Research Environment:	Software environment providing access to data and software of all kinds in a single space, allowing users to perform structure preparation, simulation setup or analysis
Data API:	Restful API to provide programmatic data access or upload

Types of federation nodes

Central MDDB Federation node:	Holds simulation metadata and provides search and browse facilities, provides MDDB identifiers and keeps track of data provenance, holds the core of MDDB management and provides the main user interfaces
Local data node:	Local site holding a dataset of simulations, typically from a given scientific study, maintained by data providers, limited computational capabilities
Long term storage node:	Computational center allowing for long-term storage but with limited computational resources (e.g RES Data), meant for archival purposes, retrieval on demand but unable to perform analyses.
Computational & Data node:	Provides computational resources for analysis and possibly simulation and allows for (limited) long term data storage (e.g. KTH, IRB)
HPC center:	Site providing resources for simulation and analysis but lacking long term storage. (e.g BSC)

Persona

Simulation provider:	Researcher using simulation for addressing scientific use cases. Make use of own or third-party facilities to generate and analyze trajectory data. Generates data and metadata.
Data holder/manager:	Manager of a MDDB storage site, communicates with central MDDB and simulation providers
MDDB Federation manager:	Managers at central MDDB, maintains central infrastructure
Simulation consumer:	Researcher willing to download data or perform selected analysis

Data life cycle

Pre-MDDB operations:	Done outside MDDB infrastructure, eventually using MDDB guidelines and software solutions. Building of collections of 3D structures (PDB, user provided, other), experiment and structures setup and trajectory obtention, predefined analysis, QC and metadata generation. Data and Metadata not yet available.

Local Repository upload:	Data/metadata upload to a local database infrastructure, eventually (but not necessarily) using MDDB recommended DB structure. Search and browse and eventually analysis data is available. Trajectory data can be accessed using local protocols.
Simulation submission & Metadata collection:	Metadata uploaded to central MDDB. Permanent identifier issued and project added to MDDB for browse and search. Additional analyses performed, QC to check publication feasibility according to MDDB policies.
Publication:	After approved submission, the project is available for external users as an MDDB entry. Analysis data could be obtained from local or central sites. Trajectory data can be obtained from data sites following MDDB protocols.