Transition towards environment friendly consumer products by co-creation of an oxidoreductase foundry

GA no 101000607
Research and Innovation Action (RIA)
Start date: 1st June 2021. End date: 31st May 2025

D2.1. Strategy for BioContainer implementation and modules to develop

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101000607

Deliverable log

Deliverable number	D2.1
Deliverable title	D2.1. Strategy for BioContainer implementation and modules to develop
Description of deliverable	This deliverable includes information on the protocols for implementing the BioContainers to be developed in OXIPRO as well as an initial list of containers.
Deliverable type – X on relevant	___Website, patents filing, etc. ___ORDP Open Research Data Pilot _X_Report ___Demonstrator ___Ethics ___Other
WP number	WP2
Deliverable due date	30-11-2021
Submission date	30-11-2021
Dissemination levels – X on relevant	_X_Public __Confidential, only for members of the consortium (including the Commission Services)
Lead beneficiary-organisation	BSC
Responsible person	Victor Guallar (BSC)
Name and affiliation contributing persons	Victor Guallar (BSC), Joan Gilabert (BSC)
Name and affiliation internal reviewers	Víctor Guallar (BSC), Gro Bjerga (NORCE), Marco Fraaije (RUG), Pål Puntervoll (NORCE)

Document history and changes

Version	Date	Author	Description
v1.0	09-11-2021	Victor Guallar	First draft to be reviewed
v2.0	28-11-2021	Victor Guallar	Final draft

1. Summary

The purpose of this document is to provide a short introduction to Biocontainers (in the form of BioBBs), the protocol to develop them and an initial list of the ones to be built. Recall that this list will be upgraded along the project when new requirements at the Bioinformatics/Modelling level originate. Thus, this is a dynamic document, meaning that many changes can be and will be added to it during the extent of the project.

2. BioBBs

Software for biomolecular simulations is typically complex and computationally expensive. Such programs are usually compiled to a particular machine in order to maximise the performance, however, this makes the distribution and installation processes more difficult. Additionally, running the software requires careful setup and initialisation. These factors have prevented the community of users from adopting tools like workflow managers such as

Galaxy or KNIME to automate pipelines as it is commonly done in bioinformatics. In response to that, initiatives like the BioExcel Building Blocks (BioBBs) attempt to bring the advantages of automated workflows to typical biomolecular simulations.

BioBBs are designed following the FAIR principles (Findability, Accessibility, Interoperability and Reusability). The findability of the modules is ensured by making their source code available in a github repository. Accessibility and reusability are provided through accessible packaging and containerisation. The modules (written in the python programming language) are installable through the Python Packaging Index (Pypi) and Bioconda, which automatically install the necessary dependencies. Container technologies (such as Docker or Singularity) can be thought of as light-weight virtual machines that contain a piece of software along with all its dependencies, allowing easier distribution, use and reproducibility. Finally BioBBs provide interoperability by designing a layer of abstraction with a common interface for outputs and inputs, facilitating the construction of pipelines by chaining different BioBBs together.

3. Protocol for developing a BioBBs

The process of creating a new BioBB starts by identifying the software that it will encompass, along with its inputs, outputs and optional parameters. Once this information is collected, the next task is to process the inputs and set up an invocation command for the software of interest, which is typically called through the command line tools provided by the already existing BioBB ecosystem. During this process, the developer usually ensures that the software can be found either through a local installation (manually or through a dependency system such as conda) or via a container. After this, the implementation is tested and properly documented and the new BioBB is published to different channels: Pypi, Bioconda, Quay.io (docker container) or singularity hub.

4. List of BioBBs to be developed

We are currently testing the development of BioContainers with: i) a modeling software involving the Protein Energy Landscape Exploration (PELE) software and other techniques that use it; ii) a text mining container. The first module will allow the user to run biomolecular simulations to , for example, explore possible binding sites for a given substrate, run computational saturated mutagenesis (satumut) or create additional catalytic centers to an enzyme (plurizymer). The second module will allow retrieving data (papers, annotations, etc.) from the web and main databases using (and tailored to) enzymatic keywords. Text mining modules are developed in collaboration with the text mining unit at BSC, managed by Martin Krallinger.

Other BioBBs will provide access to additional software developed at BSC, such as SCOT, a Random Forest based Machine Learning meta predictor that combines the estimations of 8 already published protein stability predictors (MAESTRO, CUPSAT, AUTOMUTE-SVM and AUTOMUTE-TR, FOLDX, INPS3D, MUPRO and I-MUTANT), or Enzyminer which is an ensemble classifier that predicts the promiscuity of an esterase according to physicochemical properties and a PSSM (position specific scoring matrix). We also plan to build BioBBs for other tools commonly used in enzyme engineering such as the Framework for Rapid Enzyme Stabilization by Computational libraries (FRESCO), a tool for computational improvement of an enzyme thermostability or techniques based on hidden markov models used to analyse metagenomics data and extract sequences of interest. A list of the BioBBs that we aim at developing is listed in Table 1.

We should emphasize that module development is performed in collaboration with the group of Salvador Capella, responsible for the technical coordination of **ELIXIR** in Spain,

5. Availability of containers

As stated in the DMP, deliverable 8.2, all software developed in the form of Biocontainers will be public available from different channels, such as Pypi, Bioconda, Quay.io (docker container) or singularity hub. Some implementations, such as the ones using the PELE or FRESCO software, might require for additional license agreements (and/or data access) from their vendors.

Table 1. Initial list for BioBBs development

Original Software	Description	Developer
PELE	Enzyme-Substrate induced fit modeller	BSC
SatuMut	Saturated mutagenesis module	BSC
DEMut	Directed evolution module	BSC
EnzTexter	Text mining with specific enzymatic keywords	BSC
DataFetch	Data module to extract/prepare data from enzymology experiments	BSC/NORCE
PluriZymer	Built a plurizyme from a given enzyme	BSC
SCOT	Thermostability module by Consensus analysis	BSC
EnzyMiner	Machine Learning sequence ensemble classifier	BSC
HMMER	Hidden Makov Model (HMM) module for sequence exploration	NORCE
HMMER + ML	A module combining HMM with machine learning for biopropspecting	NORCE
FRESCO	Thermostability predictor	RUG
CASCO	Activity Predictor	RUG
GANe	Enzyme generative module	BSC
TLG	Defined tag-containing loops predictor into a carbohydrate oxidase	RUG