Interoperable and reproducible biomolecular simulations workflows using BioExcel building blocks (BioBB)

Exploring Biomolecular Modeling & Simulations 9-10/04/2025

EuroCC4 & BioExcel training workshop

Adam Hospital

adam.hospital@irbbarcelona.org

Agenda

THURSDAY, 10 APRIL

10:00 → 11:00	**Interoperable and reproducible biomolecular simulation workflows using BioExcel Building Blocks (BioBB)** **Speaker**: Dr Adam Hospital (Institute for Research in Biomedicine in Barcelona (IRB-Barcelona) and Spanish National Institute of Bioinformatics (INB, ELIXIR-ES))	1h
11:00 → 11:30	**Coffee Break**	30m
11:30 → 13:00	**Hands on - Interoperable and reproducible biomolecular simulation workflows using BioExcel Building Blocks (BioBB)** **Speaker**: Dr Adam Hospital (Institute for Research in Biomedicine in Barcelona (IRB-Barcelona) and Spanish National Institute of Bioinformatics (INB, ELIXIR-ES))	1h 30m
13:00 → 14:00	**Lunch Break**	1h
14:00 → 15:00	**Fundamentals and practical use cases of free energy calculations with PMX** **Speaker**: Dr Sudarshan Behera (Max Planck Institute for Multidisciplinary Sciences, Goettingen)	1h
15:00 → 15:30	**Coffee Break**	30m
15:30 → 17:00	**Hands on - Fundamentals and practical use cases of free energy calculations with PMX** **Speaker**: Dr Sudarshan Behera (Max Planck Institute for Multidisciplinary Sciences, Goettingen)	1h 30m

Session 1 (lecture):

Interoperable and reproducible biomolecular simulation workflows using BioBB

- Workflows & Biomolecular workflows
- BioBB library
- BioBB workflows
  - Demonstration workflows (JN)
  - Pre-exascale workflows (HPC)

Session 2 (hands-on):

Hands-on session on BioBB workflows:

- BioBB demonstration workflow tutorial: GROMACS Protein-ligand complex MD setup
- Quick view on BioBB workflow collection
  - Jupyter Notebooks
  - Pure Python (High Throughput)

bioexcel

Molecular Modeling and Bioinformatics

Theoretical study of biological systems

IRB BARCELONA
INSTITUTE FOR RESEARCH IN BIOMEDICINE

Barcelona

Dr. Modesto Orozco

Unit: Mechanisms of Disease
Position: Group Leader
Office: EMPBC33-34
Telephone: +34 93 40 37156
Email: modesto.orozco@irbbarcelona.org

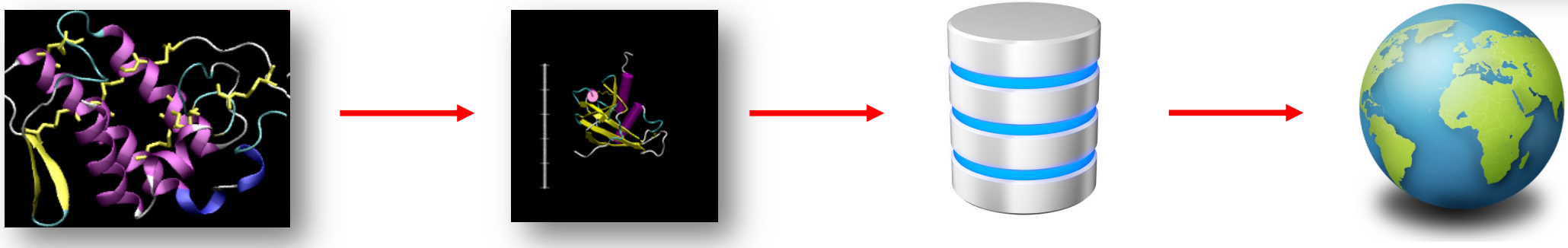
ERC Advanced Grant, Professor (Biochemistry and Molecular Biology Dept. - UB)

Dr. Adam Hospital

Unit: Mechanisms of Disease
Position: Research Associate
Office: EMPBC33-34
Telephone: +34 93 40 37155
Email: adam.hospital@irbbarcelona.org

Genis Bayarri

Unit: Mechanisms of Disease
Position: Web Developer
Office: EMPBC33-34
Telephone: +34 93 40 37155
Email: genis.bayarri@irbbarcelona.org



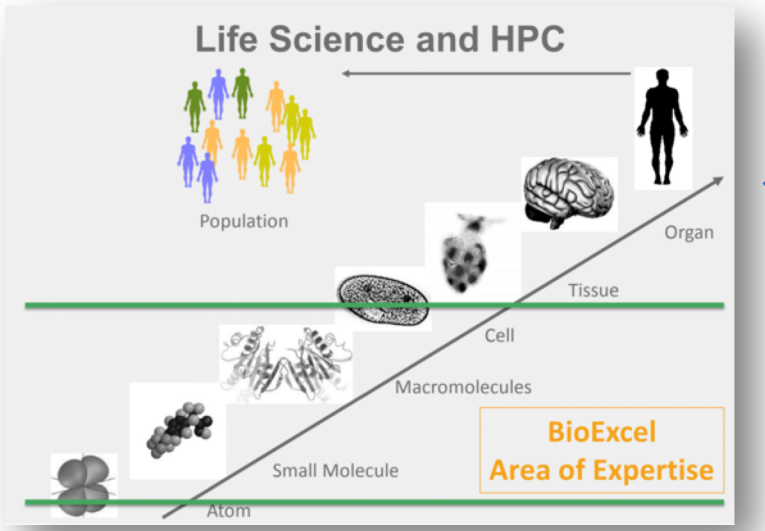
bioexcel

https://www.irbbarcelona.org/en/research/molecular-modelling-and-bioinformatics
https://mmb.irbbarcelona.org/www/

bioexcel

Centre of Excellence for Computational Biomolecular Research

A central hub for biomolecular modelling and simulations



Enabling **better science** by:

- **Improving** the **performance** and **functionality** of **key applications**
- Providing **support** to **non-experts** and **advanced** users
- Developing **user-friendly computational workflows**

KTH VETENSKAP OCH KONST

MAX-PLANCK-GESELLSCHAFT

Universiteit Utrecht

IRB BARCELONA INSTITUTE FOR RESEARCH IN BIOMEDICINE

MMB

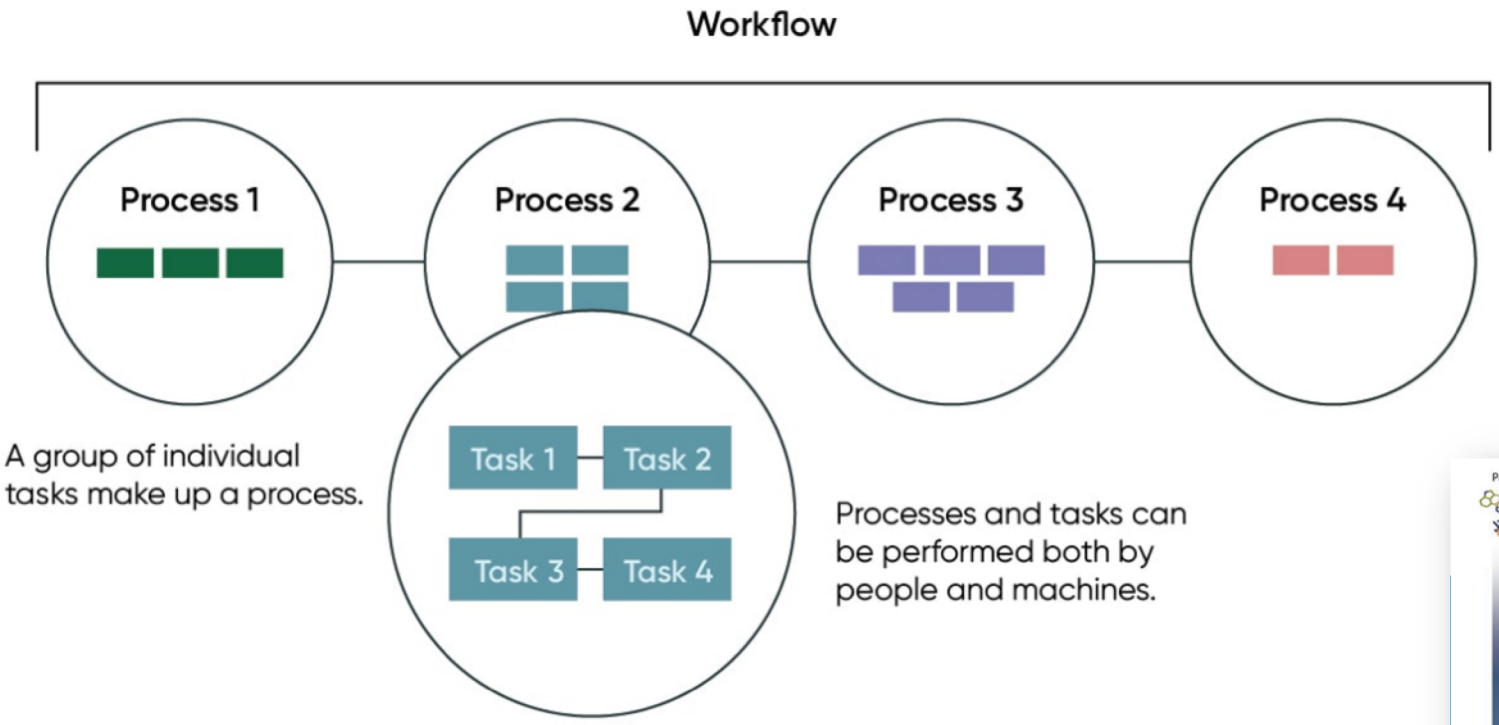
BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

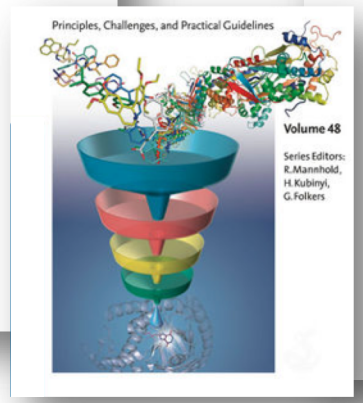
CSC

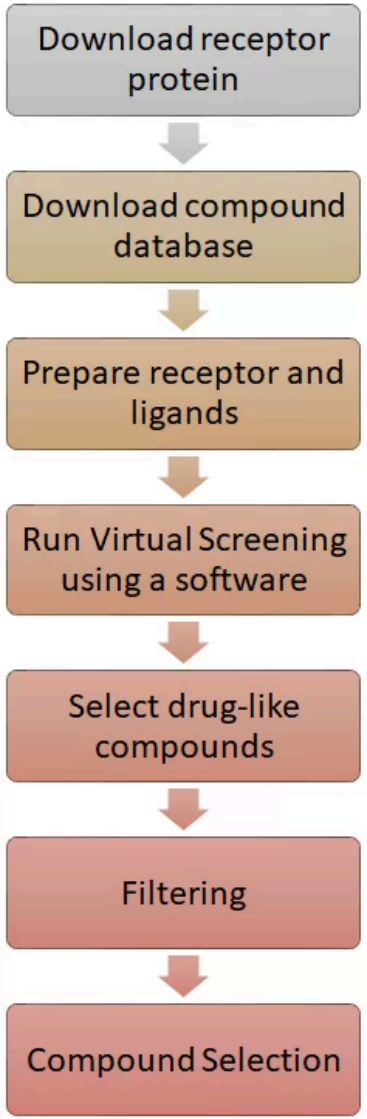
NORMAN CONSULTING

What is a workflow

A workflow is a group of interdependent processes and tasks that achieve a specific business outcome.





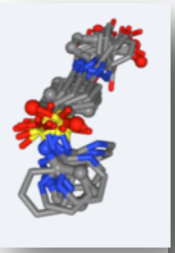


bioexcel

Virtual Screening Workflow

bioactive compounds

BioActive Compounds pipeline: a platform designed to efficiently generate **bioactive conformers** and speed up the **drug discovery process**.



http://mmb.irbbarcelona.org/BCE/

Sanja Zivanovic et al., J. Chem. Theory Comput. 2020, 16, 10, 6586–6597

Input Ligand
3-letter Code
SMILES
PDB File

Generate structure
OpenBabel

Ligand Parameterization

Step 1. Ligand Parameterization.
Adding Hydrogens, energetically minimizing the system, and generating parameters.

- **Adding Hydrogens** — OpenBabel
- **Energetically minimizing system** — Obminimize
- **Ligand parameterization** — ACPype

MD System Setup

Step 2. MD System Setup.
Building topology and solvent box.

- **Building GROMACS Topology** — GROMACS
- **Building System Box** — GROMACS editconf
- **Solvating the System** — GROMACS genbox

Hamiltonian Replica Exchange Simulations (Equilibration Phase)

Generate Hamiltonian REMD equilibration configuration files — GROMACS + plumed

Step 3. Hamiltonian Replica Exchange Simulations (Equilibration Phase).
Energetically equilibrating the system (ligand + solvent), for N replicas.

Replica 1	Replica 2	Replica N
Energy equilibration — GROMACS grompp	Energy equilibration — GROMACS grompp	Energy equilibration — GROMACS grompp
Equilibration MD run — GROMACS mdrun	Equilibration MD run — GROMACS mdrun	Equilibration MD run — GROMACS mdrun

Hamiltonian Replica Exchange Simulations (Production Phase)

Generate Hamiltonian REMD equilibration configuration files — GROMACS + plumed

Step 4. Hamiltonian Replica Exchange Simulations (Production Phase).
Running Hamiltonian REMD simulations for N replicas.

Replica 1	Replica 2	Replica N
Energy equilibration — GROMACS grompp	Energy equilibration — GROMACS grompp	Energy equilibration — GROMACS grompp
Equilibration MD run — GROMACS mdrun	Equilibration MD run — GROMACS mdrun	Equilibration MD run — GROMACS mdrun

Trajectory Clusterization

Step 5. Trajectory Clusterization.
Extract representative conformations from Hamiltonian REMD generated trajectory.

- **Trajectory Clusterization** — GROMACS cluster
- **Trajectory Clusterization post-process** — Dihedral-clustering

Quantum Mechanics

Step 6. Quantum Mechanics.
Computing QM energy using 2 levels of theory: Hartree Fock (HF/3-21G) and B3LYP (B3LYP/6-31G).

- **QM Hartree Fock HF/3-21G** — Gaussian
- **QM B3LYP B3LYP/6-31G*** — Gaussian

Output
- *QM energy conformer 1*
- *QM energy conformer 2*
- *QM energy conformer 3*
- *QM energy conformer N*

Biomolecular simulation workflows are usually built from a number of **tools** performing different **tasks.**

- ***Molecular Structure File format conversions***
- ***Structure Modelling***
- ***Molecular Dynamics***
- ***Quantum Mechanics***
- ***QM / MM***
- ***Trajectory analyses***
- ***Docking***
- ***Free energy***
- ***Ligand parameterization***
- ***Cheminformatics***
- ***Data analytics***

SCHRÖDINGER.

BIOVIA

MOE

Tasks performed by …

FAST. FLEXIBLE. FREE.
GROMACS

AMBER

AmberTools19

OpenMM

bioinfo
mutagenesis
nmr
xl-ms
saxs
ADDOCK
High-Ambiguity Driven Docking

NAMD
Scalable Molecular Dynamics

VMD
Visual Molecular Dynamics

CPMD
QM/MM

ACPYPE

SMILES, InChI
PDB, Mol2
SD File, CML
Gaussian
CIF...
MMFF94
UFF
GAFF
Fingerprints
Charges
Descriptors
Operations
Depiction
Conformers
Open Babel
Databases
Drug Design, QSAR
Computational Modelling
Software
Web Services
File Conversion
Substructure Search
C++
Python, Java
Ruby, Perl, C#

RDKit

CP2K

pmx

MDTRAJ

Gaussian, Inc.

Bio3D

MD
ANALYSIS

AutoDock 4

Modeller
Program for Comparative Protein
Structure Modelling by Satisfaction
of Spatial Restraints

TensorFlow

scikit
learn

bioexcel

And many many more…

Biomolecular workflows: challenges

Shell script:

- ***Step 1***
- ***Step 2***

- ***Step n***

```
# MD SIMULATION (part 1, of 50 ns)
cp /gpfs/scratch/bsc23/bsc23513/EGFR/Gromacs/mdp/md.mdp .
gmx_mpi grompp -f md.mdp -c ../EQ/npt6/WT_apo_npt6.gro -p ../topology/WT_apo.top -r ../EQ/npt6/WT_apo_npt6.gro -t ../EQ/npt6/WT_apo_np
gmx_mpi convert-tpr -s WT_apo_md_1_raw.tpr -extend 49000 -o WT_apo_md_1.tpr -nobackup &> md_1_convert-tpr.log
srun gmx_mpi mdrun -v -deffnm WT_apo_md_1 -nobackup &> mdrun.log
echo "0" | gmx_mpi trjconv -f WT_apo_md_1.gro -s WT_apo_md_1.tpr -o WT_apo_md_1.pdb -ur compact -pbc atom -nobackup &> md_1_trjconv.lo
echo "System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1.xtc -o WT_apo_md_1_whole.xtc -pbc whole -nobackup &>> md_1_trjconv.
echo "System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1.gro -o WT_apo_md_1_whole.gro -pbc whole -nobackup &>> md_1_trjconv.
echo "Protein System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_whole.gro -o WT_apo_md_1_cluster.gro -pbc cluster -nobackup
echo "System" | gmx_mpi trjconv -s WT_apo_md_1_cluster.gro -f WT_apo_md_1_whole.xtc -o WT_apo_md_1_nojump.xtc -pbc nojump -nobackup &>
rm WT_apo_md_1_whole.xtc WT_apo_md_1_whole.gro WT_apo_md_1_cluster.gro
echo "Protein System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_nojump.xtc -o WT_apo_md_1_imaged.xtc -pbc mol -center -ur co
rm WT_apo_md_1_nojump.xtc
echo "Backbone System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_imaged.xtc -o WT_apo_md_1_imagedFit.xtc -fit rot+trans -not
rm WT_apo_md_1_imaged.xtc
gmx_mpi check -f WT_apo_md_1_imagedFit.xtc -nobackup &> md_1_check.log
FIRST_FRAME=$(grep "Reading frame" md_1_check.log | head -1 | sed 's/time/_/' | cut -d '_' -f 2 | sed 's/ //g')
LAST_FRAME=$(grep "Last frame" md_1_check.log | sed 's/Last/_/' | cut -d '_' -f 2 | sed 's/time/_/' | cut -d '_' -f 2 |
echo "System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_imagedFit.xtc -o WT_apo_md_1_imagedFit_first.gro -e $
echo "System" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_imagedFit.xtc -o WT_apo_md_1_imagedFit_last.gro -b
echo "Protein" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_imagedFit.xtc -o WT_apo_md_1_DRY_imagedFit.xtc -n
echo "Protein" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_imagedFit_first.gro -o WT_apo_md_1_DRY_imagedFit_first.gro -nobacku
echo "Protein" | gmx_mpi trjconv -s WT_apo_md_1.tpr -f WT_apo_md_1_imagedFit_last.gro -o WT_apo_md_1_DRY_imagedFit_last.gro -nobackup
```

Usability

Interoperability

Portability

Reproducibility

Scalability

Biomolecular workflows: BioExcel

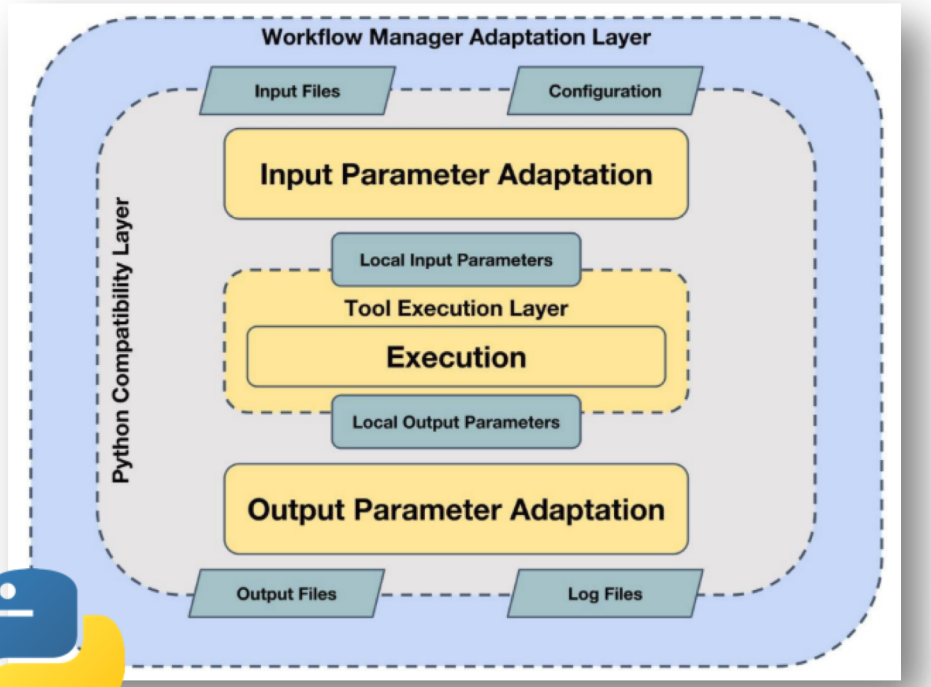
Centre of Excellence for Computational Biomolecular Research

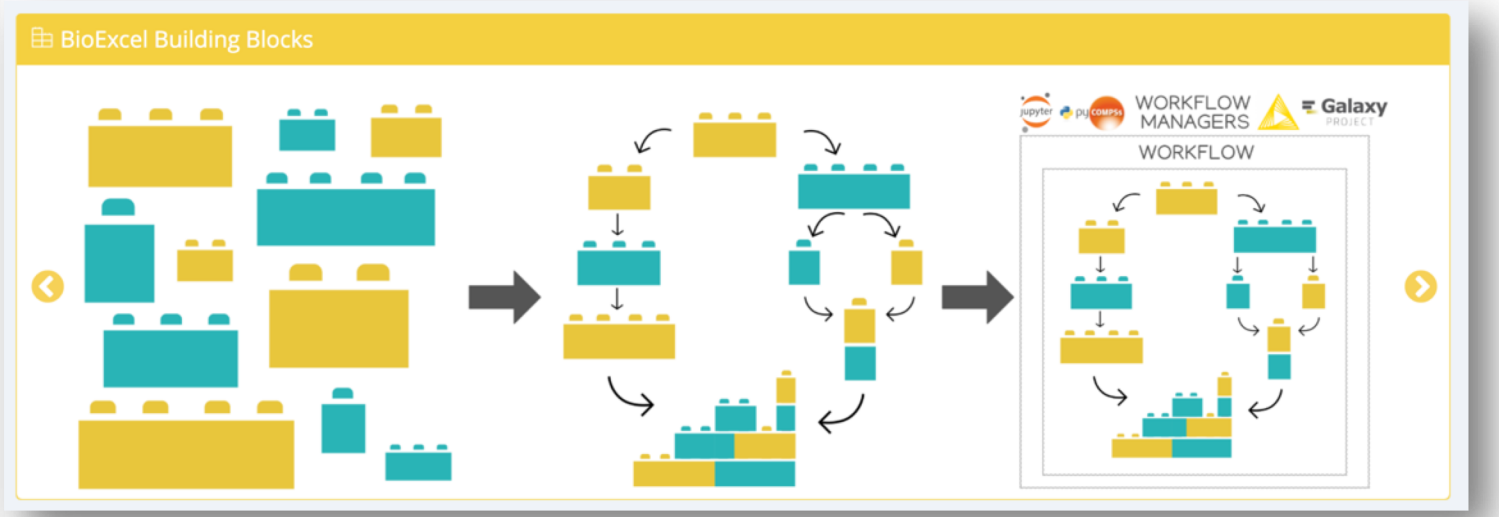
A central hub for biomolecular modelling and simulations

Best practices in research software/workflows development

https://mmb.irbbarcelona.org/biobb/

BioExcel Building Blocks: BioBB



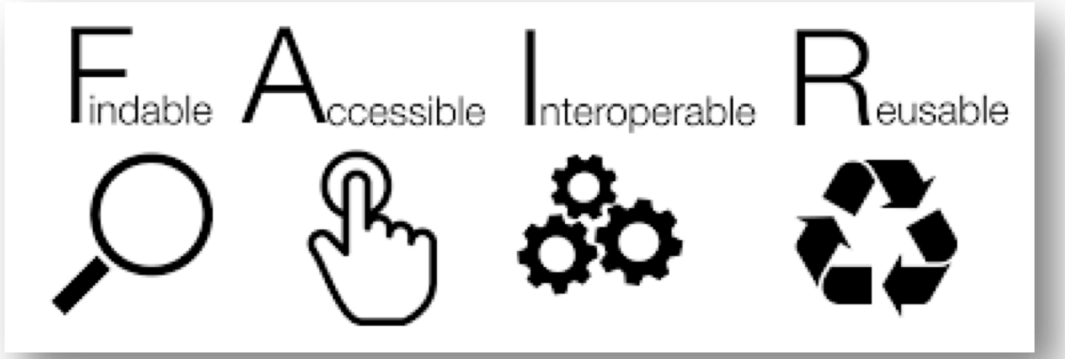


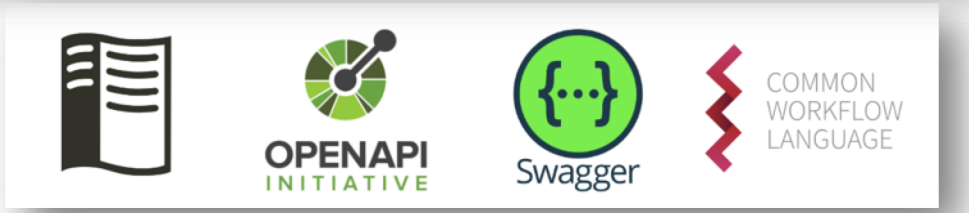
1) Building Blocks *2) Workflows* *3) Workflow Managers*

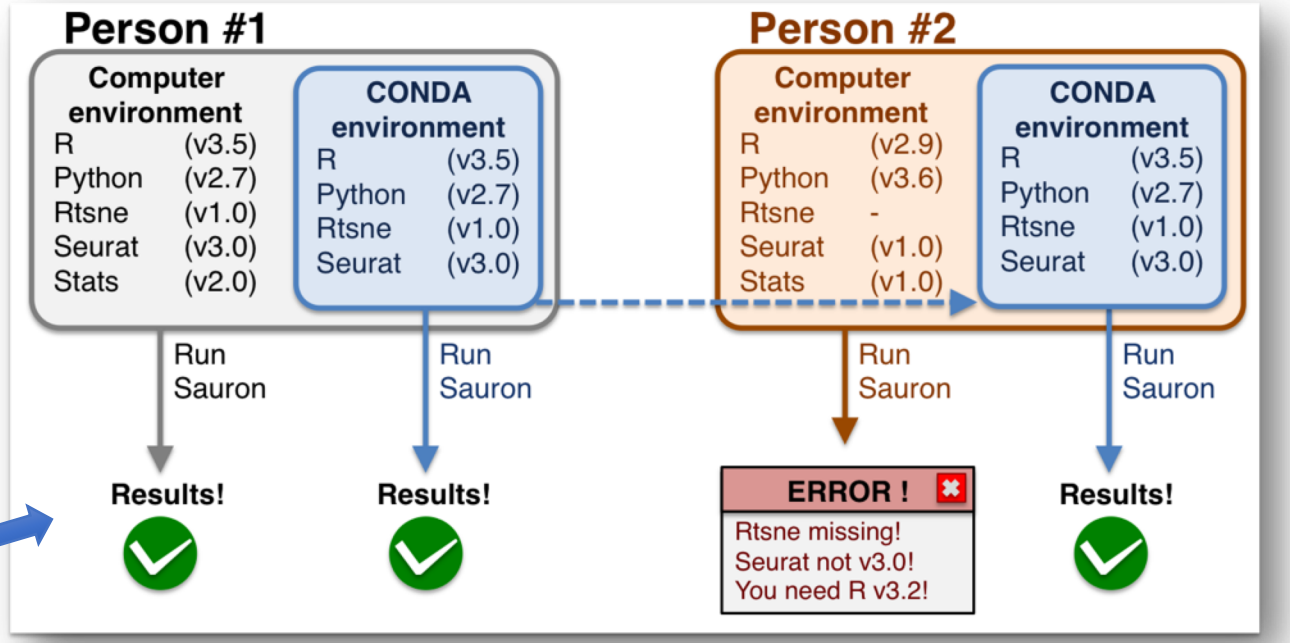
https://mmb.irbbarcelona.org/biobb/

biobb

BioBB & software best practices







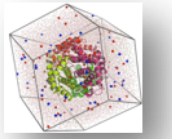
https://nbisweden.github.io/

CONDA® → ***Reproducibility***

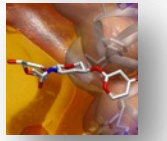
https://bioexcel.eu/webinar-using-interactive-jupyter-notebooks-and-bioconda-for-fair-and-reproducible-biomolecular-simulation-workflows-2024-05-28/

BioBB Modules (20, Release 2024.2)

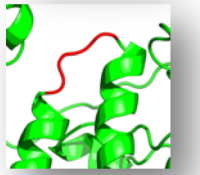
biobb_common
Common auxiliar functions



biobb_gromacs
Molecular Dynamics GROMACS

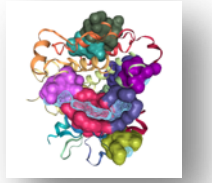


biobb_amber
Molecular Dynamics AMBER



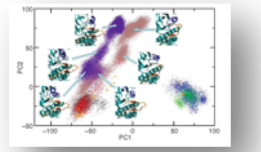
biobb_model
Molecular Modelling

biobb_ml
Machine learning

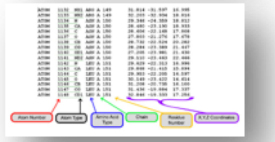


biobb_vs
Virtual Screening

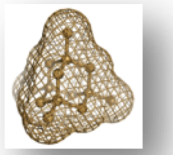
biobb_io
Biological databases



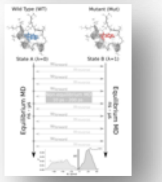
biobb_analysis
MD trajectories analysis



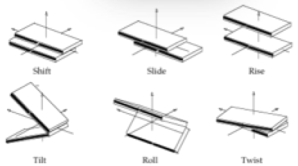
biobb_structure_utils
Modify or extract information from PDB



biobb_chemistry
Chemoinformatics functionalities



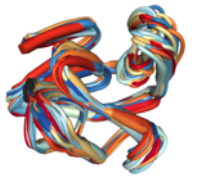
biobb_pmx
Free energy calculations



biobb_dna
Nucleic Acids MD Trajectory analyses

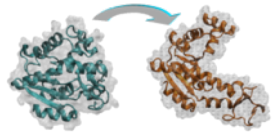
biobb_cmip
Molecular Interaction Potentials

biobb_cp2k
Quantum Mechanics



biobb_flexdyn
NMA-based conformational ensemble generation

biobb_flexserv
Coarse-Grained conformational ensemble generation and flexibility analysis



biobb_godmd
Coarse-Grained conformational transitions

BioBB Packaging / Containerization

SOURCE AND DOCS FOR BIOEXCEL BUILDING BLOCKS

Search by text: Write something

Search by keywords: Select or write keyword(s)

Package	Description	Python	ReadTheDocs	Bioconda	Docker	Singularity	Version
biobb_amber	biobb_amber is a BioBB category for AMBER MD package, allowing setup and simulation of atomistic MD simulations using AMBER MD package and its associated AMBER tools						4.0.1
biobb_analysis	Biobb_analysis is the Biobb module collection to perform analysis of molecular dynamics simulations						4.0.2

Building block	Wrapped tool	Description
GMXCluster	gmx cluster	Wrapper of the GROMACS cluster module for clustering structures from a given GROMACS compatible trajectory.
GMXRms	gmx rms	Wrapper of the GROMACS module for calculating the Root Mean Square deviation (RMSd) of a given GROMACS compatible trajectory.
GMXRgyr	gmx gyrate	Wrapper of the GROMACS gyrate module for computing the radius of gyration (Rgyr) of a molecule about the x-, y- and z-axes, as a function of time, from a given GROMACS compatible trajectory.
GMXEnergy	gmx energy	Wrapper of the GROMACS energy module for extracting energy components from a given GROMACS energy file.

conda install biobb_analysis

AmberTools19

FAST. FLEXIBLE. FREE. GROMACS

conda install biobb_vs

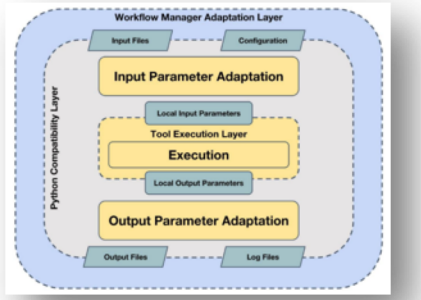
Autodock Vina

fpocket
scalable high performance pocket detection

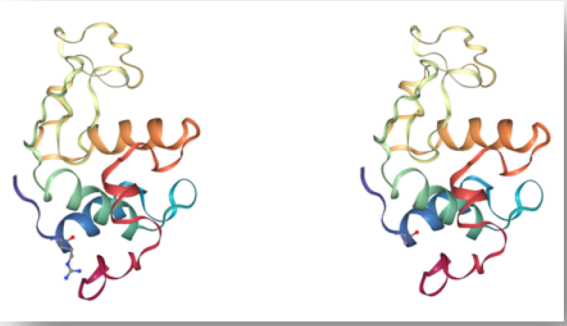
https://mmb.irbbarcelona.org/biobb/documentation/source

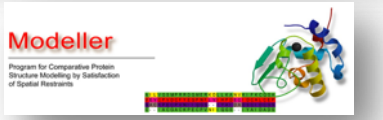
BioBB Syntax

- **Import Module**
- **Define:**
  - **inputs** & **output** paths
  - **properties** dictionary
- **Launch** building block




```python
# Import module
from biobb_model.model.mutate import mutate

# Create properties dict and inputs/outputs
input_pdb = "1AKI.pdb"
output_pdb = "1AKI_V2A.pdb"

prop = {
    'mutation_list': 'A:Val2Ala',
    'use_modeller' : True
}

# Create and launch bb
mutate(input_pdb_path=input_pdb,
       output_pdb_path=output_pdb,
       properties=prop)
```

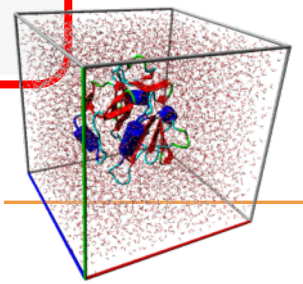




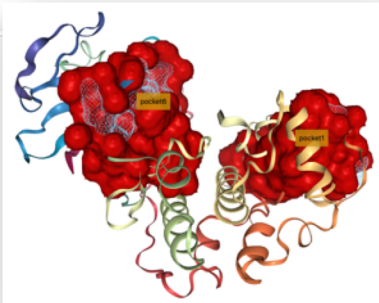

```python
# Editconf: Create solvent box
# Import module
from biobb_gromacs.gromacs.editconf import editconf
```

```python
# Create prop dict and inputs/outputs
input_pdb2gmx_gro = '1AKI_pdb2gmx.gro'
output_editconf_gro = '1AKI_editconf.gro'

prop = {
    'box_type': 'cubic',
    'distance_to_molecule': 1.0
}
```

```python
# Create and launch bb
editconf(input_gro_path=input_pdb2gmx_gro,
         output_gro_path=output_editconf_gro,
         properties=prop)
```



```python
# Import module
from biobb_vs.fpocket.fpocket_run import fpocket_run

# Define input/output and properties
pdb_protein = "pdb_protein.pdb"
fpocket_all_pockets = "fpocket_all_pockets.zip"
fpocket_summary = "fpocket_summary.json"

prop = {
    "min_radius": 3,
    "max_radius": 6,
    "num_spheres": 35
}

# Launch bb
fpocket_run(input_pdb_path=pdb_protein,
        output_pockets_zip = fpocket_all_pockets,
        output_summary=fpocket_summary,
        properties=prop)
```



fpocket

bioexcel

FAST. FLEXIBLE. FREE. GROMACS

[2]:
```
# Ligand: Download ligand structure from MMB PDB mirror REST API (https://mmb.irbbarcelona.org/api/)
# Import module
from biobb_io.api.ligand import ligand

# Create prop dict and inputs/outputs
input_structure = ligandCode + '.pdb'

prop = {
    'ligand_code' : ligandCode
}

#Create and launch bb
ligand(output_pdb_path=input_structure,
       properties=prop)
```

REST-API

[4]:
```
# Babel_add_hydrogens: add Hydrogen atoms to a small molecule
# Import module
from biobb_chemistry.babelm.babel_add_hydrogens import babel_add_hydrogens

# Create prop dict and inputs/outputs
output_babel_h = ligandCode + '.H.mol2'

prop = {
    'ph' : pH,
    'input_format' : 'pdb',
    'output_format' : 'mol2'
}

#Create and launch bb
babel_add_hydrogens(input_path=input_structure,
                    output_path=output_babel_h,
                    properties=prop)
```

Open Babel

[6]:
```
# Babel_minimize: Structure energy minimization of a small molecule after being modified adding hydrogen atoms
# Import module
from biobb_chemistry.babelm.babel_minimize import babel_minimize

# Create prop dict and inputs/outputs
output_babel_min = ligandCode + '.H.min.pdb'
prop = {
    'method' : 'sd',
    'criteria' : '1e-10',
    'force_field' : 'GAFF'
}


#Create and launch bb
babel_minimize(input_path=output_babel_h,
               output_path=output_babel_min,
               properties=prop)
```

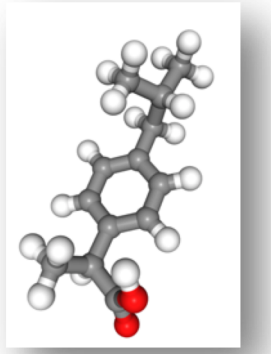
Open Babel

[9]:
```
# Acpype_params_gmx: Generation of topologies for GROMACS with ACPype
# Import module
from biobb_chemistry.acpype.acpype_params_gmx import acpype_params_gmx

# Create prop dict and inputs/outputs
output_acpype_gro = ligandCode + 'params.gro'
output_acpype_itp = ligandCode + 'params.itp'
output_acpype_top = ligandCode + 'params.top'
output_acpype = ligandCode + 'params'
prop = {
    'basename' : output_acpype,
    'charge' : mol_charge
}

#Create and launch bb
acpype_params_gmx(input_path=output_babel_min,
                  output_path_gro=output_acpype_gro,
                  output_path_itp=output_acpype_itp,
                  output_path_top=output_acpype_top,
                  properties=prop)
```

ACPYPE

Ligand parameterization workflow:

1. Download Ligand Structure
2. Add hydrogen atoms
3. Energetically minimize H atoms
4. Generate parameters



jupyter

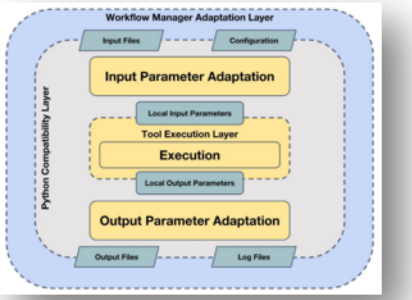
```
name: biobb_wf_ligand_parameterization
channels:
  - conda-forge
  - bioconda
dependencies:
  - biobb_io==4.2.0
  - biobb_chemistry==4.2.0
  - jupyter
  - nglview
```

CONDA®

bioexcel

BioBB Documentation

- **Import Module**
- **Define:**
  - **inputs** & **output** paths
  - **properties** dictionary
- **Launch** building block



biobb

latest

Search docs

Introduction & installation

API Documentation

acpype package

Submodules

acpype.acpype_params_ac module

acpype.acpype_params_cns module

acpype.acpype_params_gmx module

acpype.acpype_params_gmx_opls module

babelm package

ambertools package

Command Line Documentation

Changelog

acpype.acpype_params_gmx module

Module containing the Acpype class and the command line interface.

class **acpype.acpype_params_gmx.AcpypeParamsGMX**(*input_path, output_path_gro, output_path_itp, output_path_top, properties=None, **kwargs*) [source]

Bases: `object`

Small molecule parameterization for GROMACS MD package. Wrapper for the Acpype module. Generation of topologies for GROMACS. Acpype is a tool based in Python to use Antechamber to generate topologies for chemical compounds and to interface with others python applications like CCPN or ARIA. Visit the official page.

Parameters:

- **input_path** (*str*) –
  Path to the input file. File type: input. Sample file. Accepted formats: pdb, mdl, mol2.
- **output_path_gro** (*str*) –
  Path to the GRO output file. File type: output. Sample file. Accepted formats: gro.
- **output_path_itp** (*str*) –
  Path to the ITP output file. File type: output. Sample file. Accepted formats: itp.
- **output_path_top** (*str*) –
  Path to the TOP output file. File type: output. Sample file. Accepted formats: top.
- **properties** (*dic*) –
  - **basename** (*str*) - ("BBB") A basename for the project (folder and output files).
  - **charge** (*int*) - (0) Net molecular charge, for gas default is 0.
  - **acpype_path** (*str*) - ("acpype") Path to the acpype executable binary.
  - **remove_tmp** (*bool*) - (True) [WF property] Remove temporal files.
  - **restart** (*bool*) - (False) [WF property] Do not execute if output files exist.
  - **container_path** (*str*) - (None) Container path definition.
  - **container_image** (*str*) - ('mmbirb/acpype:latest') Container image definition.
  - **container_volume_path** (*str*) - ('/tmp') Container volume path definition.
  - **container_working_dir** (*str*) - (None) Container working directory definition.
  - **container_user_id** (*str*) - (None) Container user_id definition.
  - **container_shell_path** (*str*) - ('/bin/bash') Path to default shell inside the container.

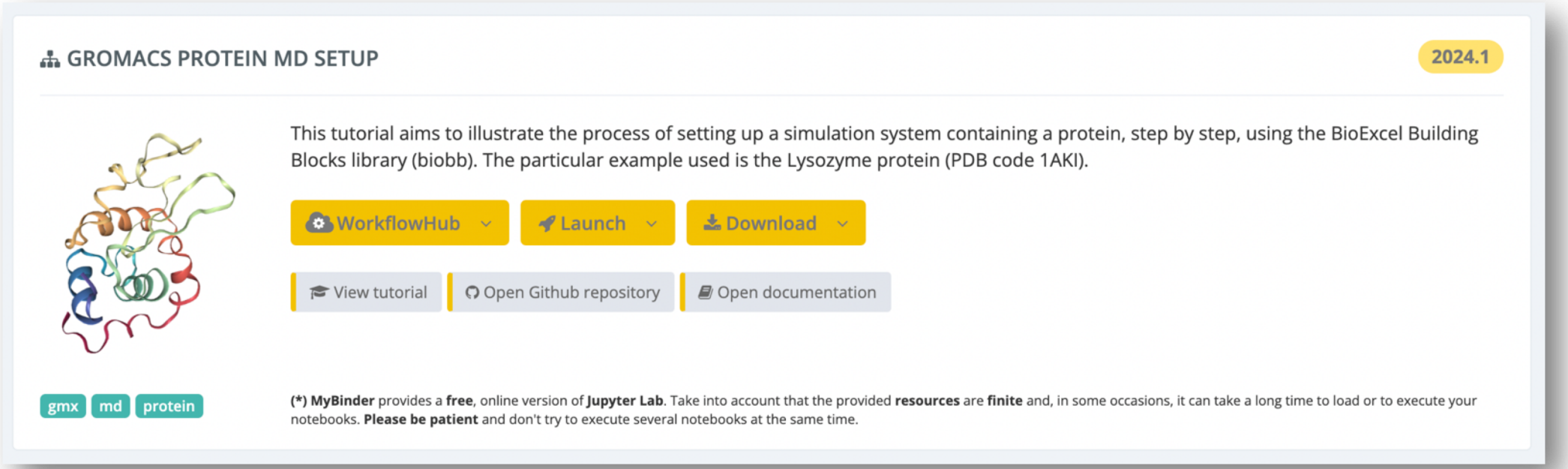
Examples:

This is a use example of how to use the building block from Python:

```
from biobb_chemistry.acpype.acpype_params_gmx import acpype_params_gmx
prop = {
    'basename': 'BBB',
    'charge': 0
}
acpype_params_gmx(input_path='/path/to/myStructure.mol2',
                  output_path_gro='/path/to/newGRO.gro',
                  output_path_itp='/path/to/newITP.itp',
                  output_path_top='/path/to/newTOP.top',
                  properties=prop)
```

bioexcel

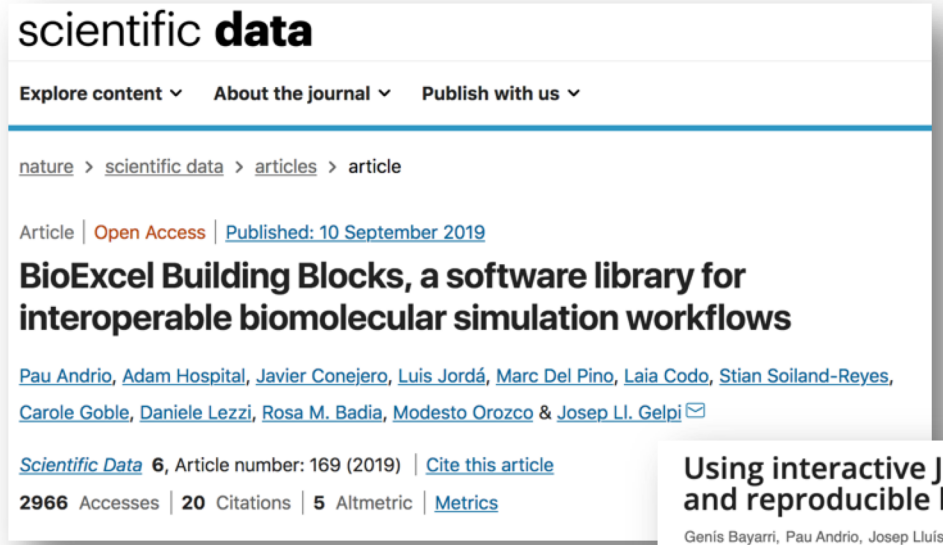
BioBB Demonstration Workflows



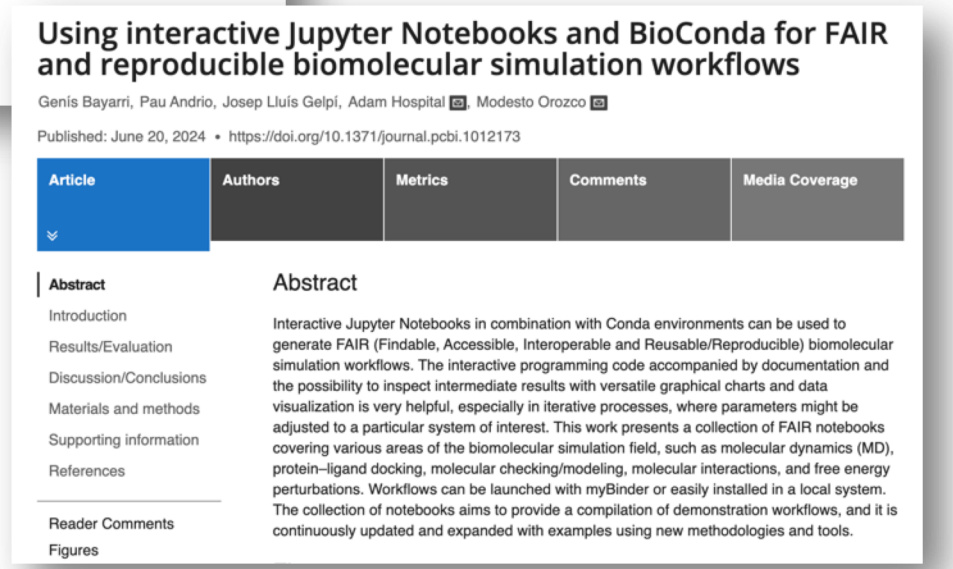
https://mmb.irbbarcelona.org/biobb/workflows

- ***MD setup (Protein / DNA) (AMBER / GROMACS)***
- ***Ligand parameterization***
- ***Protein-Ligand Docking***
- ***Free energy calculations***
- ***DNA helical parameters***
- ***Conformational Ensemble generation***

More about BioBB & BioExcel



https://doi.org/10.1038/s41597-019-0177-4



https://doi.org/10.1371/journal.pcbi.1012173

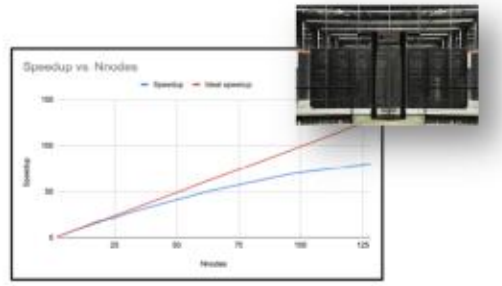
- **BioExcel Website:**
https://bioexcel.eu/

- **BioBB Website:**
https://mmb.irbbarcelona.org/biobb/

- **BioBB Webinars:**
https://bioexcel.eu/webinar-computational-biomolecular-simulation-workflows-with-bioexcel-building-blocks-2020-09-10/
https://bioexcel.eu/webinar-biobb-wfs-and-biobb-api-integrated-web-based-platform-and-programmatic-interface-for-biomolecular-simulations-workflows-using-the-bioexcel-building-blocks-library-2023-05-23/
https://bioexcel.eu/webinar-using-interactive-jupyter-notebooks-and-bioconda-for-fair-and-reproducible-biomolecular-simulation-workflows-2024-05-28/

Questions?

BioBB Library Versatility



HPC Workflows

COMMON WORKFLOW LANGUAGE

CWL Workflows

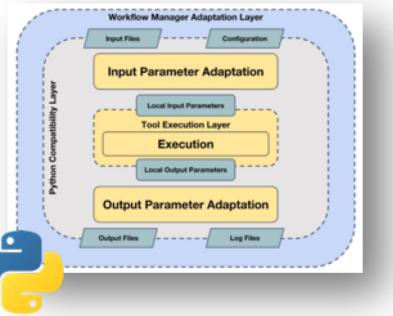
Horus

Galaxy PROJECT

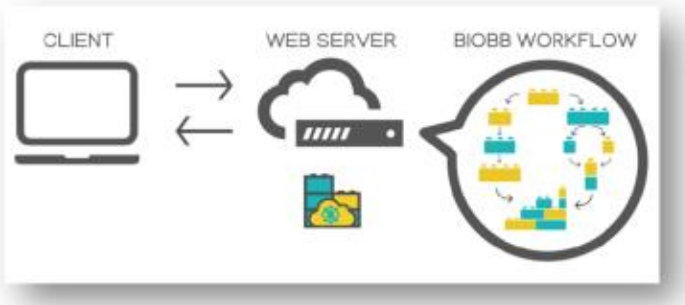
Open for Innovation KNIME

Drag & Drop GUIs

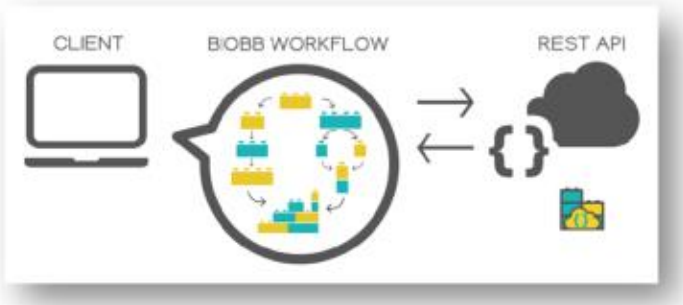
biobb



Web-based GUI



REST API



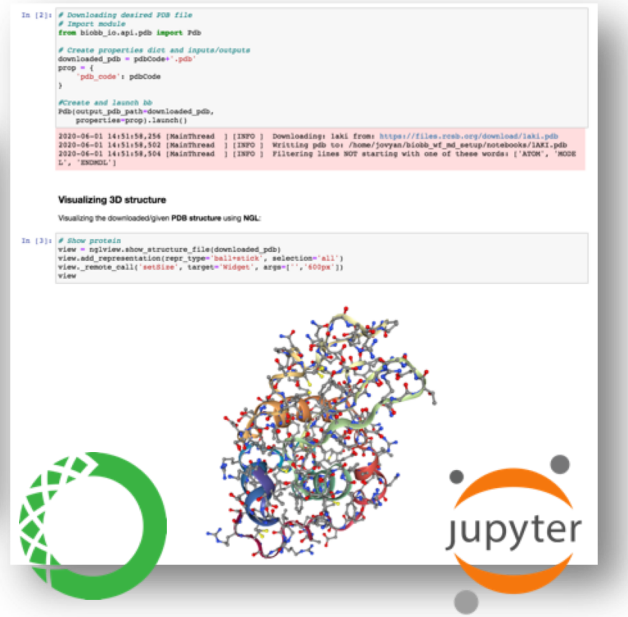
https://mmb.irbbarcelona.org/biobb-wfs

https://mmb.irbbarcelona.org/biobb-api

bioexcel

BioBB Library Versatility - Examples

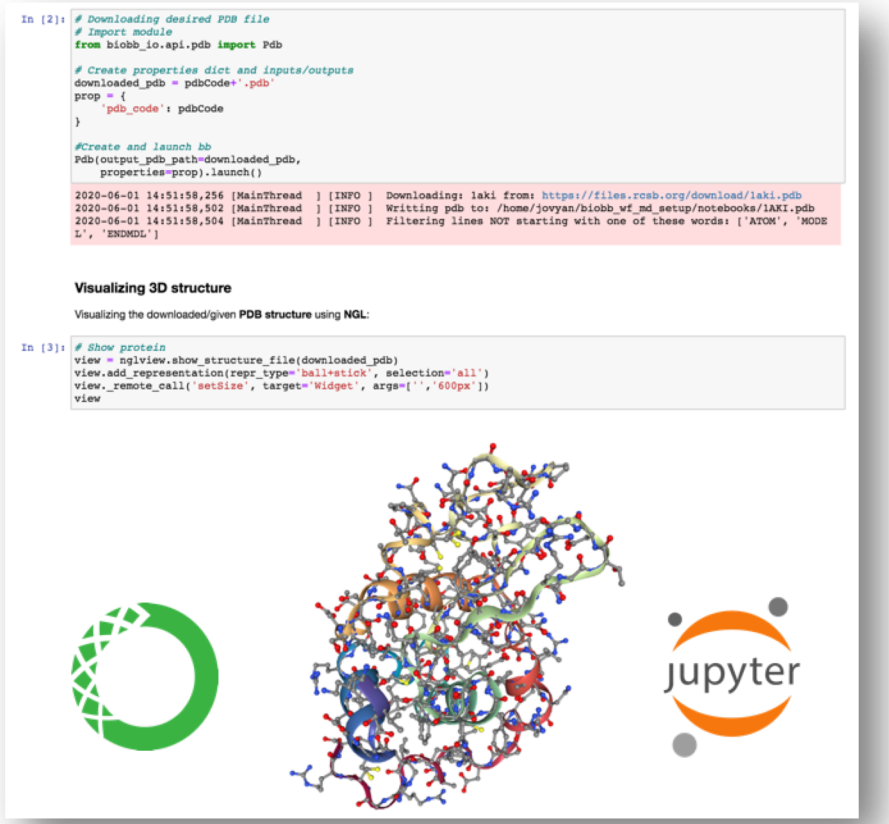
1) Demonstration Workflows (Jupyter Notebooks)



2) Pre-exascale Workflows (Python + PyCOMPSs)

Biomolecular workflows using BioBB

Demonstration Workflows (Jupyter Notebooks)



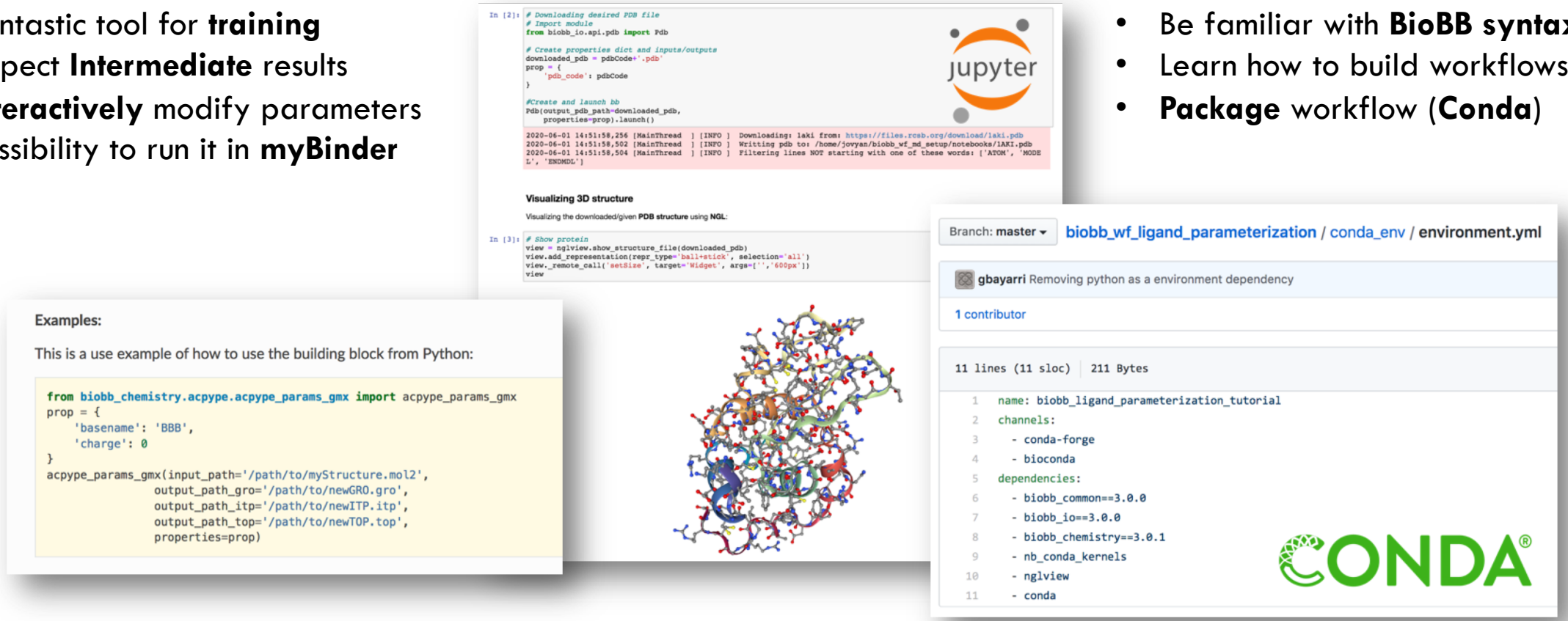
Jupyter Notebooks & BioBBs

In general:

- Fantastic tool for **training**
- Inspect **Intermediate** results
- **Interactively** modify parameters
- Possibility to run it in **myBinder**

In particular (BioBBs):

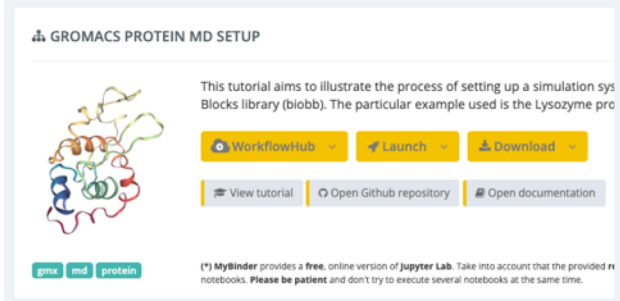
- Be familiar with **BioBB syntax**
- Learn how to build workflows (**tutorials**)
- **Package** workflow (**Conda**)

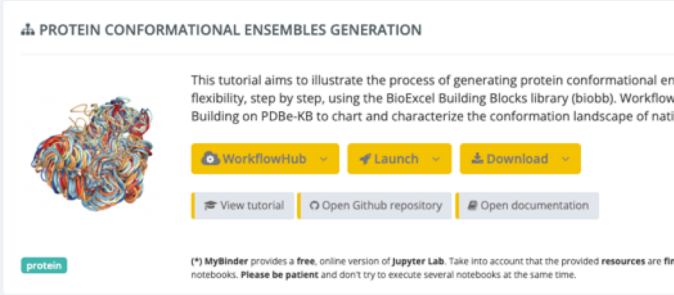


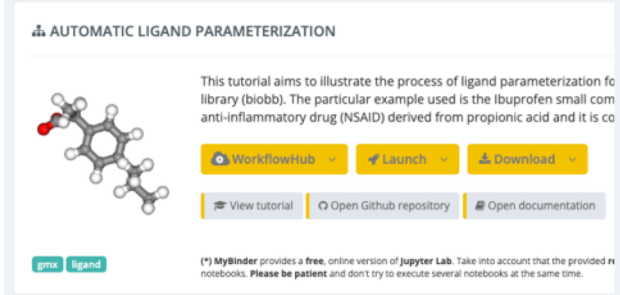
BioBB Demonstration Workflows

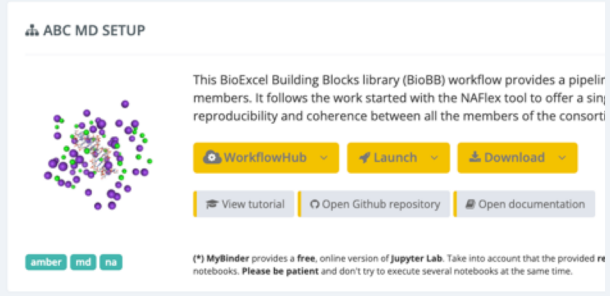
https://mmb.irbbarcelona.org/biobb/workflows

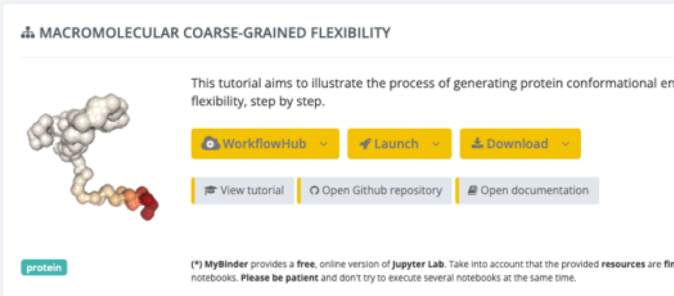
- *Showing the power of the BioBB library*
- *Transversal, generic*
- *Educational purposes (not for production usage)*

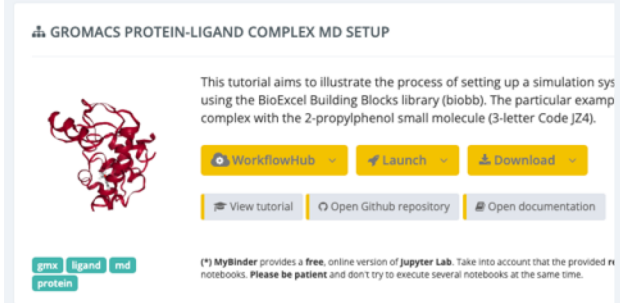


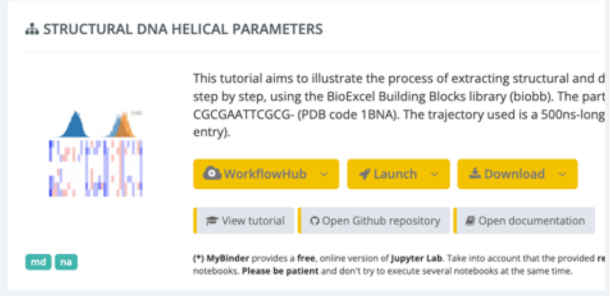


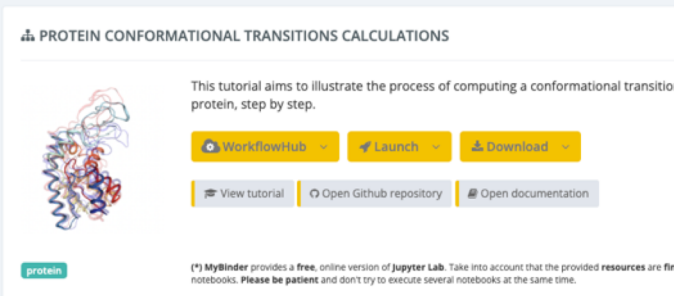


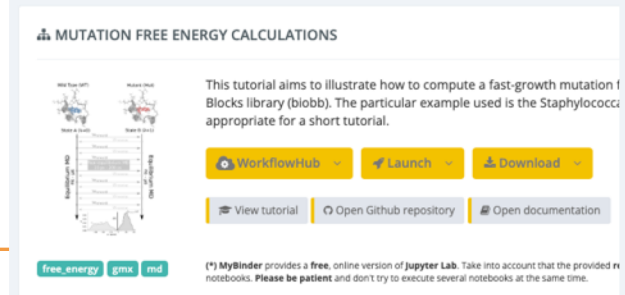


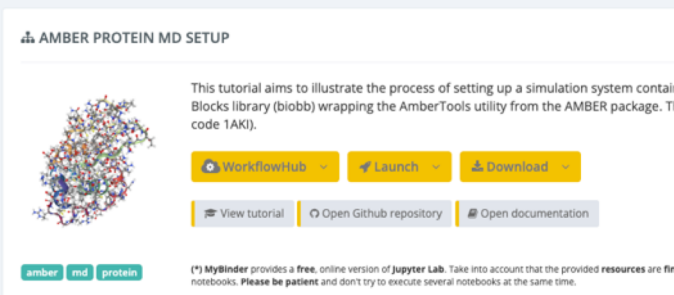












BioBB Workflows: Tutorials

Uniform header for all **BioBB tutorials**:

1) **Title and description**
2) **BioBB modules used**
3) **Auxiliary libraries used**
4) **Command lines required to install and launch**
5) **Pipeline steps**

Process

Fetching PDB structure

Downloading **PDB structure** with the **protein molecule** from the RCSB PDB database.
Alternatively, a **PDB file** can be used as starting structure.

Building Blocks used:

- Pdb from **biobb_io.api.pdb**

Documentation

```
In [8]: # Downloading desired PDB file
        # Import module
        from biobb_io.api.pdb import pdb

        # Create properties dict and inputs/outputs
        downloaded_pdb = pdbCode+'.pdb'
        prop = {
            'pdb_code': pdbCode

        }

        #Create and launch bb
        pdb(output_pdb_path=downloaded_pdb,
            properties=prop)
```

Execution

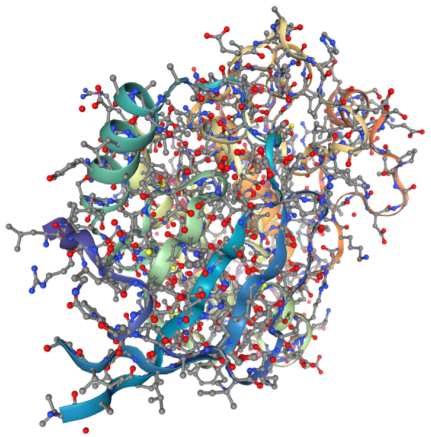
```
2023-08-27 10:36:35,865 [MainThread  ] [INFO ]  Executing biobb_io.api.pdb Version: 4.0.0
2023-08-27 10:36:35,865 [MainThread  ] [INFO ]  Downloading 1aki from: https://www.ebi.ac.uk/pdbe/entry-files/download/pdb1aki.ent
2023-08-27 10:36:36,106 [MainThread  ] [INFO ]  Writting pdb to: 1aki.pdb
2023-08-27 10:36:36,108 [MainThread  ] [INFO ]  Filtering lines NOT starting with one of these words: ['ATOM', 'MODEL', 'ENDMDL']
```

Visualizing 3D structure

Visualizing the downloaded/given **PDB structure** using **NGL**:

```
In [83]: # Show protein
         view = nglview.show_structure_file(downloaded_pdb)
         view.add_representation(repr_type='ball+stick', selection='all')
         view._remote_call('setSize', target='Widget', args=['','600px'])
         view
```

Inspection



1. Import BioBB module

```
# Downloading desired PDB file
# Import module
from biobb_io.api.pdb import pdb
```

2. Define inputs/outputs and properties

```
# Create properties dict and inputs/outputs
downloaded_pdb = pdbCode+'.pdb'
prop = {
    'pdb_code': pdbCode
}
```

3. Launch the building block execution

```
#Create and launch bb
pdb(output_pdb_path=downloaded_pdb,
    properties=prop)
```

Documentation

Execution

Create protein system topology

Building GROMACS topology corresponding to the protein structure.
Force field used in this tutorial is amber99sb-ildn: AMBER **parm99** force field with **corrections on backbone** (sb) and **side-chain torsion potentials** (ildn).
Water molecules type used in this tutorial is spc/e.
Adding **hydrogen atoms** if missing. Automatically identifying **disulfide bridges**.

Generating two output files:

- **GROMACS structure** (gro file)
- **GROMACS topology** ZIP compressed file containing:
  - *GROMACS topology top file* (top file)
  - *GROMACS position restraint file/s* (itp file/s)

Building Blocks used:

- Pdb2gmx from **biobb_gromacs.gromacs.pdb2gmx**

In [36]:
```
# Create system topology
# Import module
from biobb_gromacs.gromacs.pdb2gmx import pdb2gmx

# Create inputs/outputs
output_pdb2gmx_gro = pdbCode+'_pdb2gmx.gro'
output_pdb2gmx_top_zip = pdbCode+'_pdb2gmx_top.zip'

prop = {
    'water_type' : 'spce',
    'force_field' :  'amber99sb-ildn'
}

# Create and launch bb
pdb2gmx(input_pdb_path=fixed_pdb,
        output_gro_path=output_pdb2gmx_gro,
        output_top_zip_path=output_pdb2gmx_top_zip,
        properties=prop
       )
```

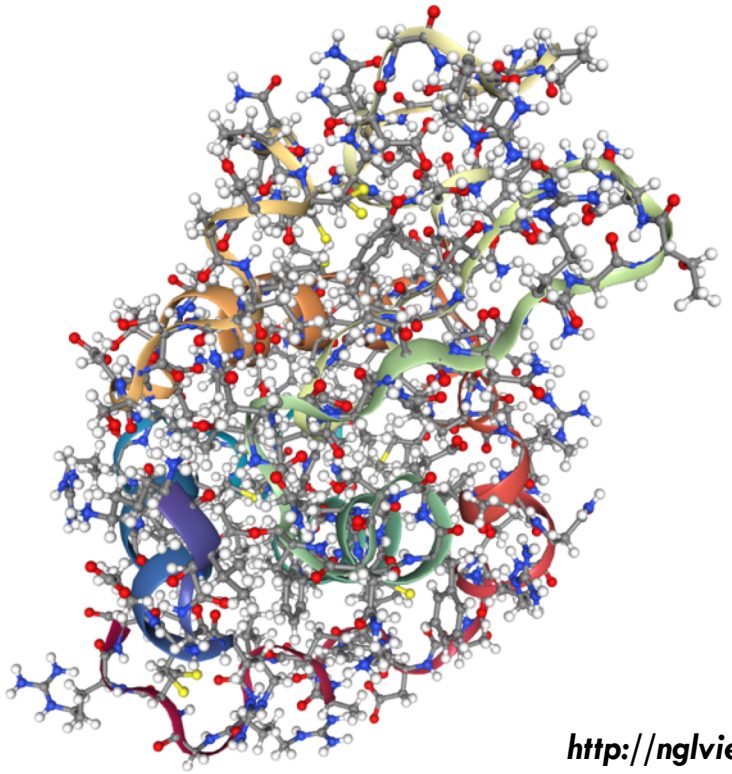
```
2023-09-04 15:33:40,575 [MainThread  ] [INFO ]  Executing biobb_gromacs.gromacs.pdb2gmx Version: 4.0.0
2023-09-04 15:33:40,579 [MainThread  ] [INFO ]  Copy: 1AKI_fixed.pdb to /home/jovyan/biobb_wf_md_setup/notebooks/c83e
58a0-e0b1-4554-ad51-351986133922
2023-09-04 15:33:40,580 [MainThread  ] [INFO ]  GROMACS Pdb2gmx 20222 version detected
2023-09-04 15:33:40,581 [MainThread  ] [INFO ]  gmx -nobackup -nocopyright pdb2gmx -f /home/jovyan/biobb_wf_md_setup/
notebooks/c83e58a0-e0b1-4554-ad51-351986133922/1AKI_fixed.pdb -o /home/jovyan/biobb_wf_md_setup/notebooks/c83e58a0-e0
b1-4554-ad51-351986133922/1AKI_pdb2gmx.gro -p p2g.top -water spce -ff amber99sb-ildn -i posre.itp

2023-09-04 15:33:40,796 [MainThread  ] [INFO ]  Exit code 0
```

Inspection

Visualizing 3D structure

Visualizing the generated **GRO structure** using **NGL**. Note that **hydrogen atoms** were added to the structure by the **pdb2gmx GROMACS tool** when generating the **topology**.

In [12]:
```
# Show protein
struct_file = nglview.FileStructure(output_pdb2gmx_gro)
view = nglview.show_file(struct_file)
view.add_representation(repr_type='ball+stick', selection='all')
view._remote_call('setSize', target='Widget', args=['','600px'])
view
```

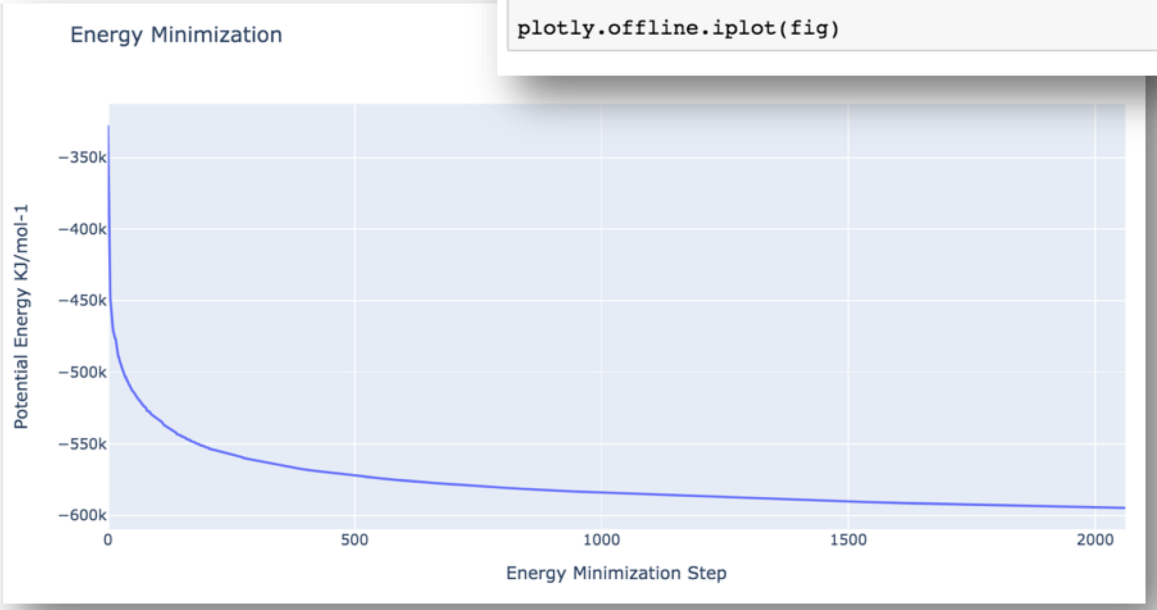


jupyter

http://nglviewer.org/nglview/latest/

Inspection


```
import plotly
import plotly.graph_objs as go

#Read data from file and filter energy values higher than 1000 Kj/mol^-1
with open(output_min_ene_xvg,'r') as energy_file:
    x,y = map(
        list,
        zip(*[
            (float(line.split()[0]),float(line.split()[1]))
            for line in energy_file
            if not line.startswith(("#","@"))
            if float(line.split()[1]) < 1000
        ])
    )

plotly.offline.init_notebook_mode(connected=True)

fig = {
    "data": [go.Scatter(x=x, y=y)],
    "layout": go.Layout(title="Energy Minimization",
                        xaxis=dict(title = "Energy Minimization Step"),
                        yaxis=dict(title = "Potential Energy KJ/mol-1")
                       )
}

plotly.offline.iplot(fig)
```



https://plotly.com/

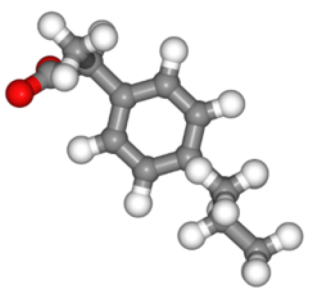
Branch: master ▾ **biobb_wf_ligand_parameterization** / conda_env / **environment.yml**

gbayarri Removing python as a environment dependency

1 contributor

11 lines (11 sloc) | 211 Bytes

```
1  name: biobb_ligand_parameterization_tutorial
2  channels:
3    - conda-forge
4    - bioconda
5  dependencies:
6    - biobb_common==3.0.0
7    - biobb_io==3.0.0
8    - biobb_chemistry==3.0.1
9    - nb_conda_kernels
10   - nglview
11   - conda
```



Automatic Ligand parameterization

CONDA®

GitHub

binder

Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

Build and launch a repository

GitHub repository name or URL

GitHub ▾ | GitHub repository name or URL

Git branch, tag, or commit | Path to a notebook file (optional) | File ▾ | launch

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

Copy the text below, then paste into your README to show a binder badge: launch binder

Conda Installation and Launch

```
git clone https://github.com/bioexcel/biobb_wf_md_setup.git
cd biobb_wf_md_setup
conda env create -f conda_env/environment.yml
conda activate biobb_GMX_MDsetup_tutorial
jupyter-notebook biobb_wf_md_setup/notebooks/biobb_MDsetup_tutorial.ipynb
```

biobb_MDsetup_tutorial.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto Copiar en Drive

Google colab

Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Based on the official GROMACS tutorial: http://www.mdtutorials.com/gmx/lysozyme/index.html

This tutorial aims to illustrate the process of **setting up a simulation system** containing a **protein**, step by step, using the **BioExcel Building Blocks library (biobb)**. The particular example used is the **Lysozyme** protein (PDB code 1AKI, https://doi.org/10.2210/pdb1AKI/pdb).

Settings

Biobb modules used

- biobb_io: Tools to fetch biomolecular data from public databases.
- biobb_model: Tools to model macromolecular structures.
- biobb_gromacs: Tools to setup and run Molecular Dynamics simulations.
- biobb_analysis: Tools to analyse Molecular Dynamics trajectories.

Auxiliary libraries used

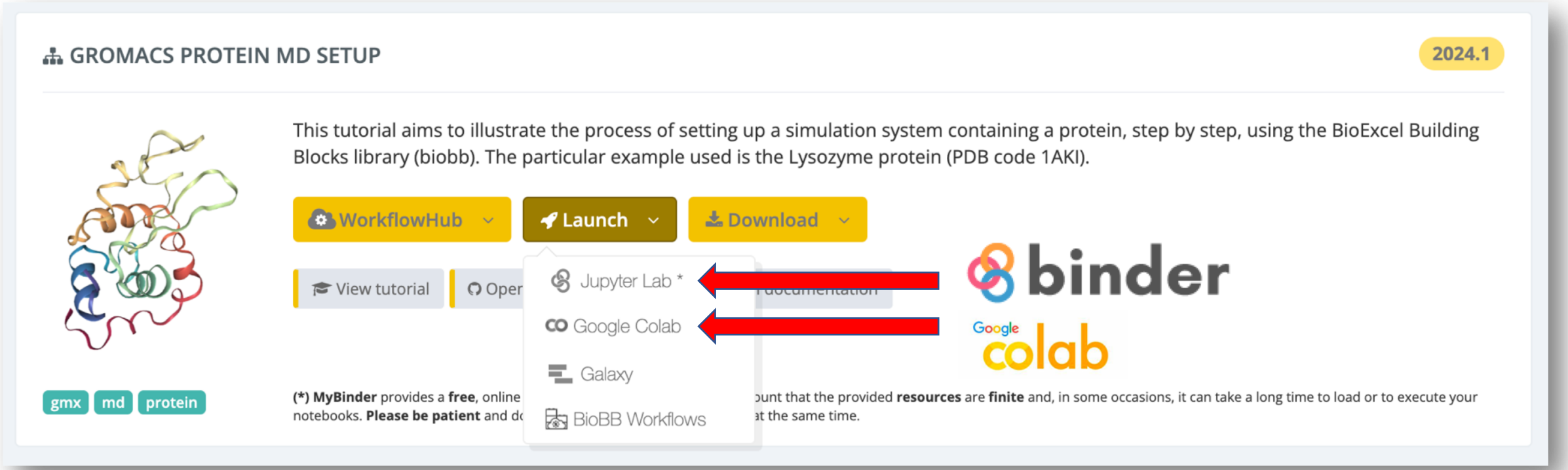
- jupyter: Free software, open standards, and web services for interactive computing across all programming languages.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in notebooks.
- plotly: Python interactive graphing library integrated in Jupyter notebooks.
- simpletraj: Lightweight coordinate-only trajectory reader based on code from GROMACS, MDAnalysis and VMD.

Conda Installation and Launch

```
git clone https://github.com/bioexcel/biobb_wf_md_setup.git
cd biobb_wf_md_setup
conda env create -f conda_env/environment.yml
```

bioexcel

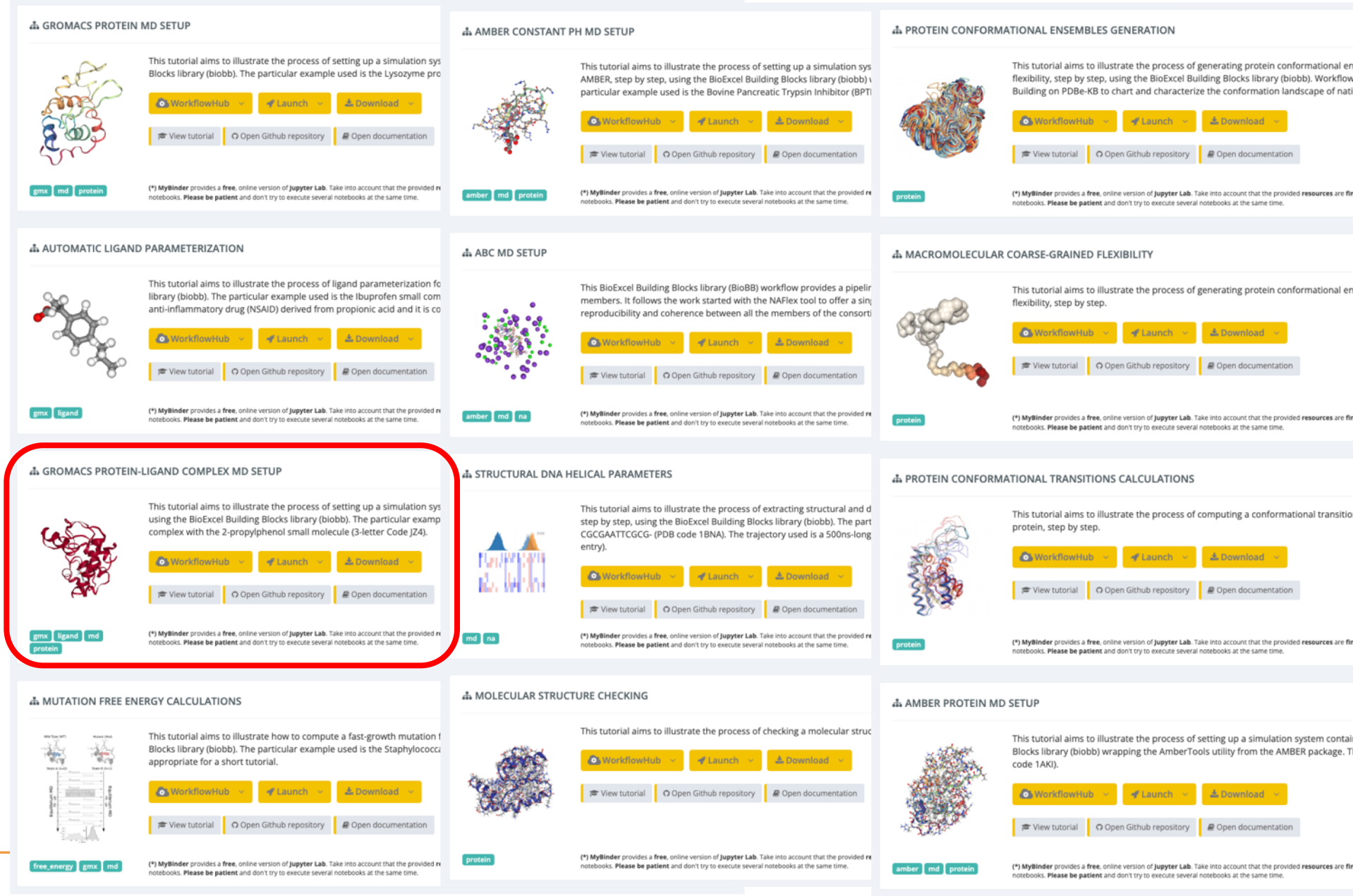
https://mmb.irbbarcelona.org/biobb/workflows

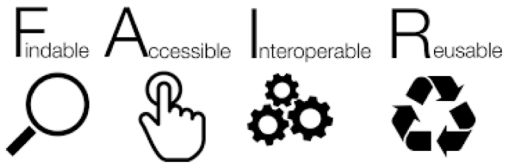


BioBB Demonstration Workflows

https://mmb.irbbarcelona.org/biobb/workflows

- *Showing the power of the BioBB library*
- *Transversal, generic*
- *Educational purposes (not for production usage)*





PLOS COMPUTATIONAL BIOLOGY

EDUCATION

Using interactive Jupyter Notebooks and BioConda for FAIR and reproducible biomolecular simulation workflows

Genís Bayarri[1], Pau Andrio[2], Josep Lluís Gelpí[2,3], Adam Hospital[1]*, Modesto Orozco[1,3]*

1 Institute for Research in Biomedicine (IRB Barcelona), the Barcelona Institute of Science and Technology, Barcelona, Spain, **2** Barcelona Supercomputing Center (BSC), Barcelona, Spain, **3** Department of Biochemistry and Biomedicine, University of Barcelona, Barcelona, Spain

* adam.hospital@irbbarcelona.org (AH); modesto.orozco@irbbarcelona.org (MO)

OPEN ACCESS

Citation: Bayarri G, Andrio P, Gelpí JL, Hospital A, Orozco M (2024) Using interactive Jupyter Notebooks and BioConda for FAIR and reproducible biomolecular simulation workflows. PLoS Comput Biol 20(6): e1012173. https://doi.org/10.1371/journal.pcbi.1012173

Editor: B. F. Francis Ouellette, bioinformatics.ca, CANADA

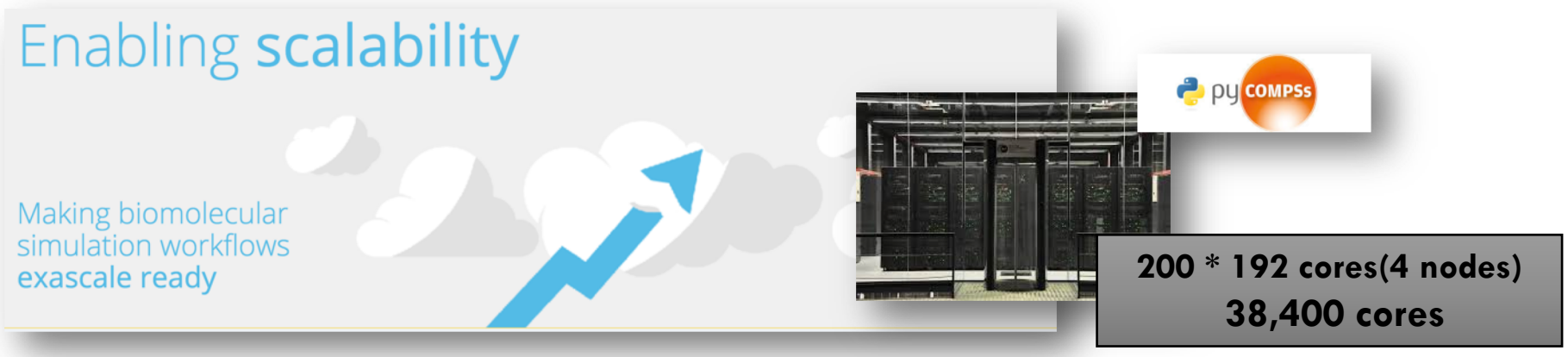
Published: June 20, 2024

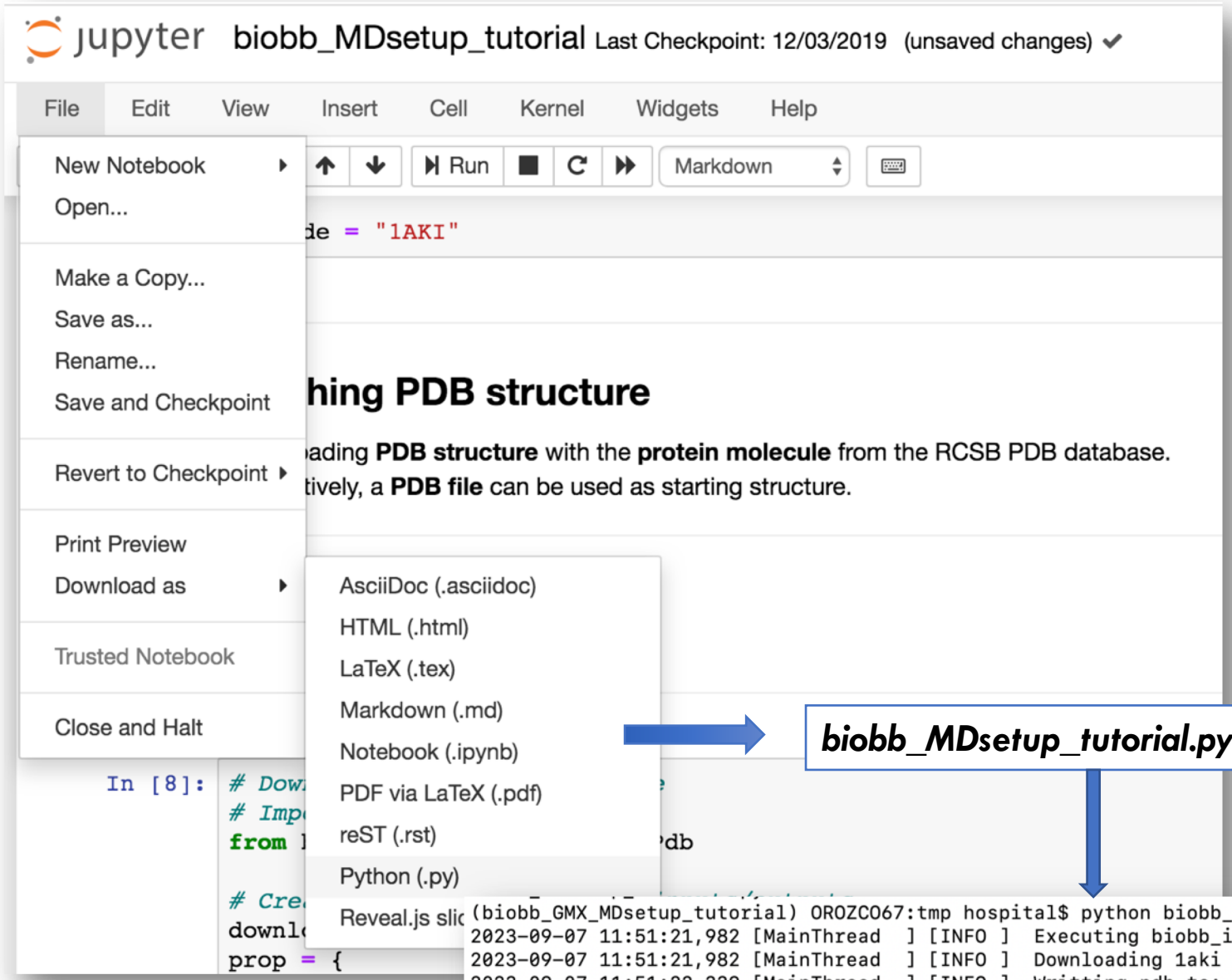
Abstract

Interactive Jupyter Notebooks in combination with Conda environments can be used to generate FAIR (Findable, Accessible, Interoperable and Reusable/Reproducible) biomolecular simulation workflows. The interactive programming code accompanied by documentation and the possibility to inspect intermediate results with versatile graphical charts and data visualization is very helpful, especially in iterative processes, where parameters might be adjusted to a particular system of interest. This work presents a collection of FAIR notebooks covering various areas of the biomolecular simulation field, such as molecular dynamics (MD), protein–ligand docking, molecular checking/modeling, molecular interactions, and free energy perturbations. Workflows can be launched with myBinder or easily installed in a local system. The collection of notebooks aims to provide a compilation of demonstration workflows, and it is continuously updated and expanded with examples using new methodologies and tools.

Biomolecular workflows using BioBB

HPC Workflows





biobb_MDsetup_tutorial.py

```
(biobb_GMX_MDsetup_tutorial) OROZCO67:tmp hospital$ python biobb_MDsetup_tutorial.py
2023-09-07 11:51:21,982 [MainThread  ] [INFO ]  Executing biobb_io.api.pdb Version: 3.8.0
2023-09-07 11:51:21,982 [MainThread  ] [INFO ]  Downloading 1aki from: https://www.ebi.ac.uk/pdbe/entry-files/download/pdb1aki.ent
2023-09-07 11:51:22,339 [MainThread  ] [INFO ]  Writting pdb to: 1AKI.pdb
2023-09-07 11:51:22,339 [MainThread  ] [INFO ]  Filtering lines NOT starting with one of these words: ['ATOM', 'MODEL', 'ENDMDL']
2023-09-07 11:51:22,360 [MainThread  ] [INFO ]  Executing biobb_model.model.fix_side_chain Version: 3.8.0
2023-09-07 11:51:22,360 [MainThread  ] [INFO ]  check_structure -i 1AKI.pdb -o 1AKI_fixed.pdb --force_save fixside --fix ALL

2023-09-07 11:51:22,677 [MainThread  ] [INFO ]  Exit code 0
```

BioBB Workflows Command-Line Interface (CLI)

Disadvantages:

- Graphical cells not showing.
- Losing interactivity

Advantages:

- Gaining High Throughput (automation, repetition)

Problem:

- Modify parameters for a certain step
  → modify the Python script

bioexcel

Workflow script

- Building blocks
- Python code
- Loops / conditionals
- Global log
- Output folders hierarchy

workflow.py

```python
global_log.info("step1_pdb: Download the initial Structure")
Pdb(**global_paths["step1_pdb"], properties=global_prop["step1_pdb"])

global_log.info("step2_fixsidechain: Modeling the missing heavy atoms in the structure side chains")
FixSideChain(**global_paths["step2_fixsidechain"], properties=global_prop["step2_fixsidechain"])

global_log.info("step3_pdb2gmx: Generate the topology")
Pdb2gmx(**global_paths["step3_pdb2gmx"], properties=global_prop["step3_pdb2gmx"])

global_log.info("step4_editconf: Create the solvent box")
Editconf(**global_paths["step4_editconf"], properties=global_prop["step4_editconf"])

global_log.info("step5_solvate: Fill the solvent box with water molecules")
Solvate(**global_paths["step5_solvate"], properties=global_prop["step5_solvate"])

global_log.info("step6_grompp_genion: Preprocess ion generation")
Grompp(**global_paths["step6_grompp_genion"], properties=global_prop["step6_grompp_genion"])

global_log.info("step7_genion: Ion generation")
Genion(**global_paths["step7_genion"], properties=global_prop["step7_genion"])

global_log.info("step8_grompp_min: Preprocess energy minimization")
Grompp(**global_paths["step8_grompp_min"], properties=global_prop["step8_grompp_min"])

global_log.info("step9_mdrun_min: Execute energy minimization")
Mdrun(**global_paths["step9_mdrun_min"], properties=global_prop["step9_mdrun_min"])

global_log.info("step10_energy_min: Compute potential energy during minimization")
GMXEnergy(**global_paths["step10_energy_min"], properties=global_prop["step10_energy_min"])
```

Workflow parameters

- Steps Inputs / Outputs
- Steps Dependencies
- Steps Properties
- Workflow inputs & parameters

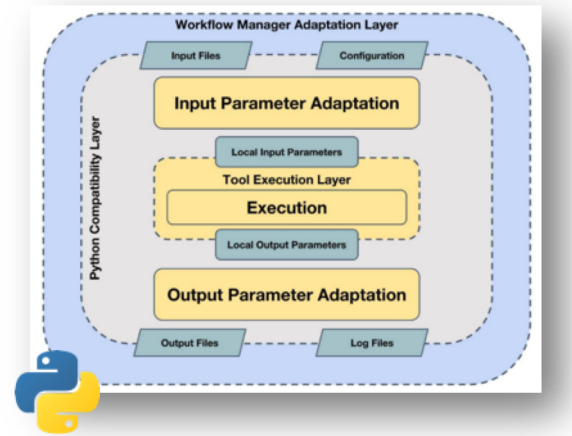
workflow.yaml

```yaml
# Example of a YAML configuration file for a BioExcel building blocks workflow

working_dir_path: md_tutorial      # Folder to write i/o files of the workflow steps
can_write_console_log: False       # Verbose writing of log information
restart: False                      # Skip steps already performed
remove_tmp: True

step1_pdb:
  paths:
    output_pdb_path: structure.pdb
  properties:
    pdb_code: 1aki

step2_fixsidechain:
  paths:
    input_pdb_path: dependency/step1_pdb/output_pdb_path
    output_pdb_path: fixsidechain.pdb

step3_pdb2gmx:
  paths:
    input_pdb_path: dependency/step2_fixsidechain/output_pdb_path
    output_gro_path: pdb2gmx.gro
    output_top_zip_path: pdb2gmx_top.zip

step4_editconf:
  paths:
    input_gro_path: dependency/step3_pdb2gmx/output_gro_path
    output_gro_path: editconf.gro

step5_solvate:
  paths:
    input_solute_gro_path: dependency/step4_editconf/output_gro_path
    output_gro_path: solvate.gro
    input_top_zip_path: dependency/step3_pdb2gmx/output_top_zip_path
    output_top_zip_path: solvate_top.zip
```

```python
# Loading the biobb configuration reader
conf = settings.ConfReader(sys.argv[1])

global_log, _ = fu.get_logs(path=conf.get_working_dir_path())
global_prop = conf.get_prop_dic(global_log=global_log)
global_paths = conf.get_paths_dic()

global_log.info("step1_pdb: Download the initial Structure")
Pdb(**global_paths["step1_pdb"], properties=global_prop["step1_pdb"])

global_log.info("step2_fixsidechain: Modeling the missing heavy atoms in the structure side chains")
FixSideChain(**global_paths["step2_fixsidechain"], properties=global_prop["step2_fixsidechain"])

for mutation in conf.properties['mutations_list']:

    mut_paths = conf.get_paths_dic(mutation)
    mut_prop = conf.get_prop_dic(mutation, global_log=global_log)

    mut_paths['step3_mutate']['input_pdb_path'] = global_paths['step2_fixsidechain']['output_pdb_path']

    global_log.info("step3_mutate: Modeling a particular residue mutation")
    Mutate(**mut_paths["step3_mutate"], properties=mut_prop["step3_mutate"])

    global_log.info("step4_pdb2gmx: Generate the topology")
    Pdb2gmx(**mut_paths["step4_pdb2gmx"], properties=mut_prop["step4_pdb2gmx"])
```

More complex workflows:

- **Loops / conditionals**
- Output folders hierarchy




```yaml
mutations_list: ["A:Arg5Ala", "A:Arg5Gly", "A:Arg5Lys"]

step1_pdb:
  paths:
    output_pdb_path: structure.pdb
  properties:
    pdb_code: 1aki
```

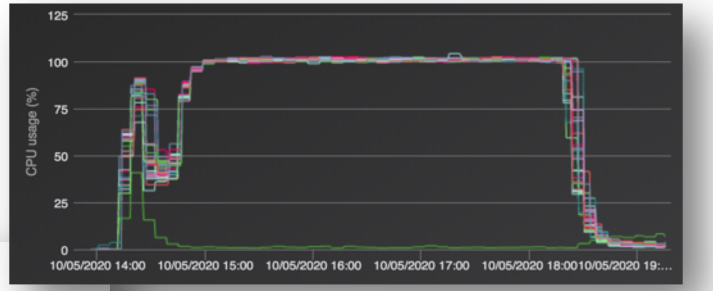
```
(biobb_Protein-Complex_MDsetup_tutorial) Adams-MacBook-Pro:Yaml hospital$ ls
biobb_MDsetup_tutorial-AlaScan.py   biobb_MDsetup_tutorial-lite.yaml    biobb_MDsetup_tutorial.py
biobb_MDsetup_tutorial-AlaScan.yaml biobb_MDsetup_tutorial-mut.py       biobb_MDsetup_tutorial.yaml
biobb_MDsetup_tutorial-lite.py      biobb_MDsetup_tutorial-mut.yaml     md_tutorial_mut
(biobb_Protein-Complex_MDsetup_tutorial) Adams-MacBook-Pro:Yaml hospital$ ls -lrht md_tutorial_mut/
total 64
-rw-r--r--   1 hospital  staff     0B Dec 11 22:43 log.err
drwxr-xr-x   5 hospital  staff   170B Dec 11 22:43 step2_fixsidechain
drwxr-xr-x   5 hospital  staff   170B Dec 11 22:43 step1_pdb
dr-xr-xr-x  24 hospital  staff   816B Dec 11 22:52 A:Arg5Ala
dr-xr-xr-x  24 hospital  staff   816B Dec 11 23:01 A:Arg5Gly
dr-xr-xr-x  24 hospital  staff   816B Dec 11 23:11 A:Arg5Lys
-rw-r--r--   1 hospital  staff    30K Dec 11 23:16 log.out
(biobb_Protein-Complex_MDsetup_tutorial) Adams-MacBook-Pro:Yaml hospital$ ls -rlht md_tutorial_mut/A\:Arg5Ala/
total 0
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:43 step5_editconf
dr-xr-xr-x  6 hospital  staff   204B Dec 11 22:43 step4_pdb2gmx
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:43 step3_mutate
dr-xr-xr-x  6 hospital  staff   204B Dec 11 22:43 step8_genion
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:43 step7_grompp_genion
dr-xr-xr-x  6 hospital  staff   204B Dec 11 22:43 step6_solvate
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:43 step9_grompp_min
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:45 step12_grompp_nvt
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:45 step11_energy_min
dr-xr-xr-x  8 hospital  staff   272B Dec 11 22:45 step10_mdrun_min
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:47 step14_energy_nvt
dr-xr-xr-x  9 hospital  staff   306B Dec 11 22:47 step13_mdrun_nvt
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:47 step15_grompp_npt
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:50 step18_grompp_md
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:50 step17_energy_npt
dr-xr-xr-x  9 hospital  staff   306B Dec 11 22:50 step16_mdrun_npt
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:52 step21_rmsexp
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:52 step20_rmsfirst
dr-xr-xr-x  9 hospital  staff   306B Dec 11 22:52 step19_mdrun_md
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:52 step24_dry
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:52 step23_image
dr-xr-xr-x  5 hospital  staff   170B Dec 11 22:52 step22_rgyr
```

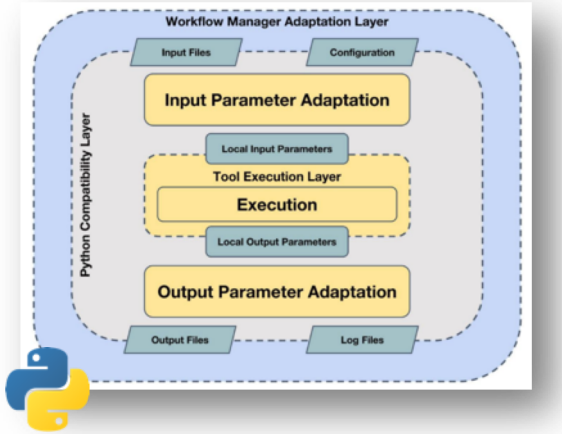
More complex workflows:

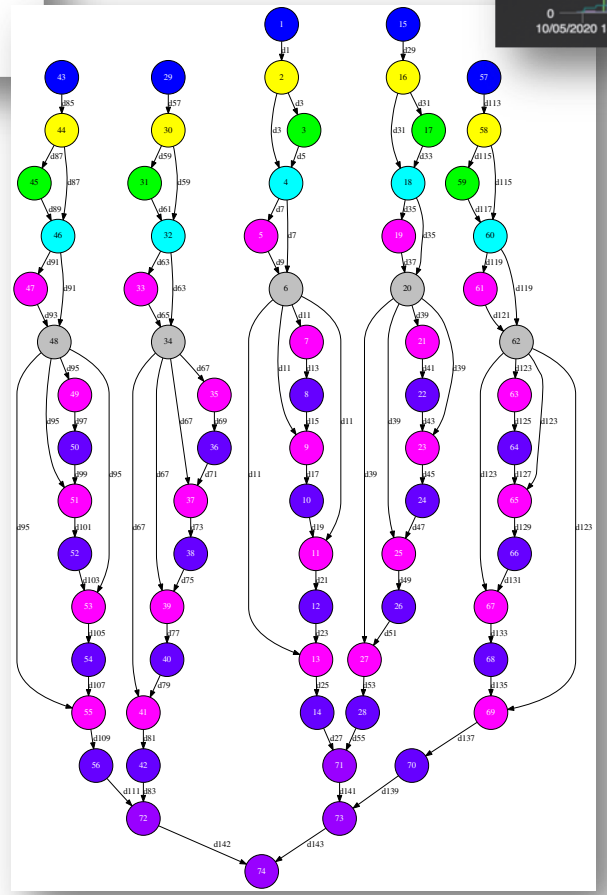
- Loops / conditionals
- **Output folders hierarchy**

BioBB HPC Workflows

```
print 'step2: mmbuniprot -- Get mutations'
mmbuniprot = uniprot.MmbVariants(prop['pdb_code'])
mutations = mmbuniprot.fetch_variants()

for mut in mutations:
    mut_path = cdir(wd, mut)

    print 'step3: scw -- Model mutation'
    scw_path = cdir(mut_path, 'step3_scw')
    scw_pdb = opj(scw_path, prop['mutated_pdb'])
    scw = scwrl.Scwrl4(mmbpdb_pdb, scw_pdb, mut, scwrl_path=scwrl_path)
    scw_pdb2 = scw.launchPyCOMPSs()
```



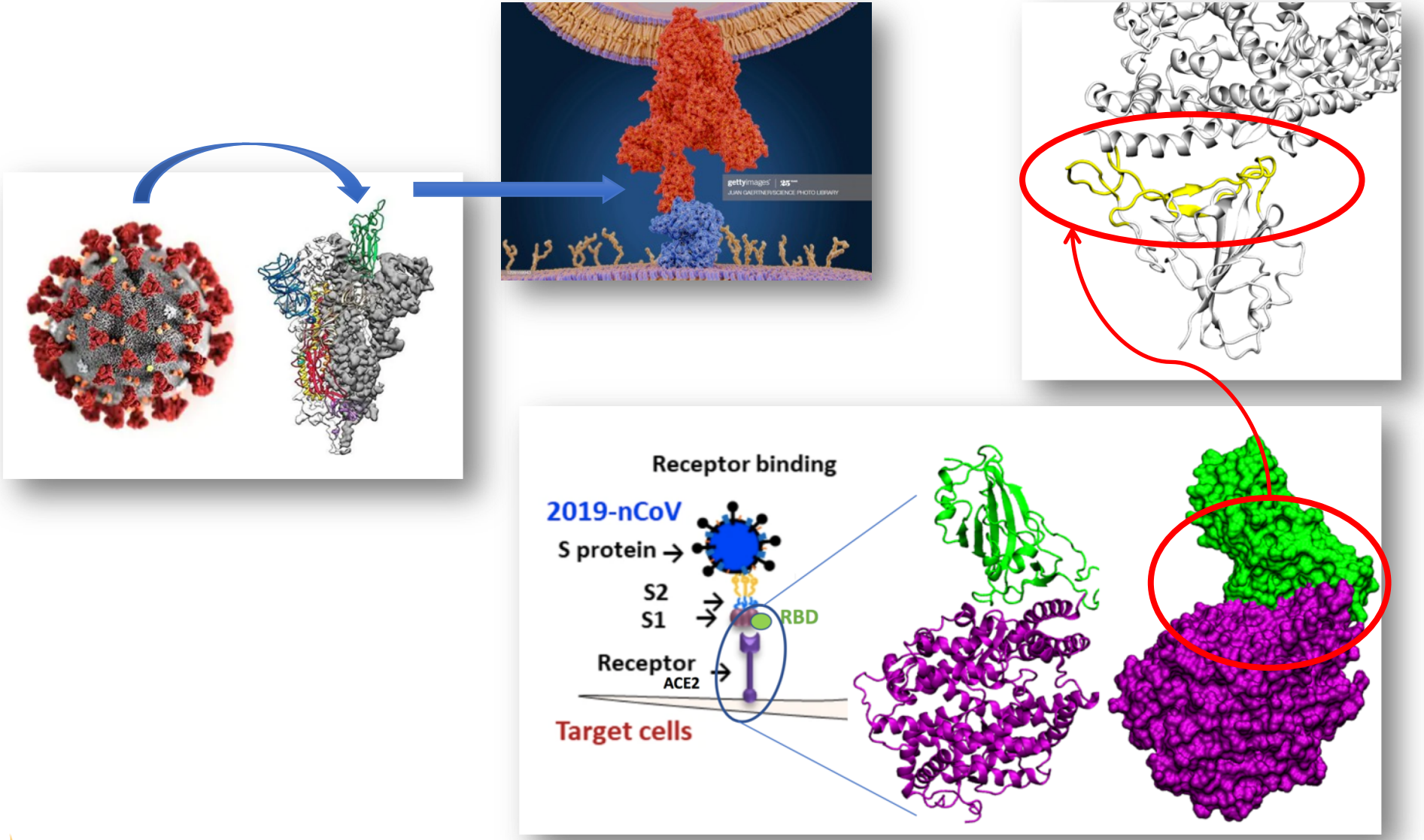




https://github.com/bioexcel/biobb_adapters

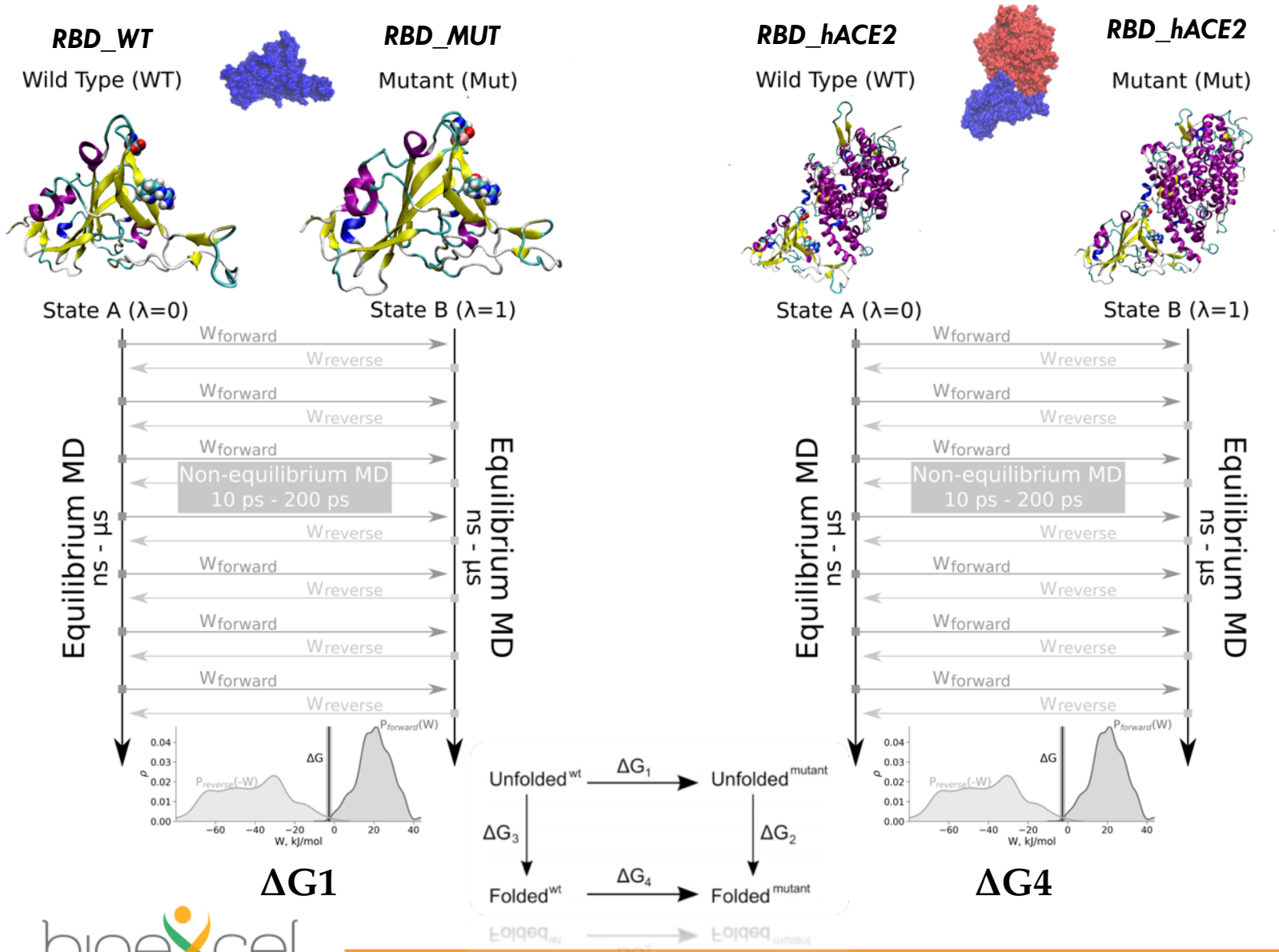
Example: pre-exascale COVID-19 BioBB workflows

COVID-19: SARS-CoV-2 RBD & hACE2



Alchemical free energy calculations
of relative protein binding free energy difference

Alchemical free energy calculations of **relative protein binding free energy** difference:
$\Delta\Delta G = \Delta G4 - \Delta G1$



Molecular Dynamics simulation data

- For each mutation:
  - MD Simulations (RBD + ACE2 + Complex)
  - Free energy calculations

Impact of mutations in binding affinity

- **Fast-growth Thermodynamic Integration**
- **1000** independent **short MD simulations** (500 forward + 500 reverse)
- **GROMACS + pmx**
- **Extremely parallelizable**

GROMACS
FAST. FLEXIBLE. FREE.

pmx: generate hybrid protein structure and topology
Computational Biomolecular Dynamics Group

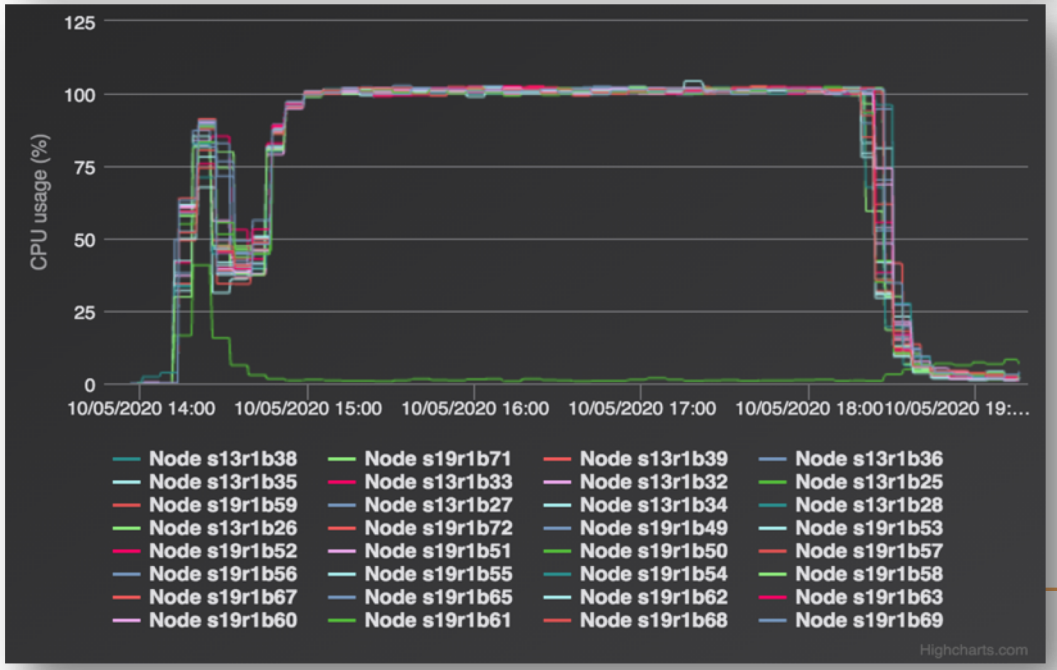
bioexcel

Molecular Dynamics simulation data

48 BSC MareNostrum nodes
2,304 cores → 1 job

12 mutations
10ns-length MDs
GROMACS 4 nodes MPI

Time: 8h



Impact of mutations in binding affinity

32 BSC MareNostrum nodes
1,536 cores → 1 job

1 mutation (RBD-ACE2)
1000 short TI MDs (50ps)
500 forward
+
500 reverse

Time: 5h

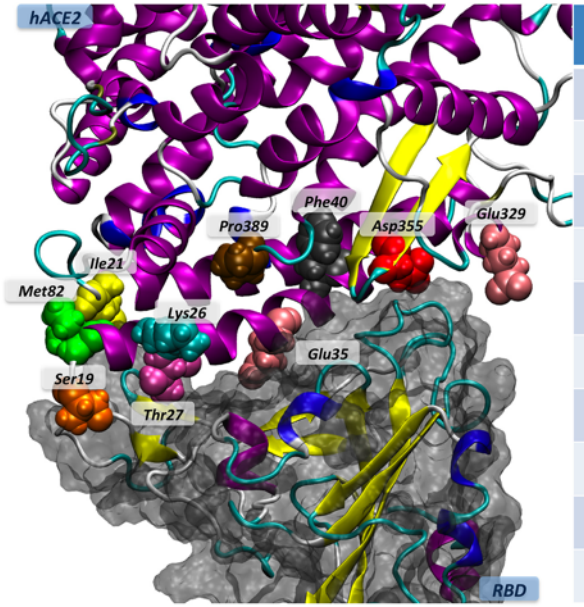
WIREs COMPUTATIONAL MOLECULAR SCIENCE

Focus Article | Free Access

Pre-exascale HPC approaches for molecular dynamics simulations. Covid-19 research: A use case

Miłosz Wieczór, Vito Genna, Juan Aranda, Rosa M. Badia, Josep Lluís Gelpí, Vytautas Gapsys, Bert L. de Groot, Erik Lindahl, Martí Municoy, Adam Hospital ✉, Modesto Orozco ✉

First published: 30 May 2022 | https://doi-org.sire.ub.edu/10.1002/wcms.1622



Mutation	Frequency	SARS-CoV-2 RBD hACE2 ΔΔG
Ser19Pro	0.000313	0.86
Ile21Val	0.000011	-0.65
Lys26Arg	0.003883	-1.12
Thr27Ala	0.000011	0.1
Glu35Lys	0.000016	4.25
Phe40Leu	0.000061	1.02
Met82Ile	0.000024	1.44
Glu329Gly	0.000034	0.36
Asp355Asn	0.000012	3.42
Pro389His	0.000038	1.36

RETURN TO ISSUE | < PREV COMPUTATIONAL BIOCHE... NEXT >

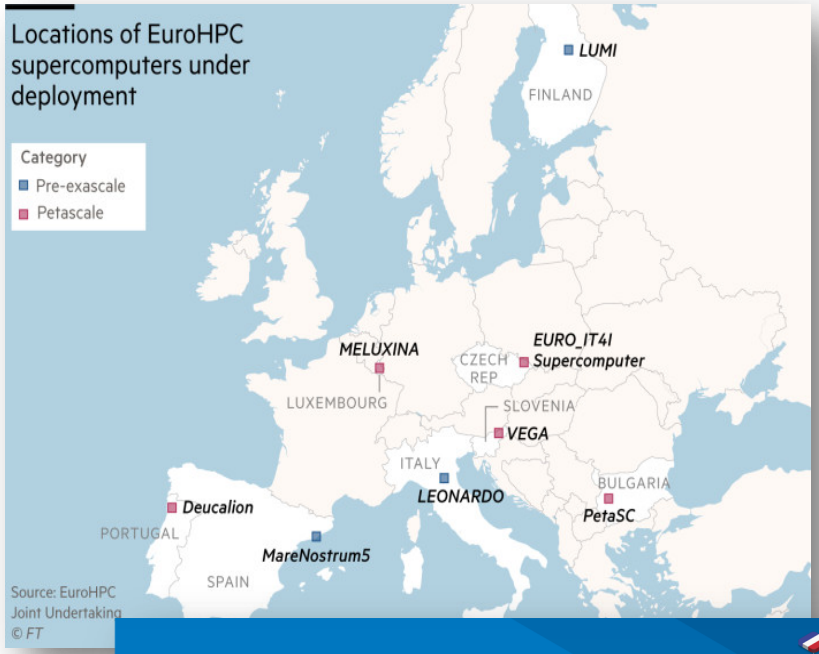
High-Throughput Prediction of the Impact of Genetic Variability on Drug Sensitivity and Resistance Patterns for Clinically Relevant Epidermal Growth Factor Receptor Mutations from Atomistic Simulations

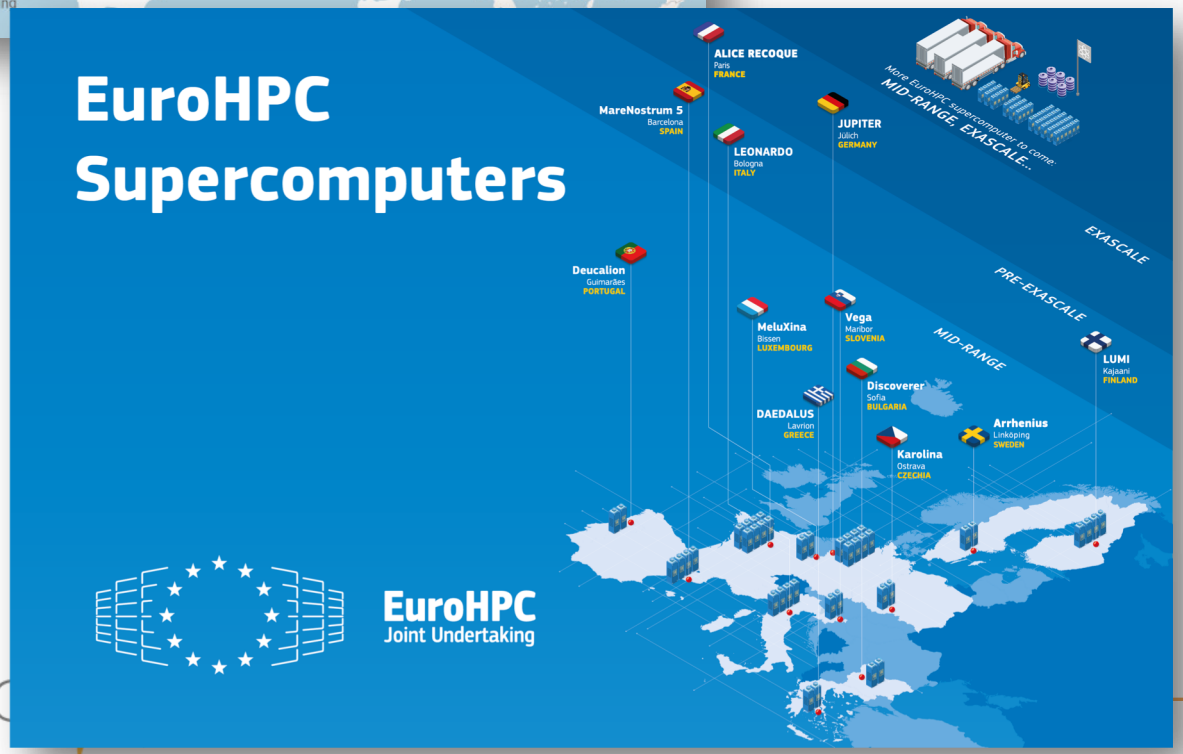
Aristarc Suriñach, Adam Hospital, Yvonne Westermaier, Luis Jordà, Sergi Orozco-Ruiz, Daniel Beltrán, Francesco Colizzi, Pau Andrio, Robert Soliva, Martí Municoy, Josep Lluís Gelpí, and Modesto Orozco*

Cite this: *J. Chem. Inf. Model.* 2023, 63, 1, 321−334
Publication Date: December 28, 2022
https://doi.org/10.1021/acs.jcim.2c01344

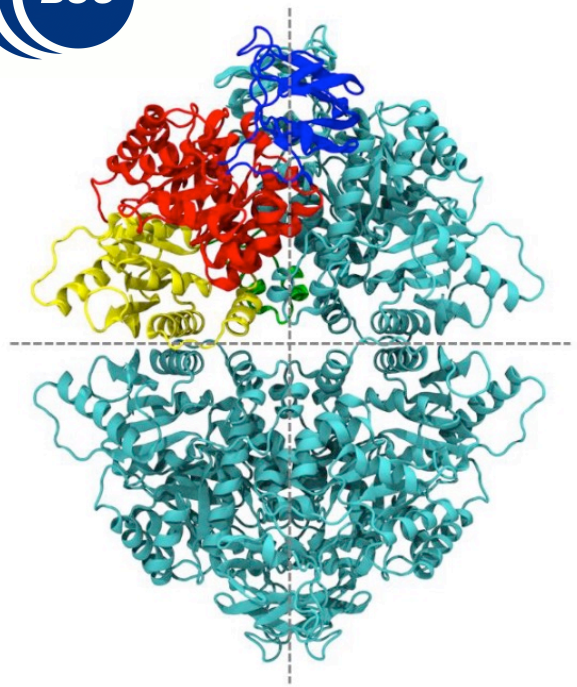
Mutation	Drug	Eprof (pred)	PELE* (pred)	ΔΔG(binding) FE/MD	Exp. Impact$^{\Delta}$
L718Q	Osimertinib	R	-	18.7(0.8) R	Resistance[1]
G719S	Gefitinib	S	S	-5.5(0.9) S	Sensitive[2]
G719S	Icotinib	S	-	-0.9(0.8)$^{\#}$ S	Sensitive[2]
G719S	Erlotinib	S	S	-1.1(0.3) S	Sensitive[3]
G719S	Lapatinib	S	S	-5.1(2.8) S	Sensitive[4]
L747S	Gefitinib	*S*	*S*	13.8(1.8) R	Resistance[5]
L747F	Osimertinib	*S*	-	4.4(1.2) R	Resistance[6]
L747H	Osimertinib	*S*	-	6.5(3.6) R	Resistance[6]
S768I	Gefitinib	S	*R*	-1.8(0.4) S	Sensitive[7]
V769M	Gefitinib	S	S	-7.7(2.2) S	Sensitive[8]
T790M	Gefitinib	*S*	*S*	15.5(0.9) R	Resistance[9]
T790M	Erlotinib	*S*	R	17.6(1.9) R	Resistance[9]
T790M	Lapatinib	*S*	R	19.3(0.9) R	Resistance[10]
T790M	Osimertinib	S	-	-0.5(2.1) S	Sensitive[11]
T790M	Icotinib	S	-	11.5(0.6) R	Resistance[12]
L792F	Osimertinib	S	-	4.2(1.0) R	Resistance[13]
L792H	Osimertinib	R	-	8.8(0.1) R	Resistance[13]
G796S	Osimertinib	*S*	-	4.8(0.4) R	Resistance[14]
C797G$^{\&}$	Osimertinib	R	R	Resistance	Resistance[15]
C797S$^{\&}$	Osimertinib	R	R	Resistance	Resistance[16]
L833V	Gefitinib	S	S	-5.7(1.4) S	Sensitive[17]
H835L	Gefitinib	S	S	-7.5(1.3)$^{\#}$ S	Sensitive[17]
L838V	Gefitinib	S	S	-6.3(3.1) S	Sensitive[18]
T854A	Gefitinib	R	S	13.2(1.1) R	Resistance[5]
L861Q	Gefitinib	S	S	-5.4(0.7)$^{\#}$ S	Sensitive[19]
T790M/C797S	Erlotinib	R	R	6.2(2.4) R	Resistance[20]

EuroHPC Supercomputers





- Lumi: CSC – IT Center for Science (Finland)
  Peak performance: 550 petaflops
- Leonardo: CINECA (Italy)
  Peak performance: 323,40 petaflops
- Marenostrum5:Barcelona Supercomputing Center (Spain)
  Peak performance: 314 petaflops
- Vega: IZUM (Slovenia)
  Peak performance: 10,05 petaflops
- Meluxina: LuxProvide (Luxemburg)
  Peak performance: 18,29 petaflops
- Karolina:IT4Innovations (Czech Rep.)
  Peak performance: 15,69 petaflops
- Discoverer: Sofia Tech Park (Bulgaria)
  Peak performance: 5,94 petaflops
- Deucalion: MACC (Portugal)
  Peak performance: 10 petaflops



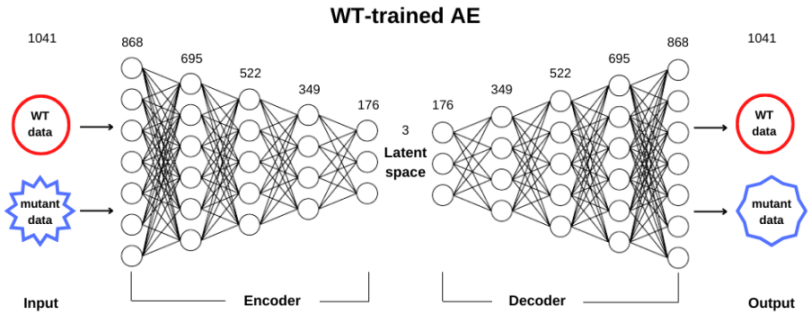
Simulation based on WT PK (2VGB)
400ns for Apo WT and selected variants

WT trajectory

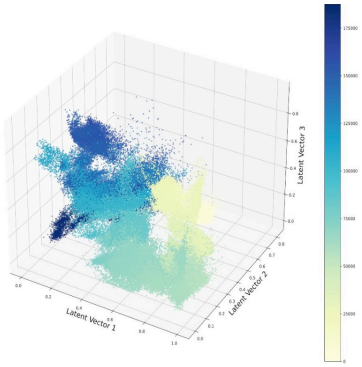
Trained WT Autoencoder

Variant trajectory

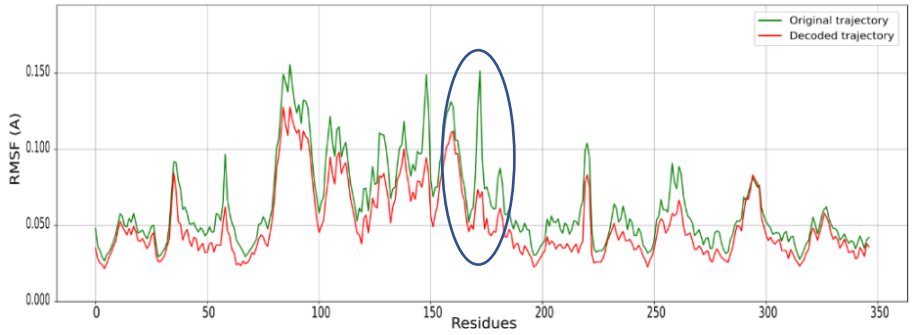
Decoded variant trajectory

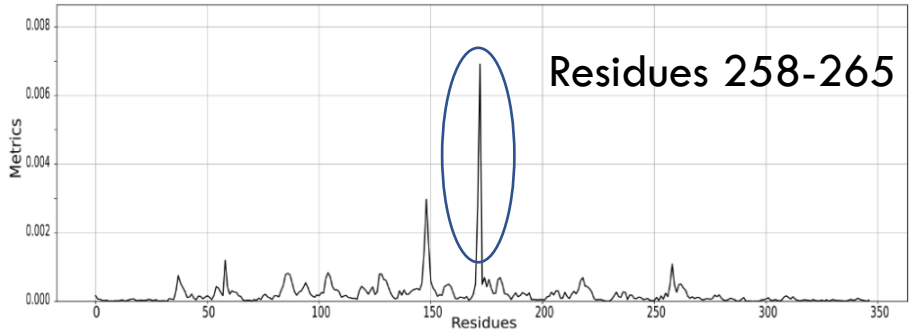


Autoassociative Neural Network trained on WT simulation



3D latent space

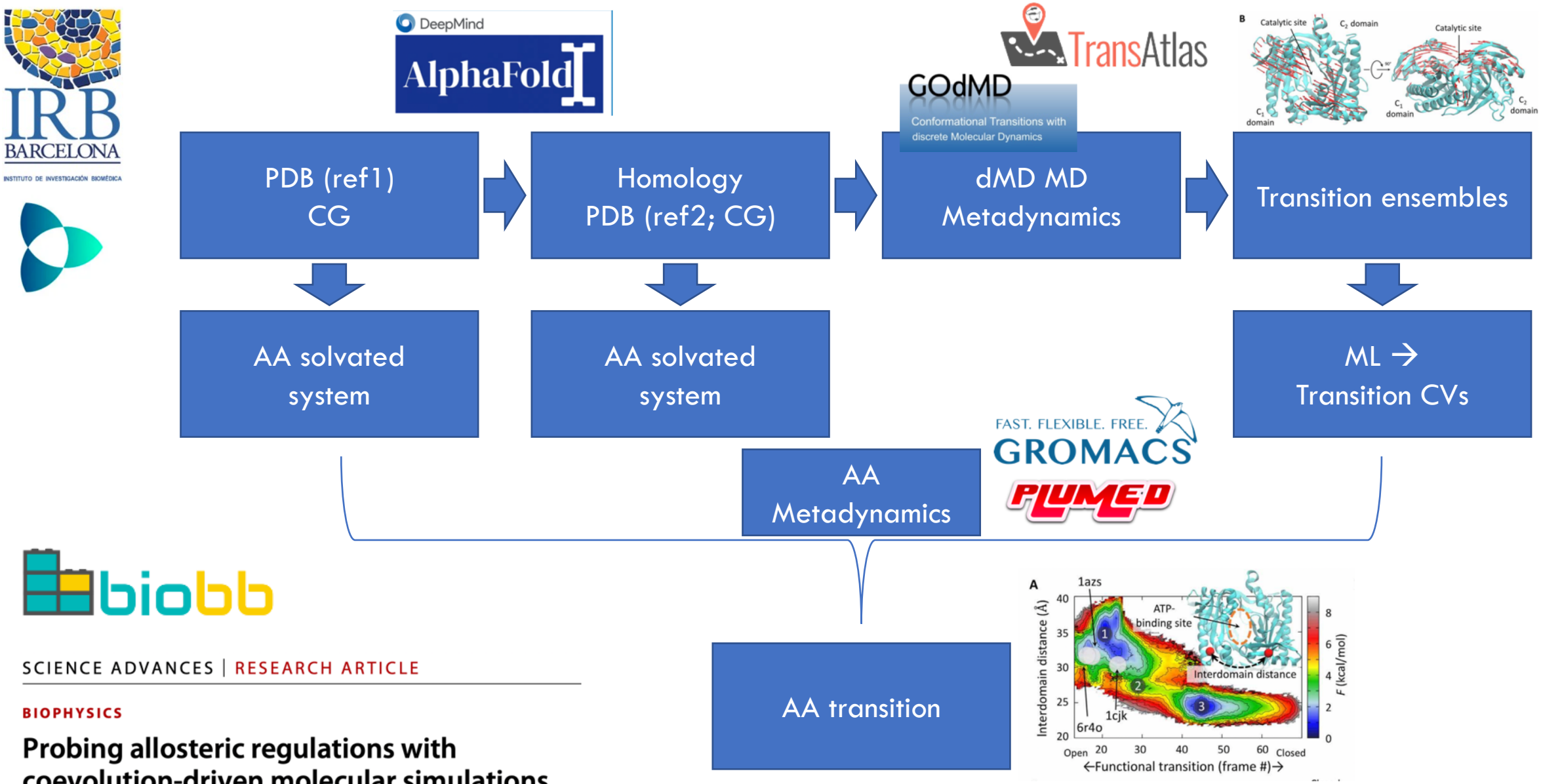




Altered Dynamics on Gly332Ser variant revealed by ΔRMSf on decoded trajectories

biobb_pytorch

Showcase 2: Coevolution-driven metadynamics simulations from CG conformational transitions and ML-derived transition coordinates



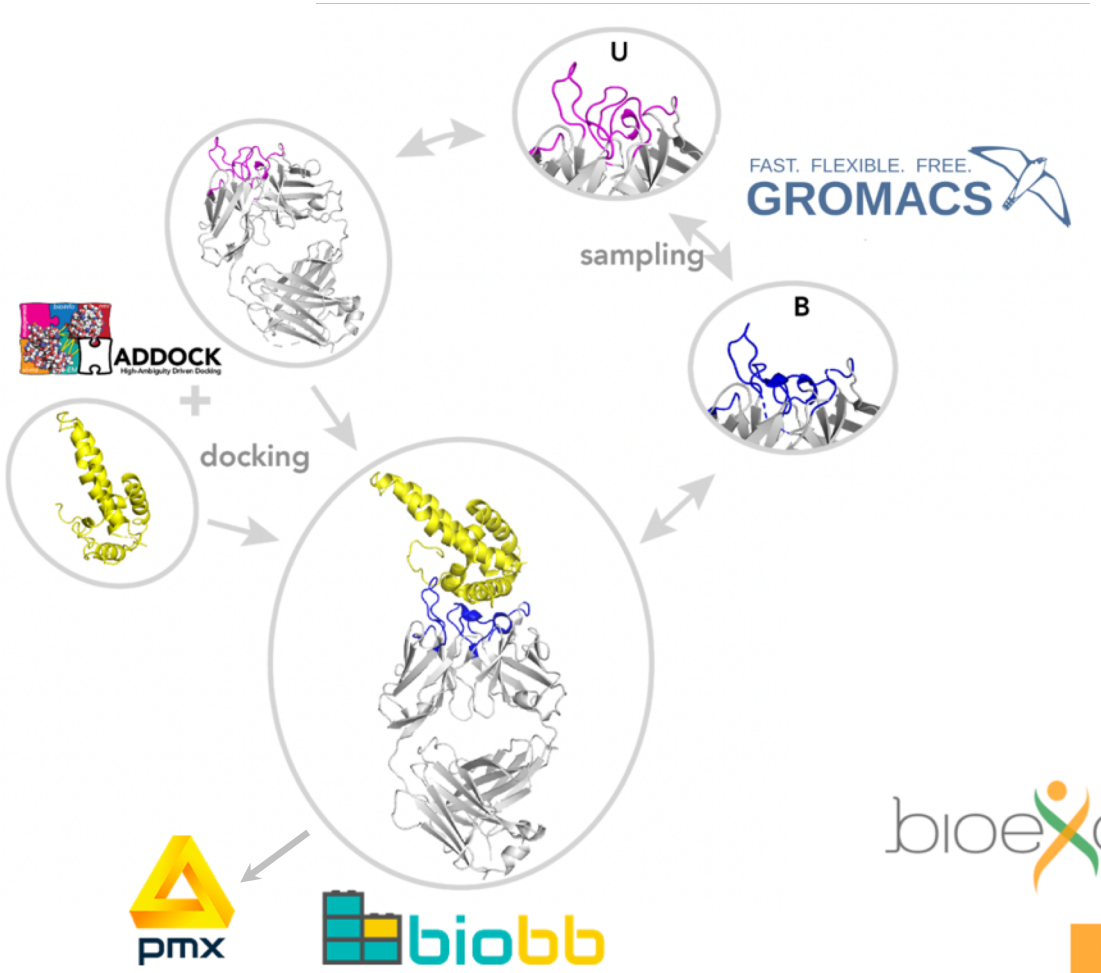
SCIENCE ADVANCES | RESEARCH ARTICLE

BIOPHYSICS

Probing allosteric regulations with coevolution-driven molecular simulations

Francesco Colizzi[1]*† and Modesto Orozco[1,2]*

Showcase 3: BioExcel Antibody Design - a BioExcel use case including GROMACS, HADDOCK, PMX and BioBB



ADDOCK High-Ambiguity Driven Docking

FAST. FLEXIBLE. FREE. GROMACS

biobb

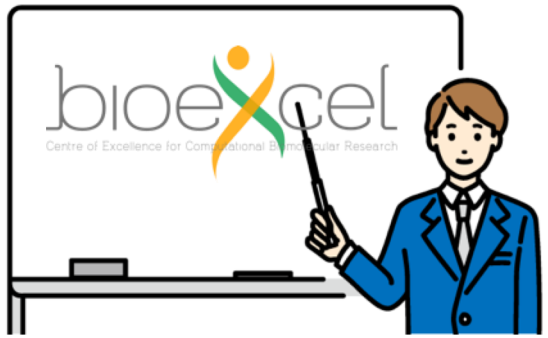
pmx

bioexcel

Outreach

Training

Webinars



Summary

- **BioExcel Building Blocks** software library offers a new layer of interoperability on biomolecular software tools.
- **Biomolecular workflows**, portable and reproducible, can be easily built using the library.
- The additional **workflow manager compatibility layer** allows **workflows** built with the library to be launched and controlled with **different frameworks** and in **different infrastructures:**
  - **Demonstration** workflows (Jupyter Notebooks)
  - **Pre-exascale HPC** workflows (PyCOMPSs)

BioBB offers:

- ✓ **Tools Interoperability: workflows easy to build**
- ✓ **Workflows reproducibility and portability**

BioBB is NOT magic:

- ❖ **You still need to know about the tool wrapped**
- ❖ **You still need to build your own workflows**

Next session: Workflow Tutorial

Acknowledgments & Questions

Federica Battistini

Genís Bayarri

Modesto Orozco

Pau Andrio

Josep Ll. Gelpí

BioExcel Center of Excellence, funded from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreements No. 675728, 823830, 101093290 (BioExcel-1, BioExcel-2 and BioExcel-3).