bioexcel
Centre of Excellence for Computational Biomolecular Research

Horizon 2020
European Union Funding
for Research & Innovation

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

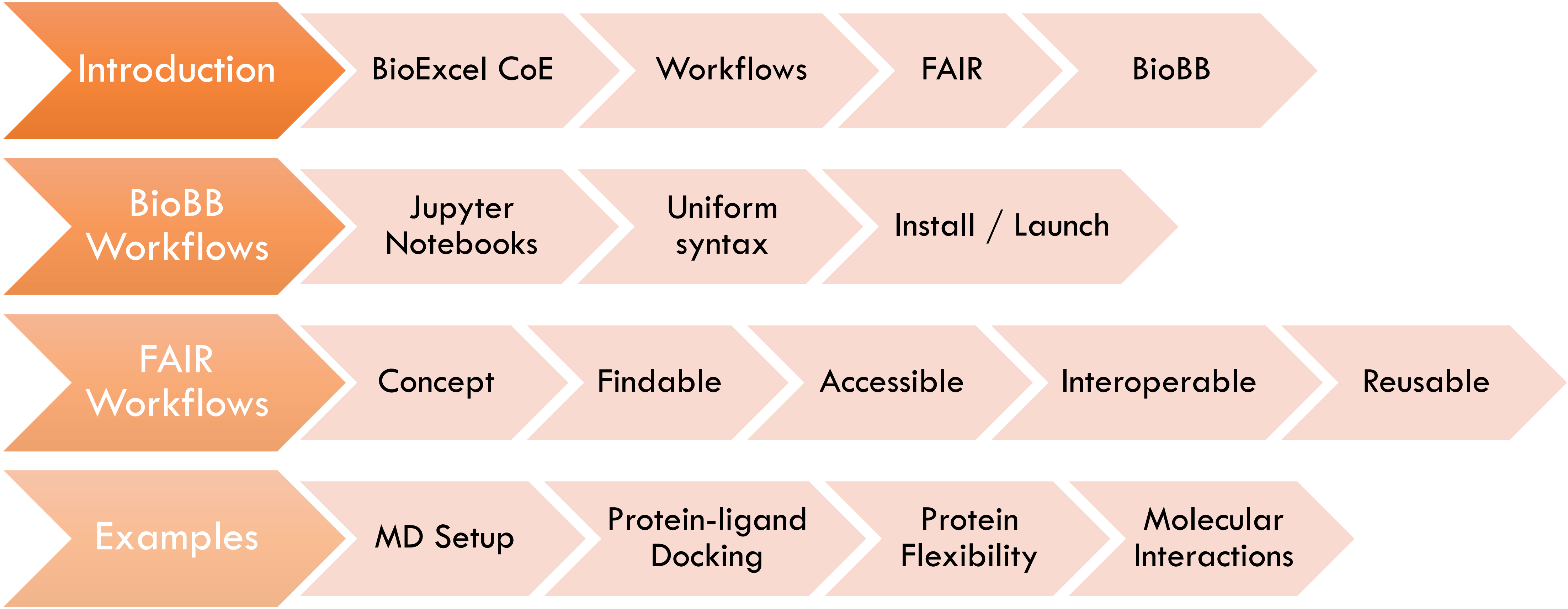
BioExcel Webinar, 2024-05-28

Adam Hospital

Institute for Research in Biomedicine, IRB-Barcelona

adam.hospital@irbbarcelona.org

Index



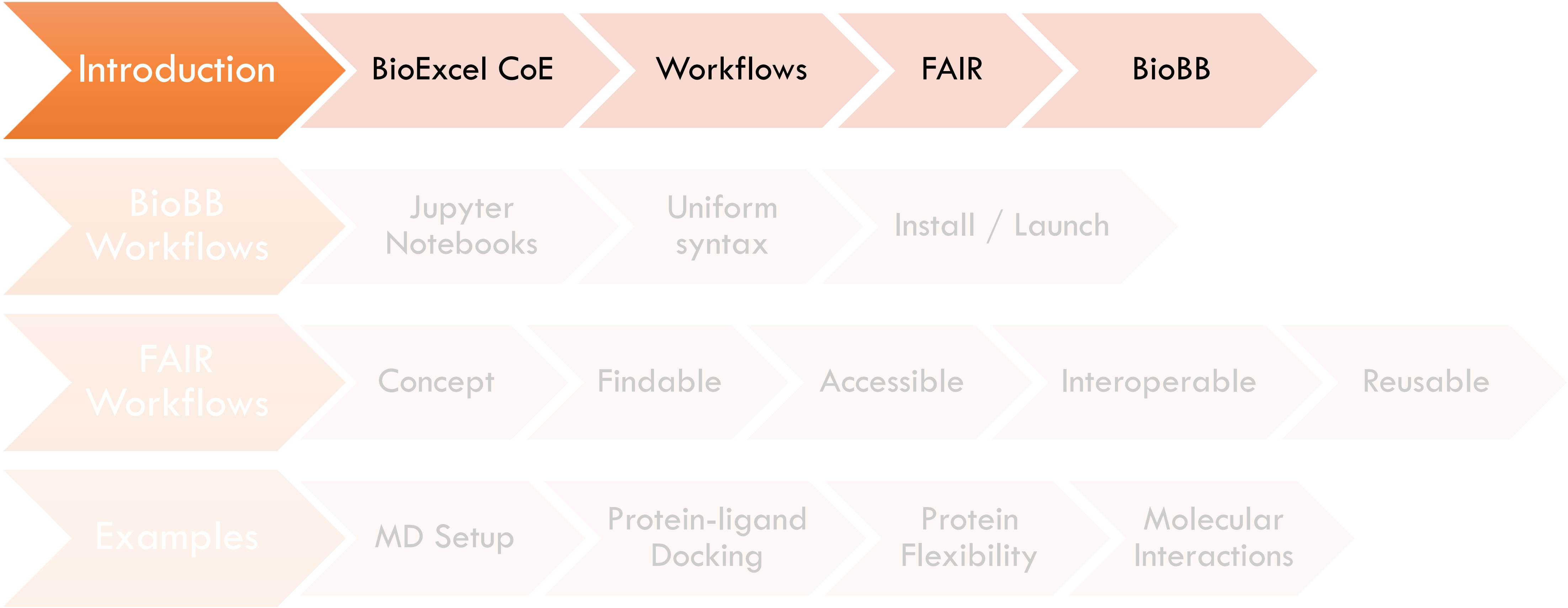
Introduction: BioExcel CoE, Workflows, FAIR, BioBB

BioBB Workflows: Jupyter Notebooks, Uniform syntax, Install / Launch

FAIR Workflows: Concept, Findable, Accessible, Interoperable, Reusable

Examples: MD Setup, Protein-ligand Docking, Protein Flexibility, Molecular Interactions

Index



Introduction – BioExcel CoE – Workflows – FAIR – BioBB

BioBB Workflows – Jupyter Notebooks – Uniform syntax – Install / Launch

FAIR Workflows – Concept – Findable – Accessible – Interoperable – Reusable

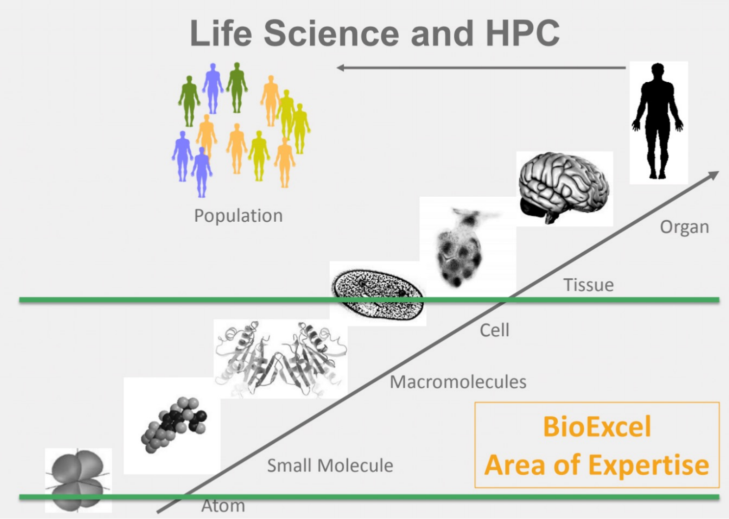
Examples – MD Setup – Protein-ligand Docking – Protein Flexibility – Molecular Interactions

bioexcel

HORIZON 2020

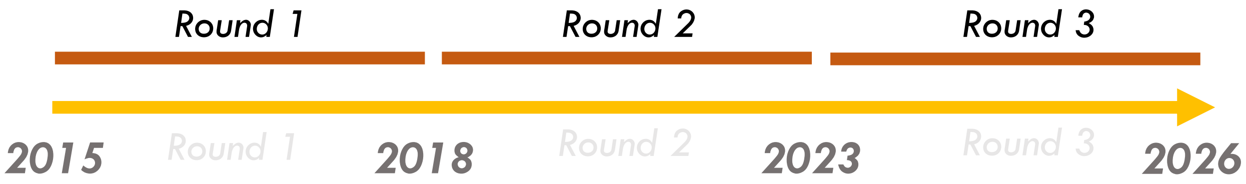
Centre of Excellence for Computational Biomolecular Research

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

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

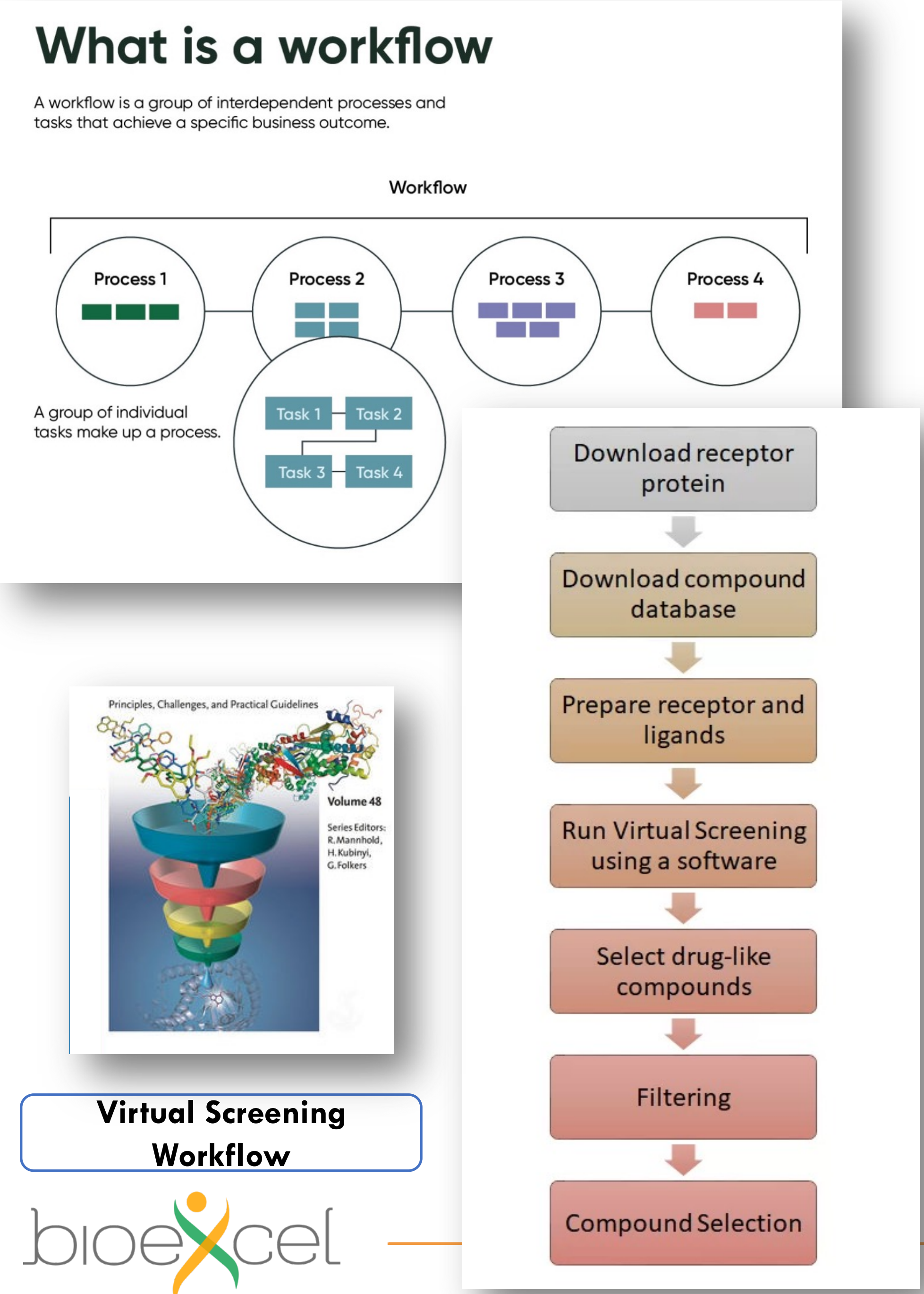
CSC

NORMAN CONSULTING

Co-funded by the European Union

EuroHPC Joint Undertaking

https://bioexcel.eu/



What is a workflow

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

Download receptor protein → Download compound database → Prepare receptor and ligands → Run Virtual Screening using a software → Select drug-like compounds → Filtering → Compound Selection

Virtual Screening Workflow

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 MOLECULAR OPERATING ENVIROMENT

FAST. FLEXIBLE. FREE.
GROMACS

AMBER 22

AmberTools19

OpenMM

HADDOCK
High-Ambiguity Driven Docking

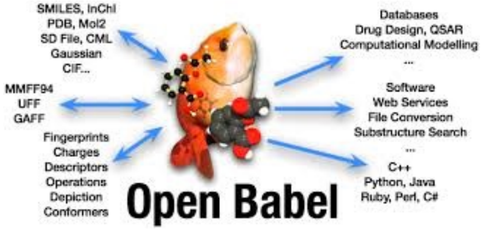
NAMD
Scalable Molecular Dynamics

VMD
Visual Molecular Dynamics

CPMD QM/MM

CP2K

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

pmx

MDTRAJ

Gaussian, Inc.

Bio3D

MD ANALYSIS

AutoDock 4

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

TensorFlow

scikit learn

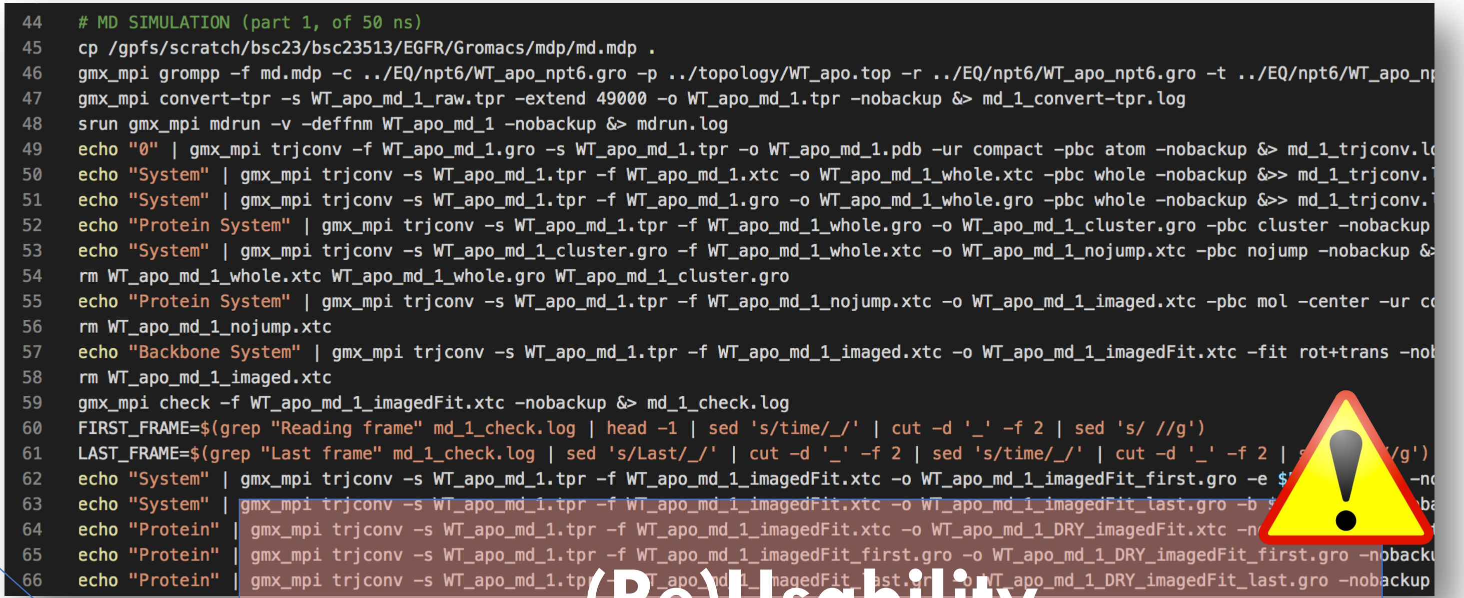
And many many more…

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 -nob
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 | s          /g')
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 $          -no
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 $          ba
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
```

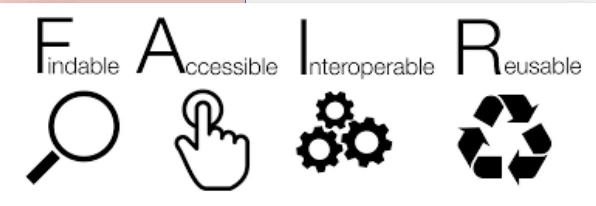
(Re)Usability

Interoperability

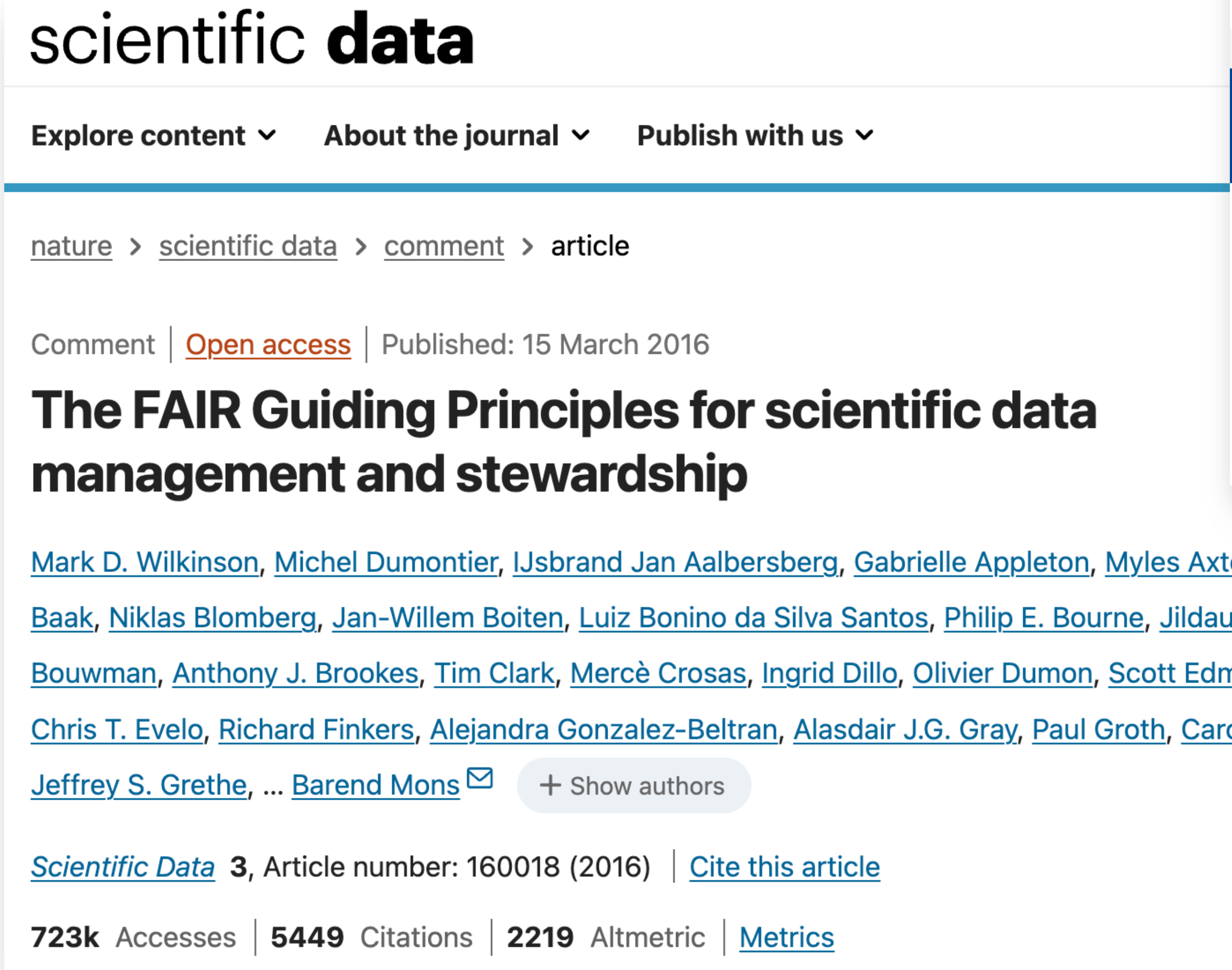
Portability

Reproducibility

Scalability



FAIR Data Principles



scientific **data**

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › scientific data › comment › article

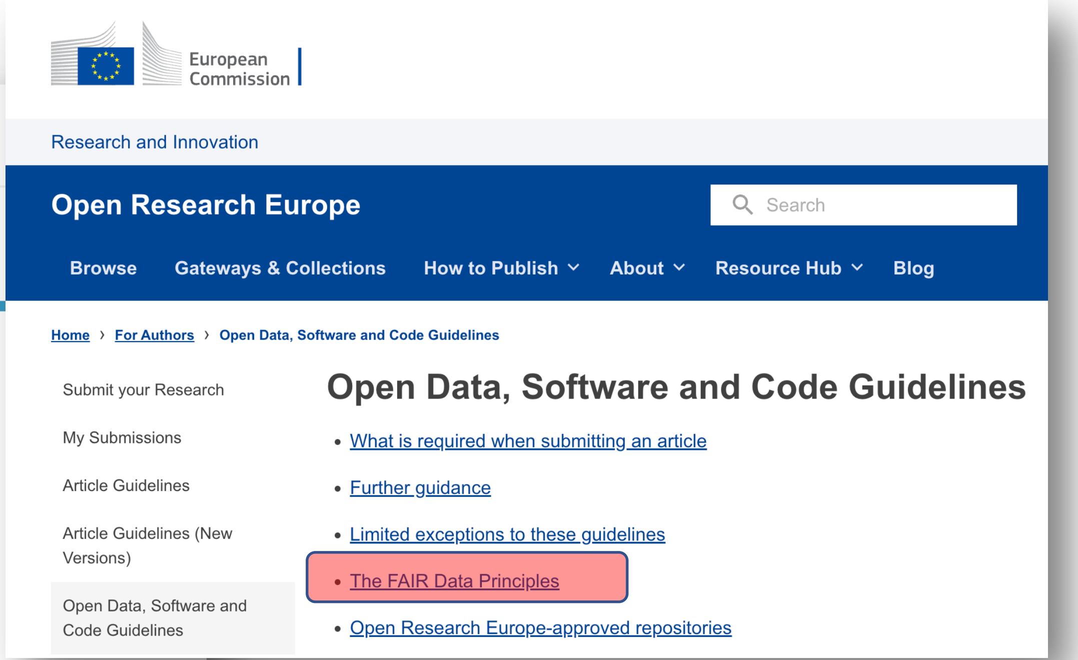
Comment | Open access | Published: 15 March 2016

The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, ... Barend Mons ✉ + Show authors

Scientific Data **3**, Article number: 160018 (2016) | Cite this article

723k Accesses | **5449** Citations | **2219** Altmetric | Metrics



European Commission

Research and Innovation

Open Research Europe Search

Browse Gateways & Collections How to Publish ⌄ About ⌄ Resource Hub ⌄ Blog

Home › For Authors › Open Data, Software and Code Guidelines

Submit your Research

My Submissions

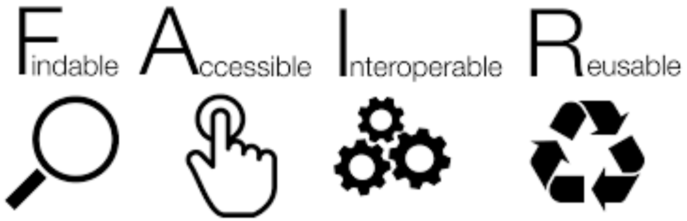
Article Guidelines

Article Guidelines (New Versions)

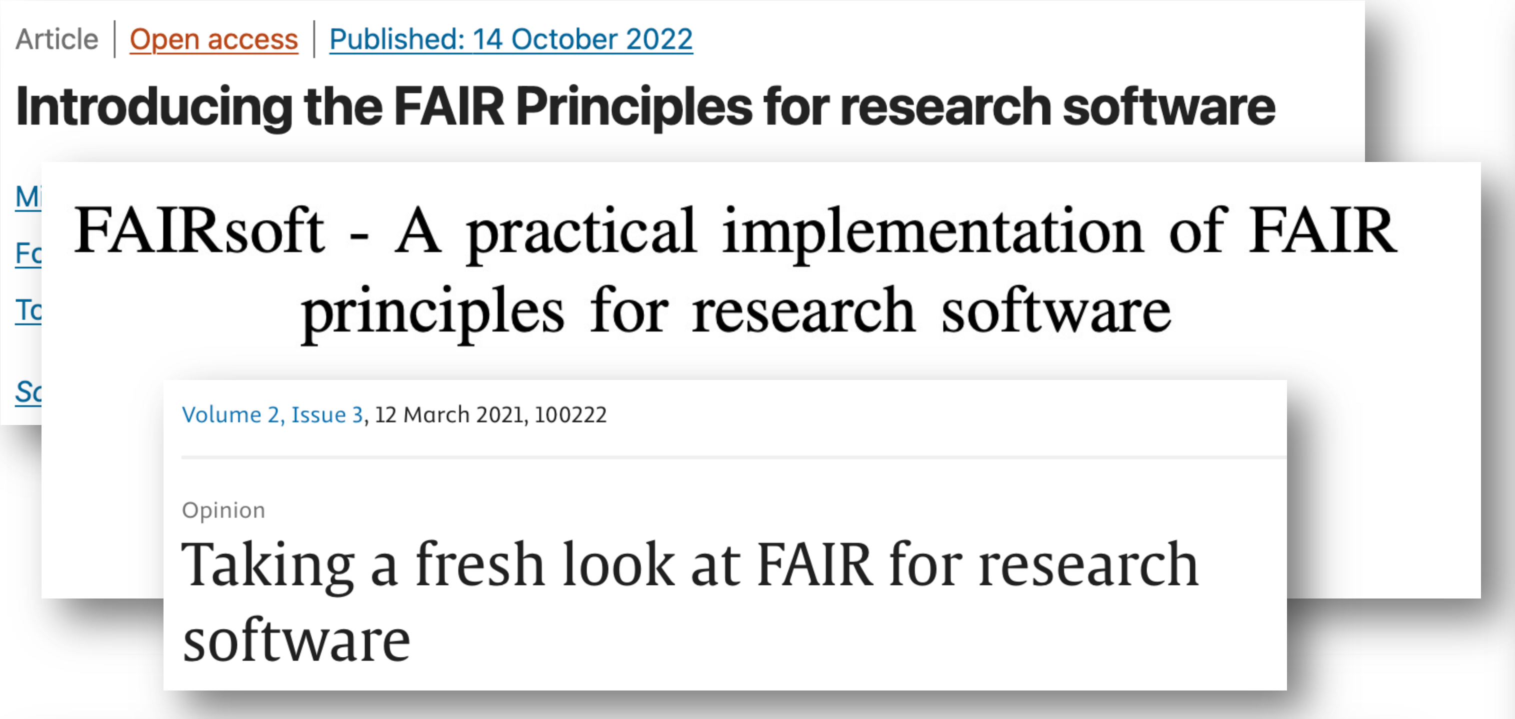
Open Data, Software and Code Guidelines

Open Data, Software and Code Guidelines

- What is required when submitting an article
- Further guidance
- Limited exceptions to these guidelines
- The FAIR Data Principles
- Open Research Europe-approved repositories



FAIR Principles for Research Software



Article | Open access | Published: 14 October 2022

Introducing the FAIR Principles for research software

Mi

Fc

Tc

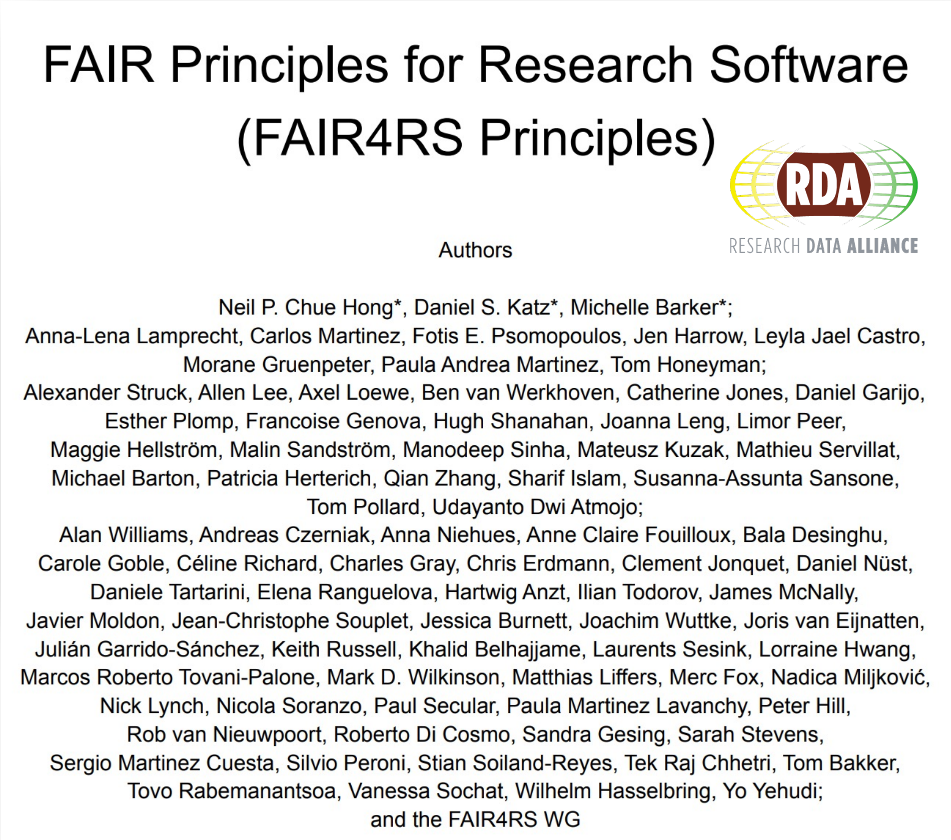
Sc

FAIRsoft - A practical implementation of FAIR principles for research software

Volume 2, Issue 3, 12 March 2021, 100222

Opinion

Taking a fresh look at FAIR for research software

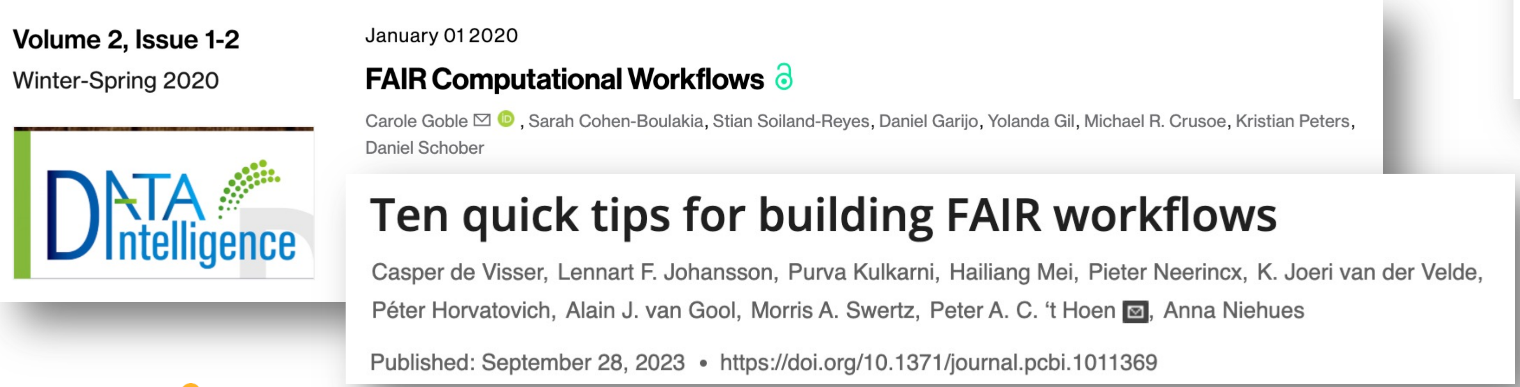


FAIR Principles for Research Software (FAIR4RS Principles)

RDA RESEARCH DATA ALLIANCE

Authors

Neil P. Chue Hong*, Daniel S. Katz*, Michelle Barker*;
Anna-Lena Lamprecht, Carlos Martinez, Fotis E. Psomopoulos, Jen Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, Tom Honeyman;
Alexander Struck, Allen Lee, Axel Loewe, Ben van Werkhoven, Catherine Jones, Daniel Garijo, Esther Plomp, Francoise Genova, Hugh Shanahan, Joanna Leng, Limor Peer, Maggie Hellström, Malin Sandström, Manodeep Sinha, Mateusz Kuzak, Mathieu Servillat, Michael Barton, Patricia Herterich, Qian Zhang, Sharif Islam, Susanna-Assunta Sansone, Tom Pollard, Udayanto Dwi Atmojo;
Alan Williams, Andreas Czerniak, Anna Niehues, Anne Claire Fouilloux, Bala Desinghu, Carole Goble, Céline Richard, Charles Gray, Chris Erdmann, Clement Jonquet, Daniel Nüst, Daniele Tartarini, Elena Ranguelova, Hartwig Anzt, Ilian Todorov, James McNally, Javier Moldon, Jean-Christophe Souplet, Jessica Burnett, Joachim Wuttke, Joris van Eijnatten, Julián Garrido-Sánchez, Keith Russell, Khalid Belhajjame, Laurents Sesink, Lorraine Hwang, Marcos Roberto Tovani-Palone, Mark D. Wilkinson, Matthias Liffers, Merc Fox, Nadica Miljković, Nick Lynch, Nicola Soranzo, Paul Secular, Paula Martinez Lavanchy, Peter Hill, Rob van Nieuwpoort, Roberto Di Cosmo, Sandra Gesing, Sarah Stevens, Sergio Martinez Cuesta, Silvio Peroni, Stian Soiland-Reyes, Tek Raj Chhetri, Tom Bakker, Tovo Rabemanantsoa, Vanessa Sochat, Wilhelm Hasselbring, Yo Yehudi;
and the FAIR4RS WG



Volume 2, Issue 1-2

Winter-Spring 2020

DATA Intelligence

January 01 2020

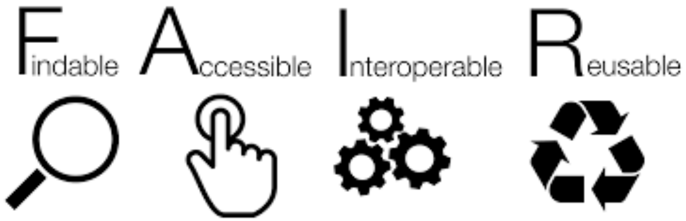
FAIR Computational Workflows

Carole Goble, Sarah Cohen-Boulakia, Stian Soiland-Reyes, Daniel Garijo, Yolanda Gil, Michael R. Crusoe, Kristian Peters, Daniel Schober

Ten quick tips for building FAIR workflows

Casper de Visser, Lennart F. Johansson, Purva Kulkarni, Hailiang Mei, Pieter Neerincx, K. Joeri van der Velde, Péter Horvatovich, Alain J. van Gool, Morris A. Swertz, Peter A. C. 't Hoen, Anna Niehues

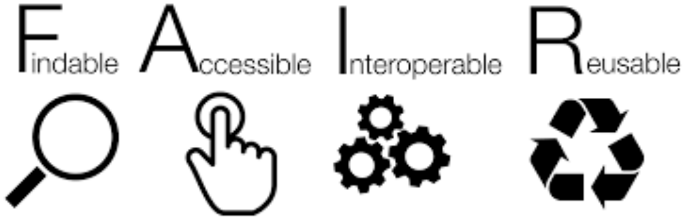
Published: September 28, 2023 • https://doi.org/10.1371/journal.pcbi.1011369



bioexcel

Biomolecular workflows: BioExcel

Biomolecular workflow


Findable Accessible Interoperable Reusable

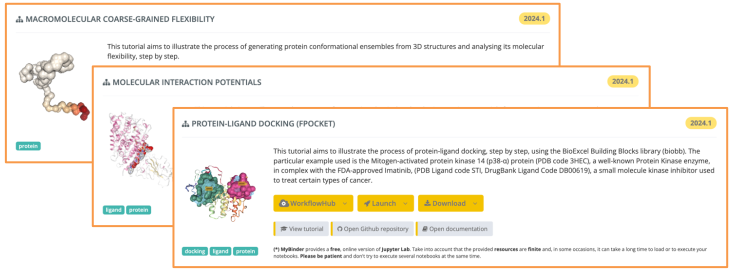
Centre of Excellence for Computational Biomolecular Research

A central hub for biomolecular modelling and simulations

European life sciences infrastructure

Best practices in research software/workflows development

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


MACROMOLECULAR COARSE-GRAINED FLEXIBILITY 2024.1

This tutorial aims to illustrate the process of generating protein conformational ensembles from 3D structures and analysing its molecular flexibility, step by step.

protein

MOLECULAR INTERACTION POTENTIALS 2024.1

ligand protein

PROTEIN-LIGAND DOCKING (FPOCKET) 2024.1

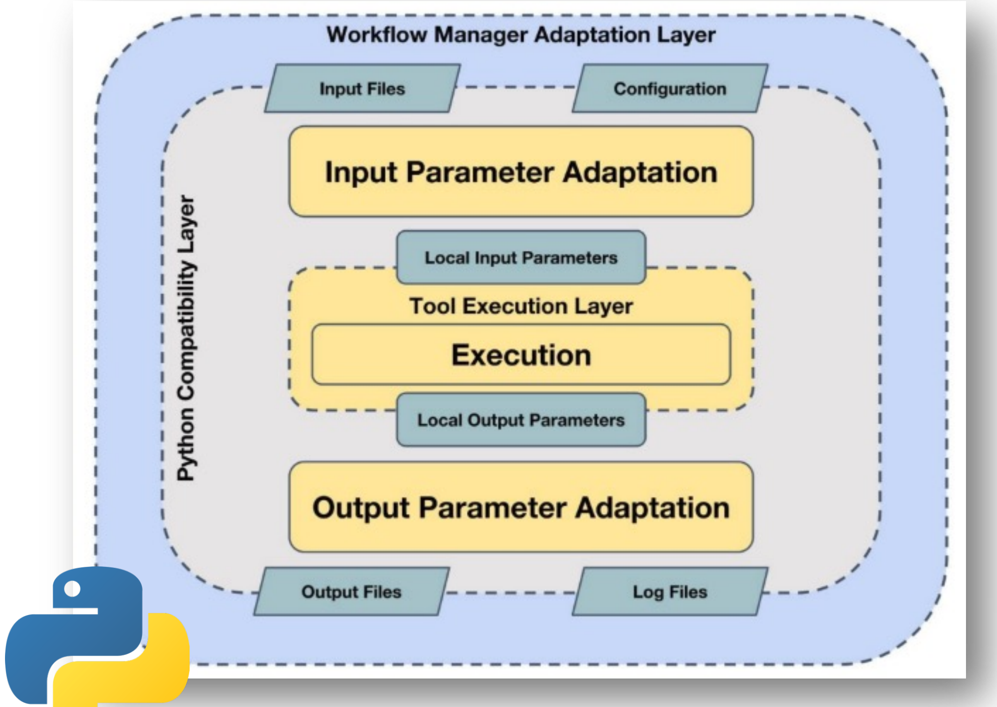
This tutorial aims to illustrate the process of protein-ligand docking, step by step, using the BioExcel Building Blocks library (biobb). The particular example used is the Mitogen-activated protein kinase 14 (p38-α) protein (PDB code 3HEC), a well-known Protein Kinase enzyme, in complex with the FDA-approved Imatinib, (PDB Ligand code STI, DrugBank Ligand Code DB00619), a small molecule kinase inhibitor used to treat certain types of cancer.

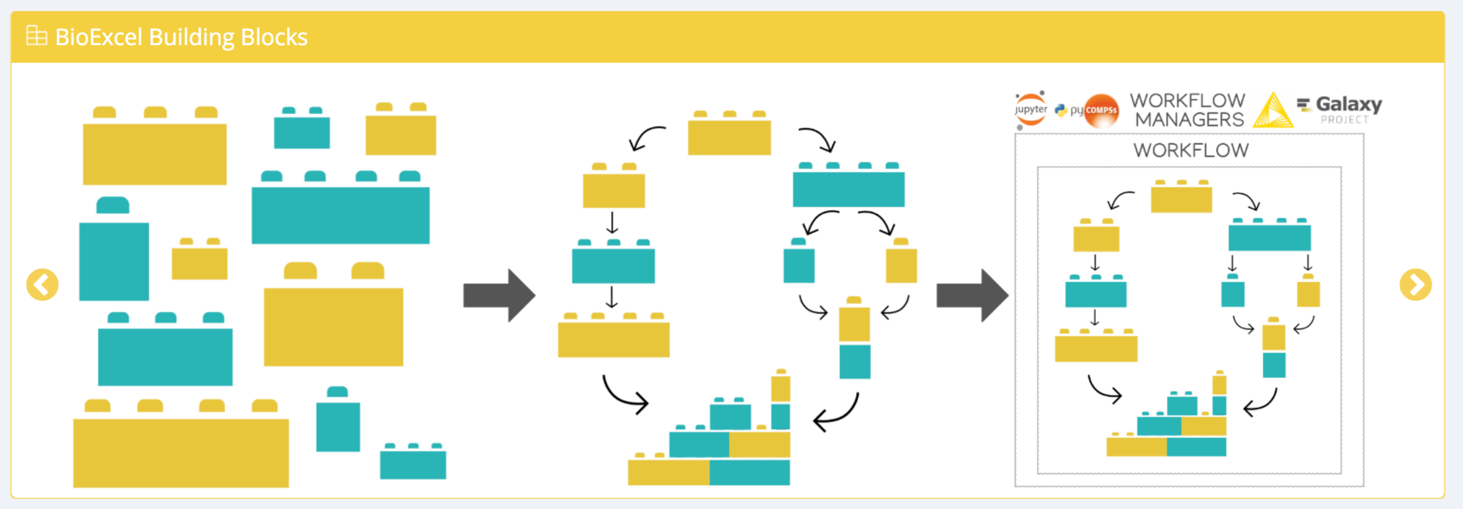
(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

docking ligand protein

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

BioExcel Building Blocks: BioBB



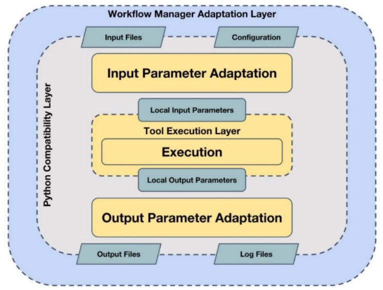


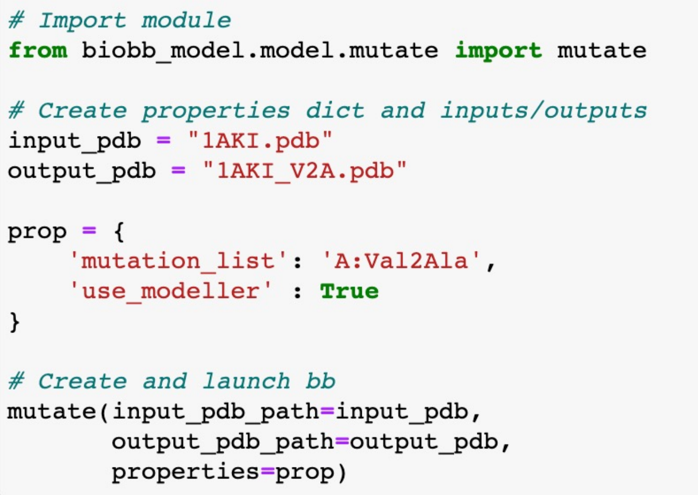
1) Building Blocks *2) Workflows* *3) Workflow Managers*

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

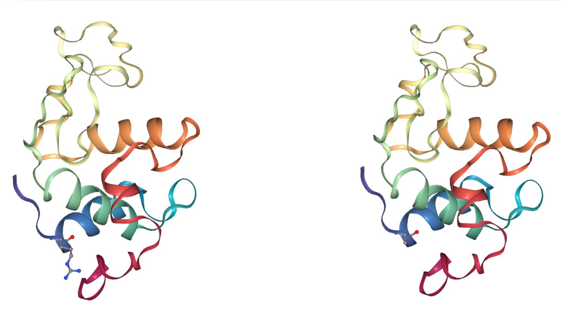
BioBB Syntax

- **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)
```

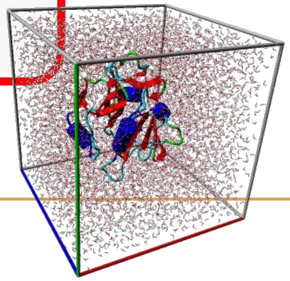


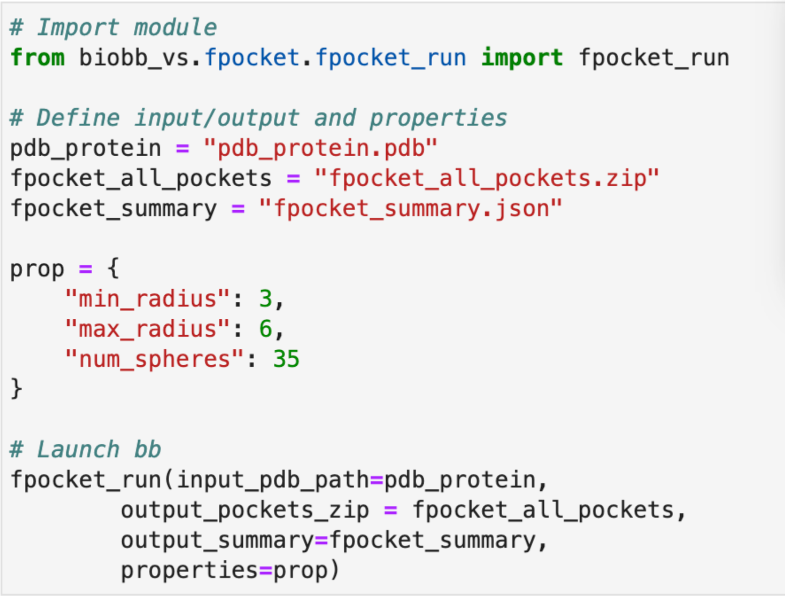
Modeller
Program for Comparative Protein Structure Modelling by Satisfaction of Spatial Restraints

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

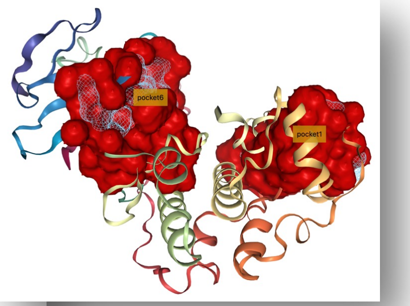
```python
# Create prop dict and inputs/outputs
input_pdb2gmx_gro = '1AKI_pdb2gmx.gro'
output_editconf_gro = '1AKI_editconf.gro'

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)
```

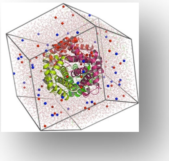


fpocket

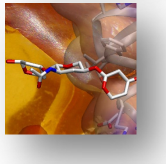
FAST. FLEXIBLE. FREE.
GROMACS

BioBB Modules (17, Release 2024.1)

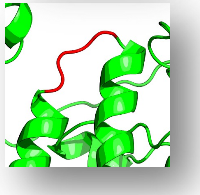
biobb_common
Common auxiliar functions



biobb_gromacs
Molecular Dynamics GROMACS

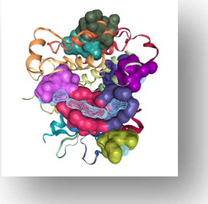


biobb_amber
Molecular Dynamics AMBER



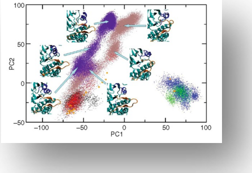
biobb_model
Molecular Modelling

biobb_ml
Machine learning

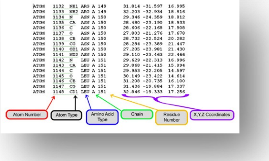


biobb_vs
Virtual Screening

biobb_io
Biological databases

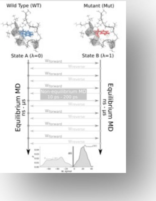


biobb_analysis
MD trajectories analysis

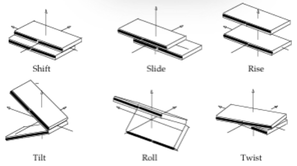

Atom Number
Atom Type
Amino Acid Type
Chain
Residue Number
X,Y,Z Coordinates

biobb_structure_utils
Modify or extract information from PDB

biobb_chemistry
Chemoinformatics functionalities



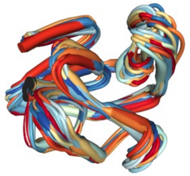
biobb_pmx
Free energy calculations



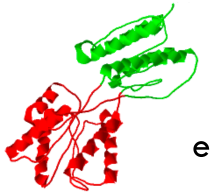
biobb_dna
Nucleic Acids MD Trajectory analyses

biobb_cmip
Molecular Interaction Potentials

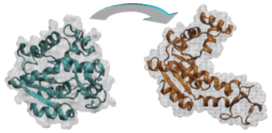
biobb_cp2k
Quantum Mechanics



biobb_flexdyn
NMA-based conformational ensemble generation



biobb_flexserv
Coarse-Grained conformational ensemble generation and flexibility analysis

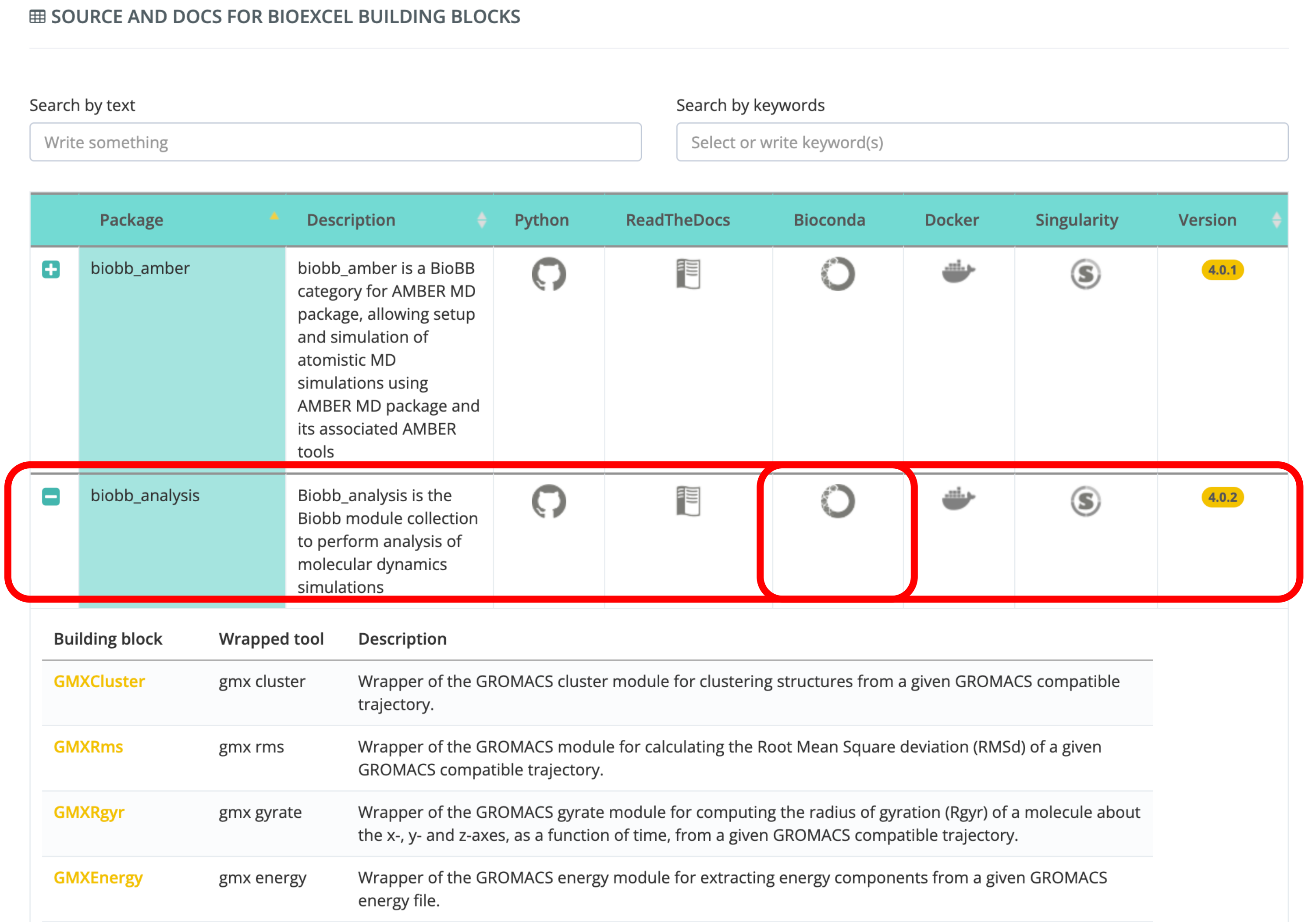


biobb_godmd
Coarse-Grained conformational transitions

biobb *Release 2024.1*

And many others to come…

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	●	●	○	●	S	4.0.1
−	biobb_analysis	Biobb_analysis is the Biobb module collection to perform analysis of molecular dynamics simulations	●	●	○	●	S	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

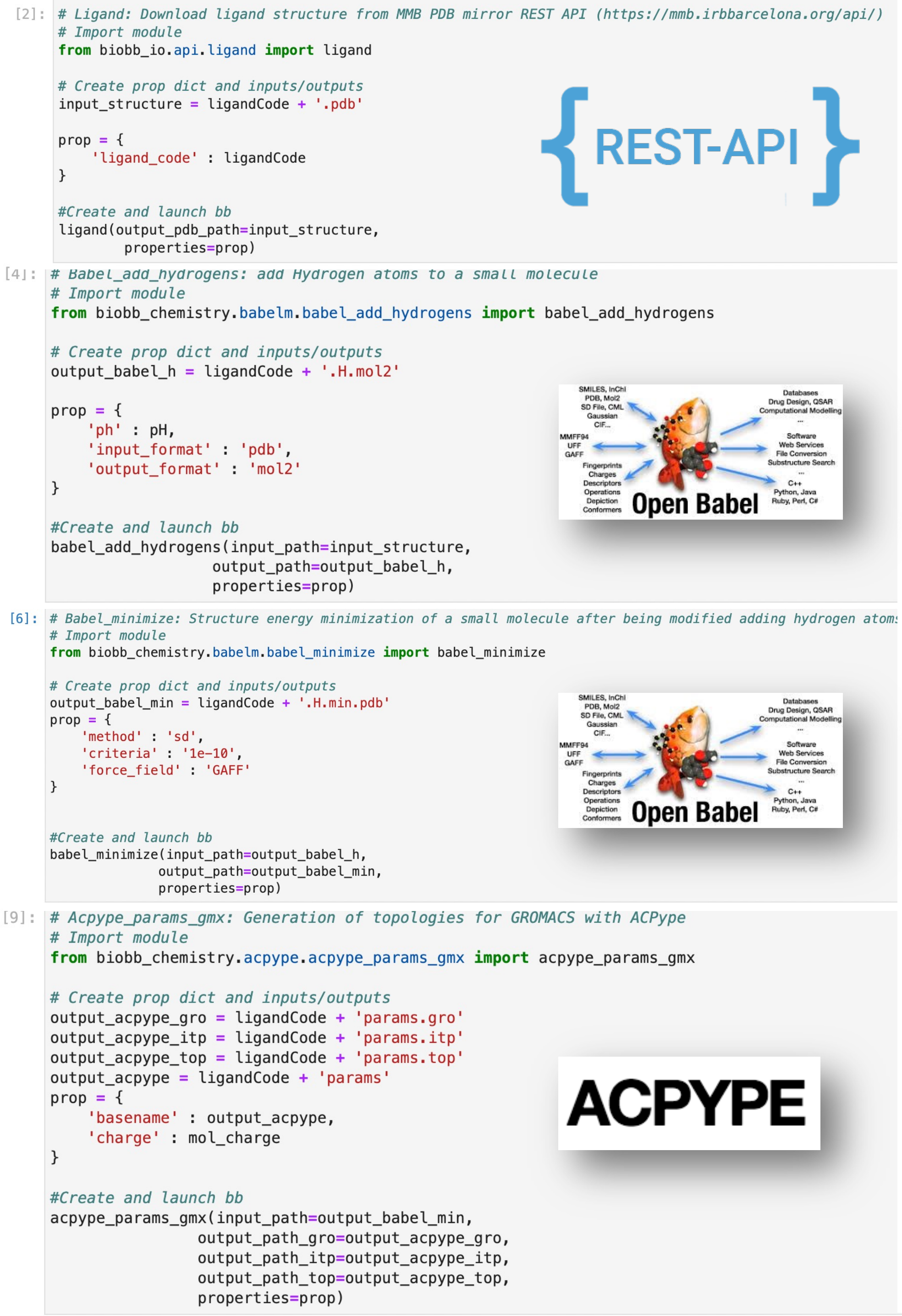
conda install biobb_vs

Autodock Vina

fpocket

scalable high performance pocket detection

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



```
[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)
```

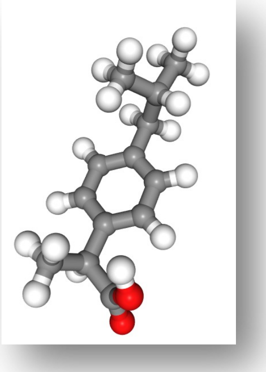
```
[4]: # Babel_add_hydrogens: add Hydrogen atoms to a small molecule
     # Import module
     from biobb_chemistry.babelm.babel_add_hydrogens import babel_add_hydrogens

     # 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)
```

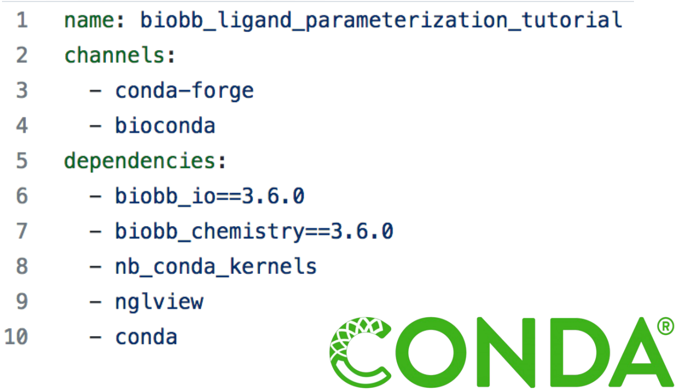
```
[6]: # Babel_minimize: Structure energy minimization of a small molecule after being modified adding hydrogen atoms
     # Import module
     from biobb_chemistry.babelm.babel_minimize import babel_minimize

     # 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)
```

```
[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)
```

Ligand parameterization workflow:

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



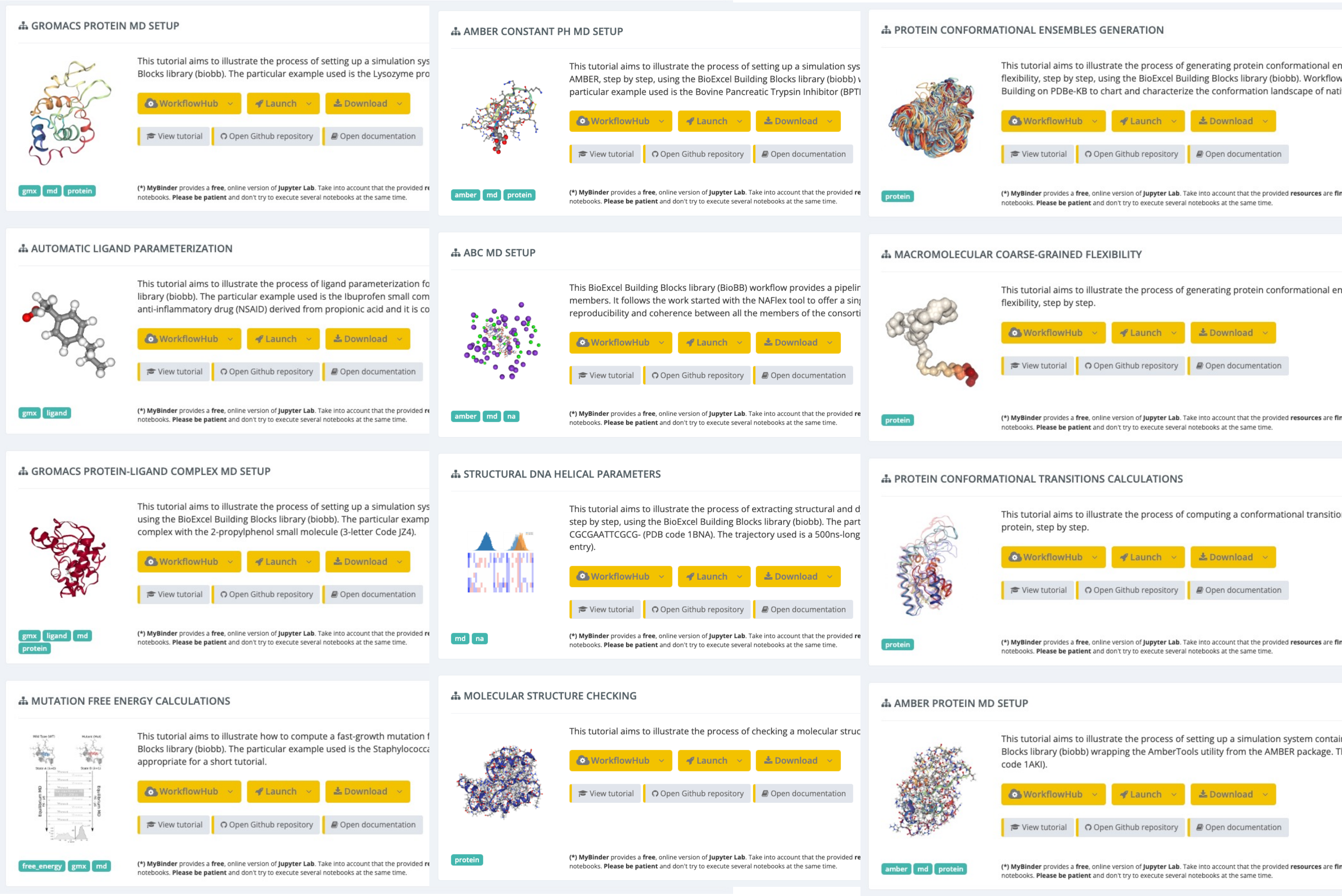


```
name: biobb_ligand_parameterization_tutorial
channels:
  - conda-forge
  - bioconda
dependencies:
  - biobb_io==3.6.0
  - biobb_chemistry==3.6.0
  - nb_conda_kernels
  - nglview
  - conda
```

BioBB Demonstration Workflows

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

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



GROMACS PROTEIN MD SETUP

This tutorial aims to illustrate the process of setting up a simulation sys Blocks library (biobb). The particular example used is the Lysozyme pro

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

AUTOMATIC LIGAND PARAMETERIZATION

This tutorial aims to illustrate the process of ligand parameterization fo library (biobb). The particular example used is the Ibuprofen small com anti-inflammatory drug (NSAID) derived from propionic acid and it is co

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

GROMACS PROTEIN-LIGAND COMPLEX MD SETUP

This tutorial aims to illustrate the process of setting up a simulation sys using the BioExcel Building Blocks library (biobb). The particular examp complex with the 2-propylphenol small molecule (3-letter Code JZ4).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

MUTATION FREE ENERGY CALCULATIONS

This tutorial aims to illustrate how to compute a fast-growth mutation f Blocks library (biobb). The particular example used is the Staphylococca appropriate for a short tutorial.

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

AMBER CONSTANT PH MD SETUP

This tutorial aims to illustrate the process of setting up a simulation sys AMBER, step by step, using the BioExcel Building Blocks library (biobb) v particular example used is the Bovine Pancreatic Trypsin Inhibitor (BPTI

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

ABC MD SETUP

This BioExcel Building Blocks library (BioBB) workflow provides a pipelir members. It follows the work started with the NAFlex tool to offer a sing reproducibility and coherence between all the members of the consorti

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

STRUCTURAL DNA HELICAL PARAMETERS

This tutorial aims to illustrate the process of extracting structural and d step by step, using the BioExcel Building Blocks library (biobb). The part CGCGAATTCGCG- (PDB code 1BNA). The trajectory used is a 500ns-long entry).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

MOLECULAR STRUCTURE CHECKING

This tutorial aims to illustrate the process of checking a molecular struc

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **re** notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

PROTEIN CONFORMATIONAL ENSEMBLES GENERATION

This tutorial aims to illustrate the process of generating protein conformational en flexibility, step by step, using the BioExcel Building Blocks library (biobb). Workflow Building on PDBe-KB to chart and characterize the conformation landscape of nati

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are fin notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

MACROMOLECULAR COARSE-GRAINED FLEXIBILITY

This tutorial aims to illustrate the process of generating protein conformational en flexibility, step by step.

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are fin notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

PROTEIN CONFORMATIONAL TRANSITIONS CALCULATIONS

This tutorial aims to illustrate the process of computing a conformational transitio protein, step by step.

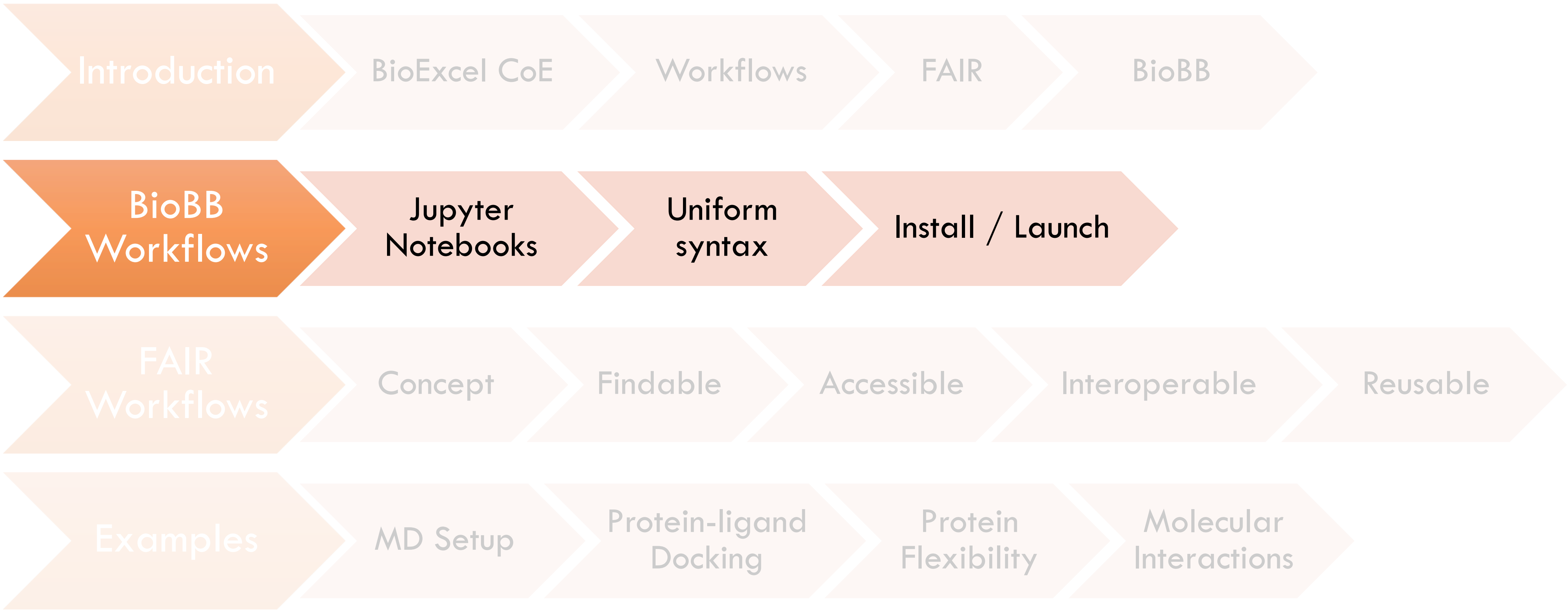
(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are fin notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

AMBER PROTEIN MD SETUP

This tutorial aims to illustrate the process of setting up a simulation system contai Blocks library (biobb) wrapping the AmberTools utility from the AMBER package. Tl code 1AKI).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are fin notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

Index



Introduction: BioExcel CoE, Workflows, FAIR, BioBB

BioBB Workflows: Jupyter Notebooks, Uniform syntax, Install / Launch

FAIR Workflows: Concept, Findable, Accessible, Interoperable, Reusable

Examples: MD Setup, Protein-ligand Docking, Protein Flexibility, Molecular Interactions

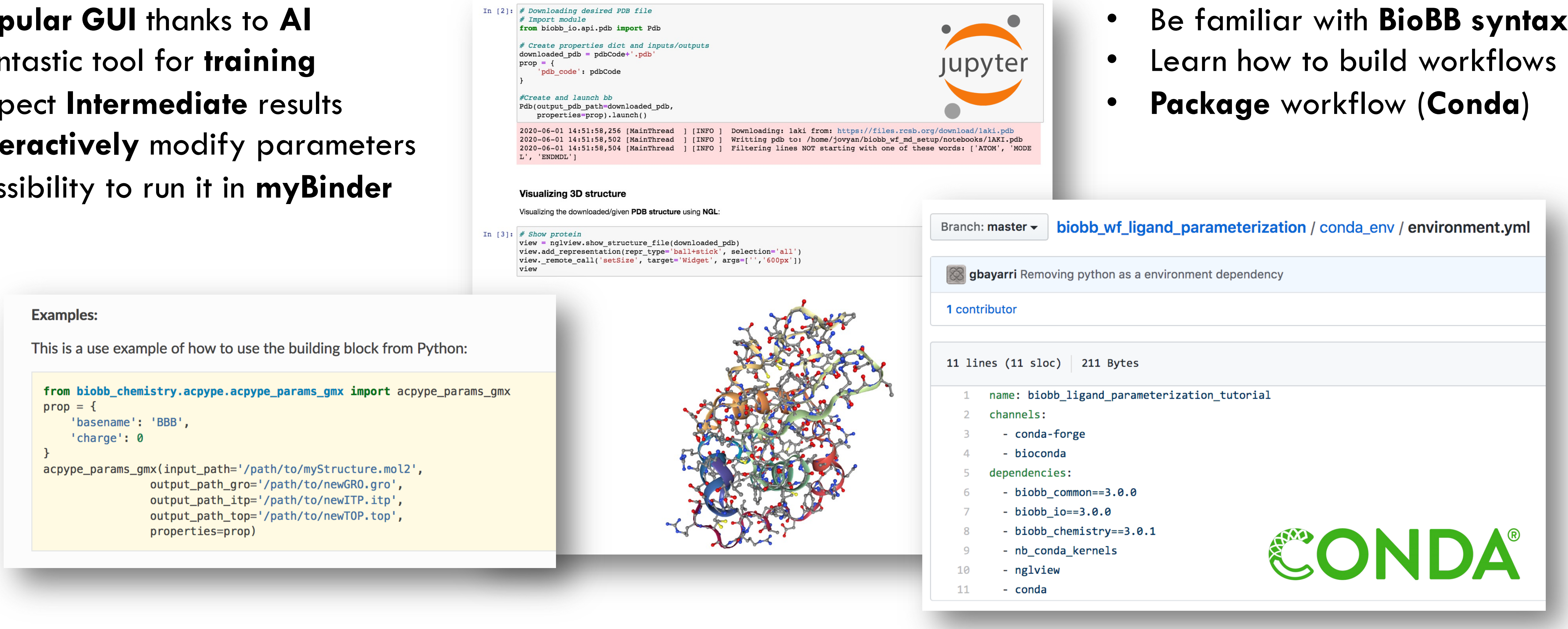
BioBB workflows with **Jupyter Notebooks** and **BioConda**

In general:

- **Popular GUI** thanks to **AI**
- 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**)



```
In [2]: # 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).launch()
```

```
2020-06-01 14:51:58,256 [MainThread  ] [INFO ]  Downloading: 1aki from: https://files.rcsb.org/download/1aki.pdb
2020-06-01 14:51:58,502 [MainThread  ] [INFO ]  Writting pdb to: /home/jovyan/biobb_wf_md_setup/notebooks/1AKI.pdb
2020-06-01 14:51:58,504 [MainThread  ] [INFO ]  Filtering lines NOT starting with one of these words: ['ATOM', 'MODE
L', 'ENDMDL']
```

Visualizing 3D structure

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

```
In [3]: # 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
```

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)
```

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

gbayarri Removing python as a environment dependency

1 contributor

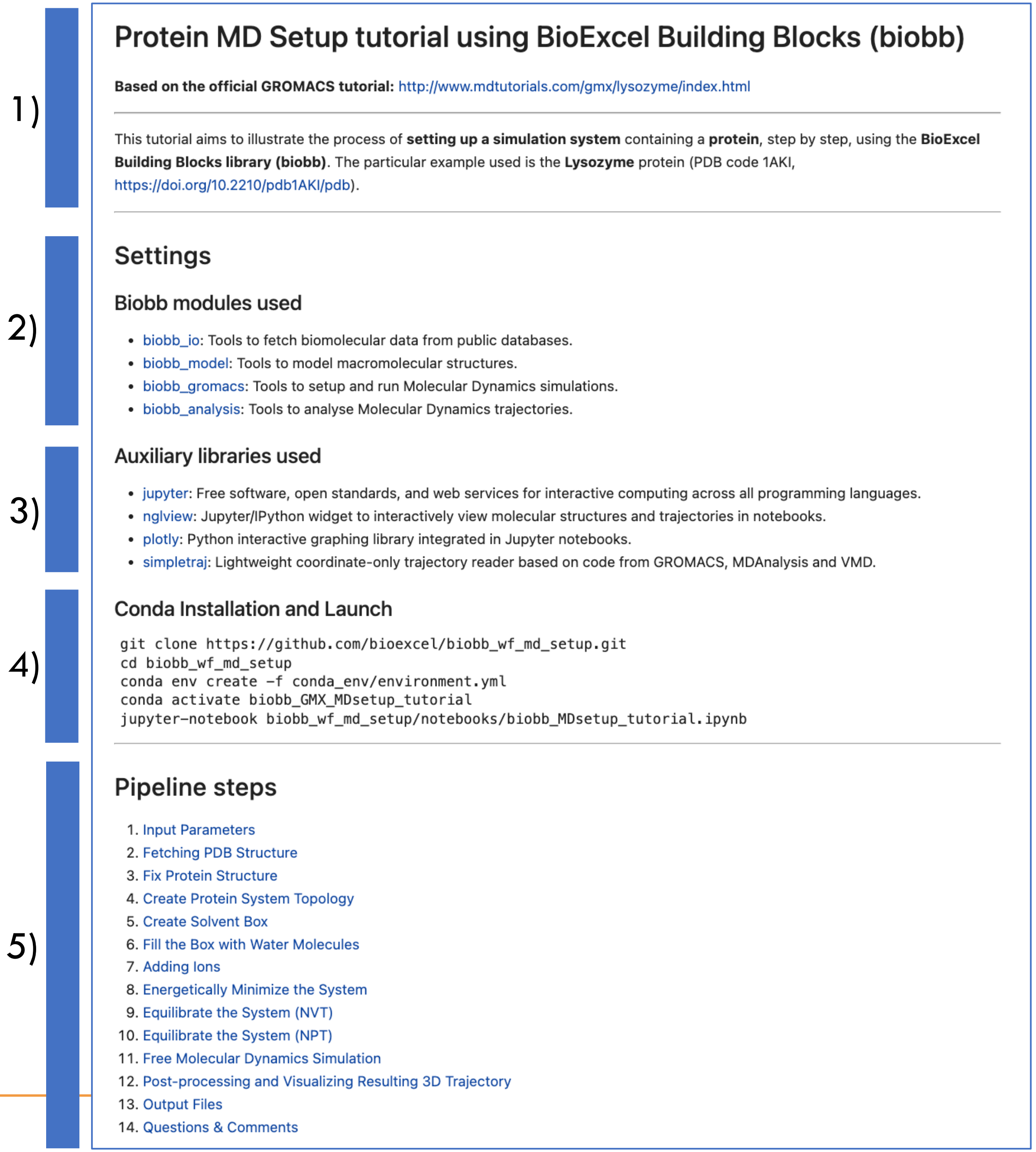
11 lines (11 sloc) 211 Bytes

```
name: biobb_ligand_parameterization_tutorial
channels:
  - conda-forge
  - bioconda
dependencies:
  - biobb_common==3.0.0
  - biobb_io==3.0.0
  - biobb_chemistry==3.0.1
  - nb_conda_kernels
  - nglview
  - conda
```

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**



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

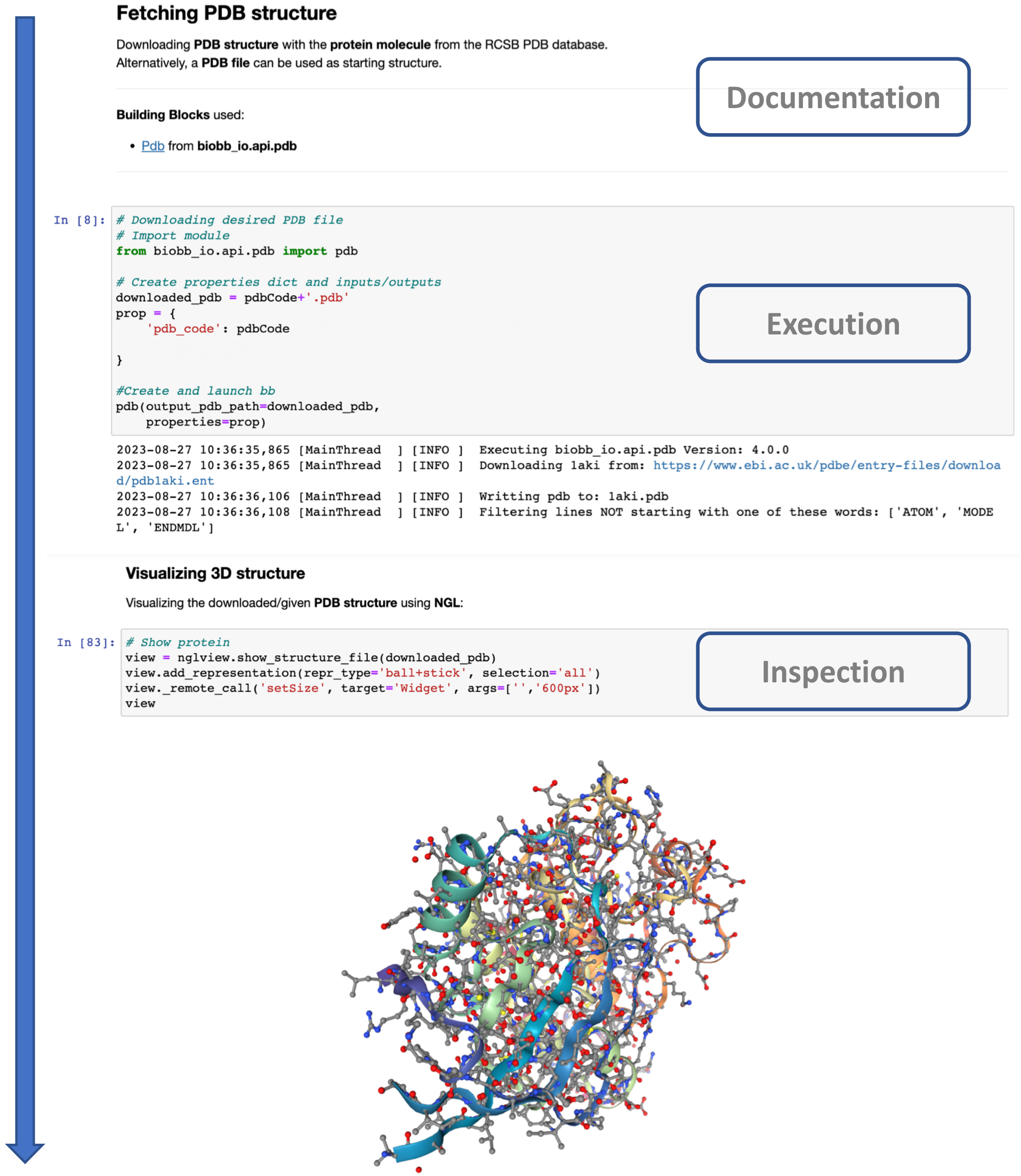
- jupyter: Free software, open standards, and web services for interactive computing across all programming languages.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in notebooks.
- plotly: Python interactive graphing library integrated in Jupyter notebooks.
- 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
conda activate biobb_GMX_MDsetup_tutorial
jupyter-notebook biobb_wf_md_setup/notebooks/biobb_MDsetup_tutorial.ipynb
```

Pipeline steps

1. Input Parameters
2. Fetching PDB Structure
3. Fix Protein Structure
4. Create Protein System Topology
5. Create Solvent Box
6. Fill the Box with Water Molecules
7. Adding Ions
8. Energetically Minimize the System
9. Equilibrate the System (NVT)
10. Equilibrate the System (NPT)
11. Free Molecular Dynamics Simulation
12. Post-processing and Visualizing Resulting 3D Trajectory
13. Output Files
14. Questions & Comments



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**

```
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)
```

```
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/downloa
d/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', 'MODE
L', '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
```

1. Import BioBB module

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

2. Define inputs/outputs and properties

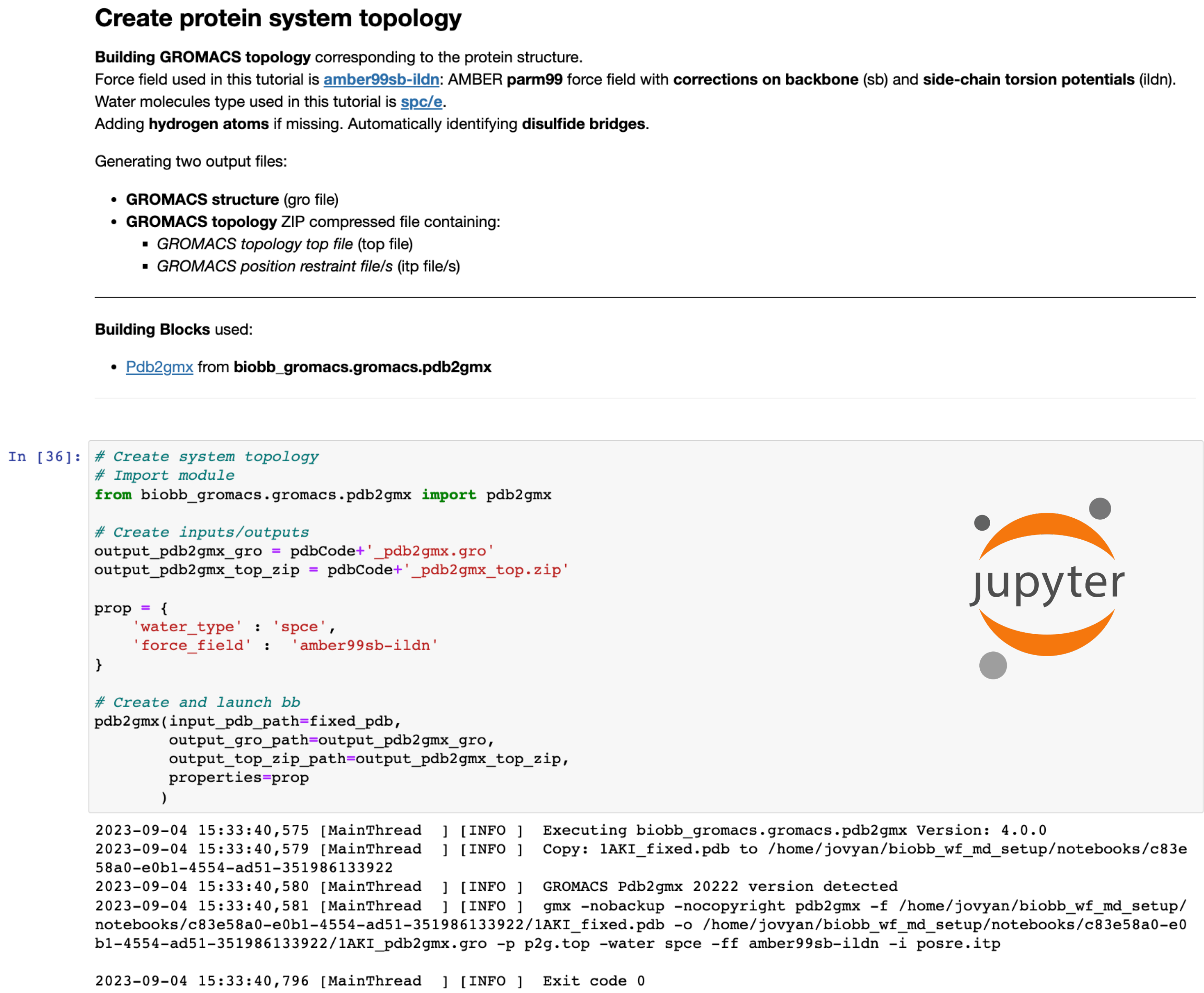
```
# Create properties dict and inputs/outputs
downloaded_pdb = pdbCode+'.pdb'
prop = {
    'pdb_code': pdbCode
}
```

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:

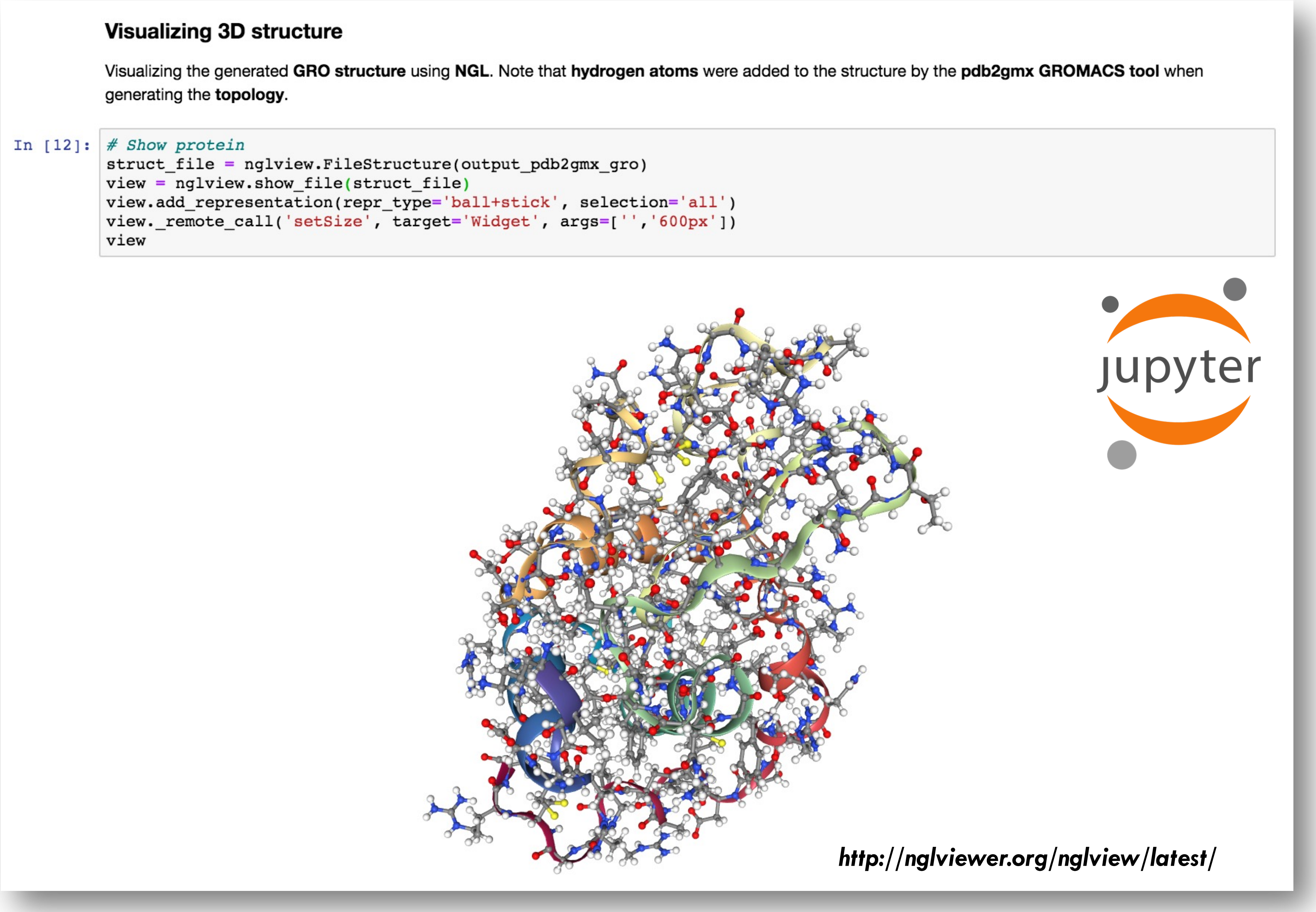
- Pdb2gmx from **biobb_gromacs.gromacs.pdb2gmx**

In [36]:

```python
# 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

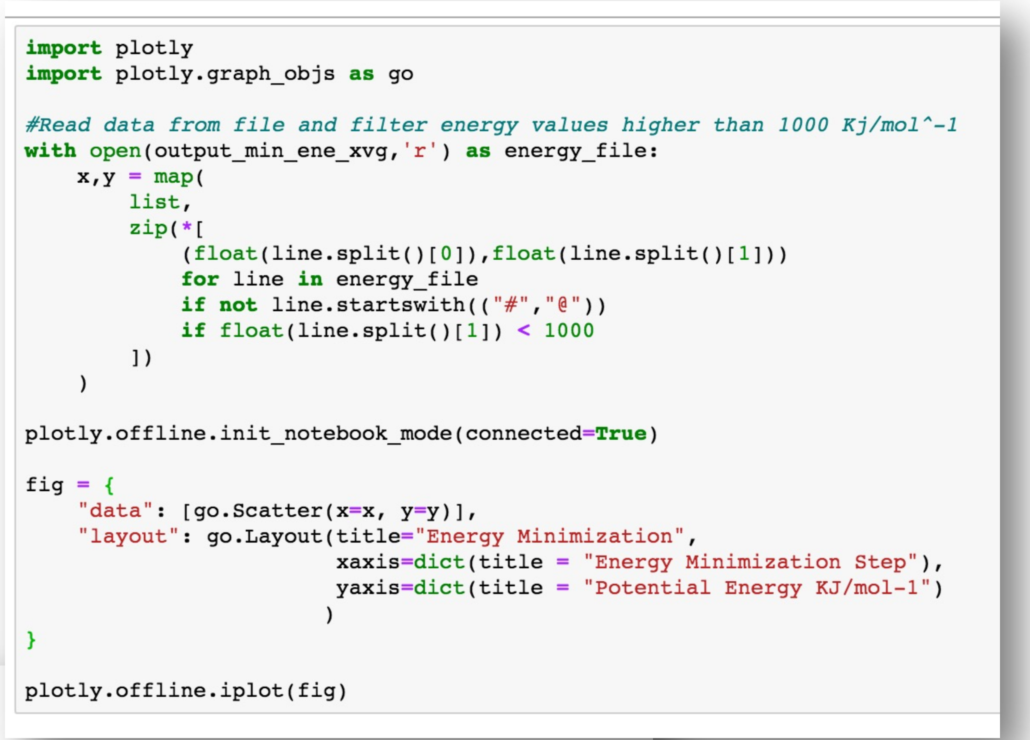
Visualizing the generated **GRO structure** using **NGL**. Note that **hydrogen atoms** were added to the structure by the **pdb2gmx GROMACS tool** when generating the **topology**.

```
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
```

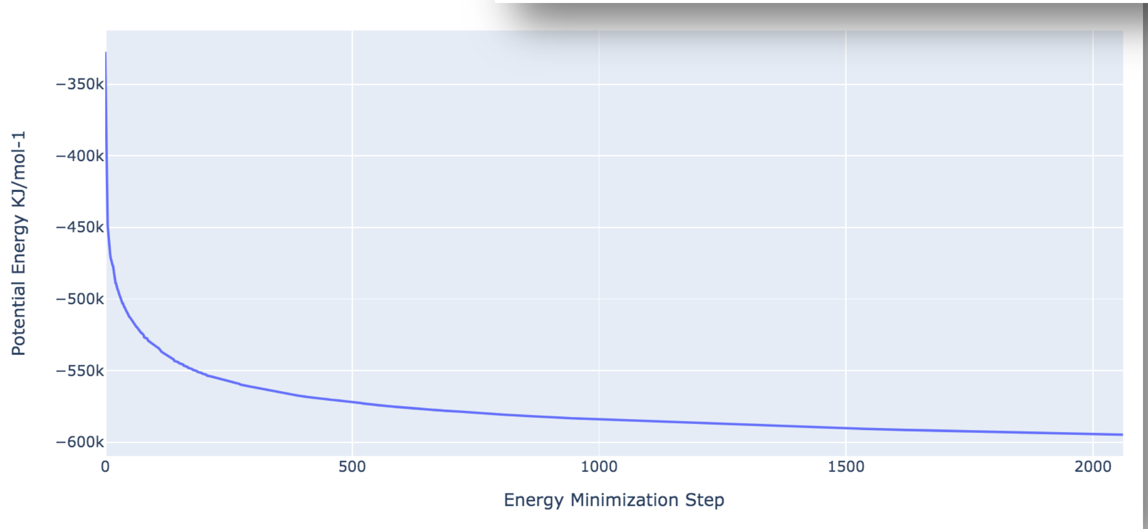
http://nglviewer.org/nglview/latest/

Inspection

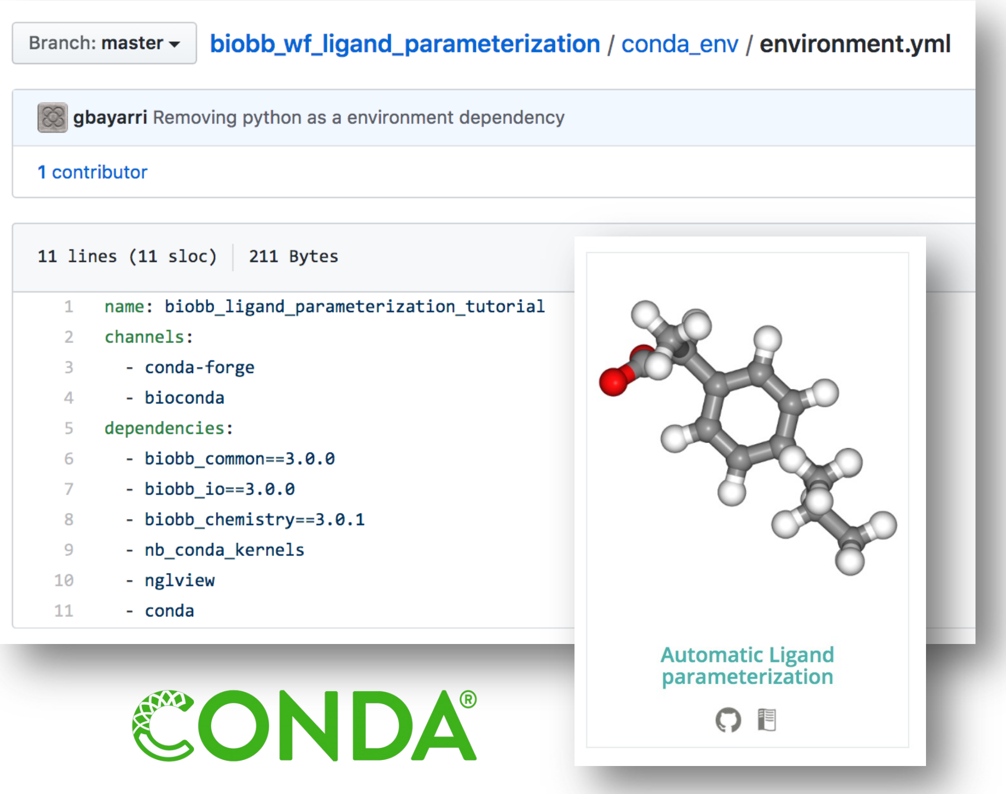
jupyter



```
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

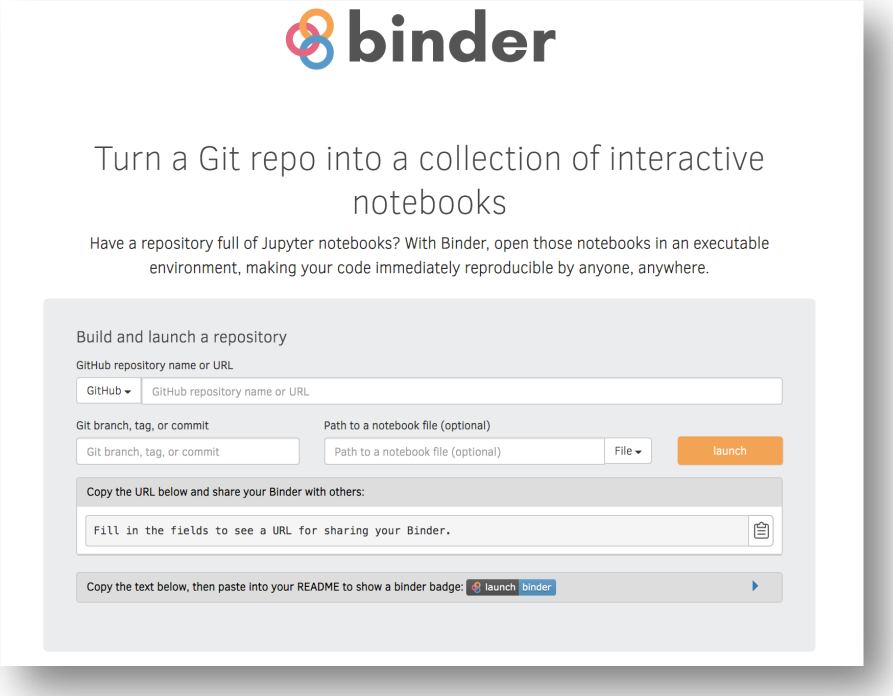
11 lines (11 sloc) | 211 Bytes

```
name: biobb_ligand_parameterization_tutorial
channels:
  - conda-forge
  - bioconda
dependencies:
  - biobb_common==3.0.0
  - biobb_io==3.0.0
  - biobb_chemistry==3.0.1
  - nb_conda_kernels
  - nglview
  - 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

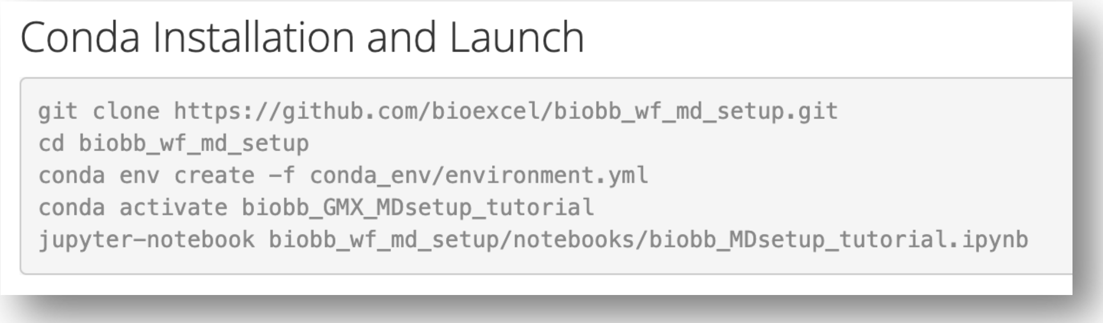
Git branch, tag, or commit: Git branch, tag, or commit

Path to a notebook file (optional): Path to a notebook file (optional) | File ▾ | launch

Copy the URL below and share your Binder with others:

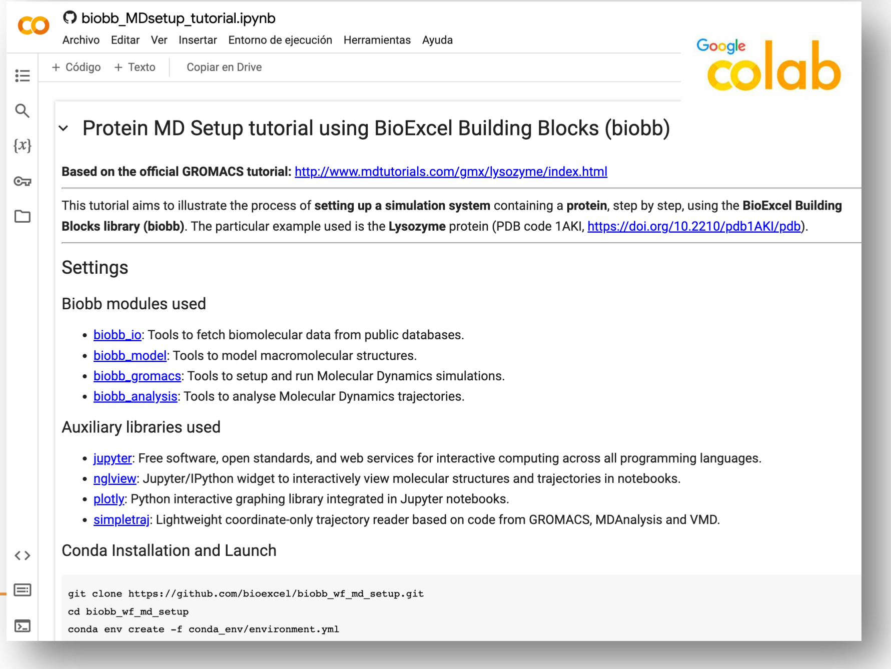
```
Fill in the fields to see a URL for sharing your Binder.
```

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
```



CO 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

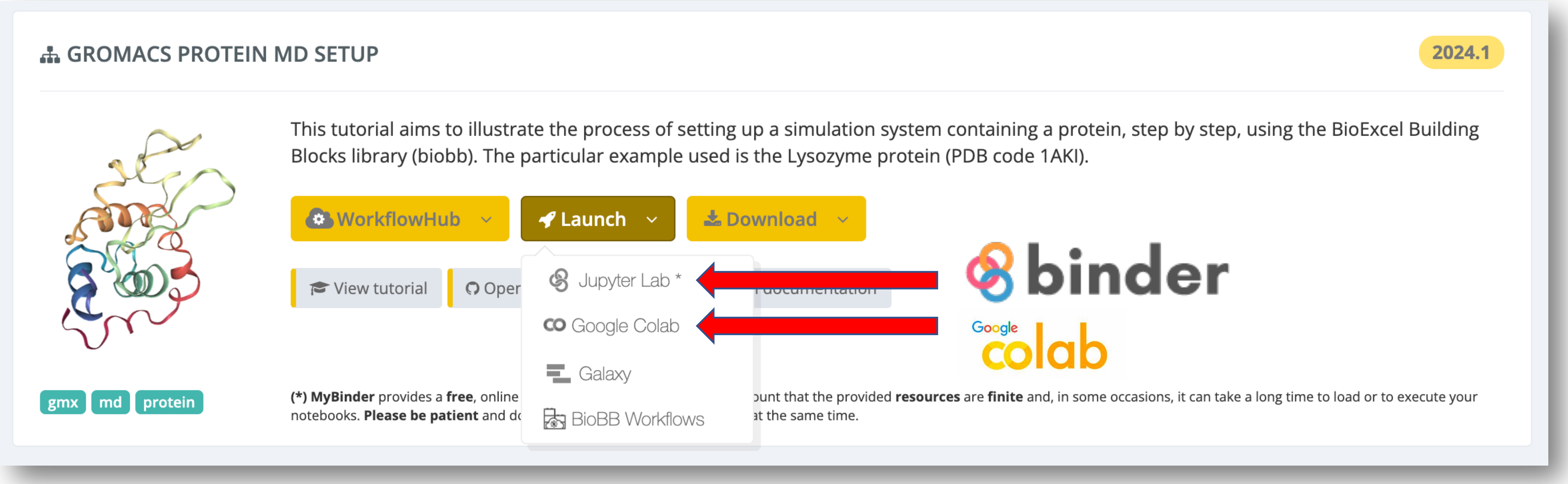
- jupyter: Free software, open standards, and web services for interactive computing across all programming languages.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in notebooks.
- plotly: Python interactive graphing library integrated in Jupyter notebooks.
- 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



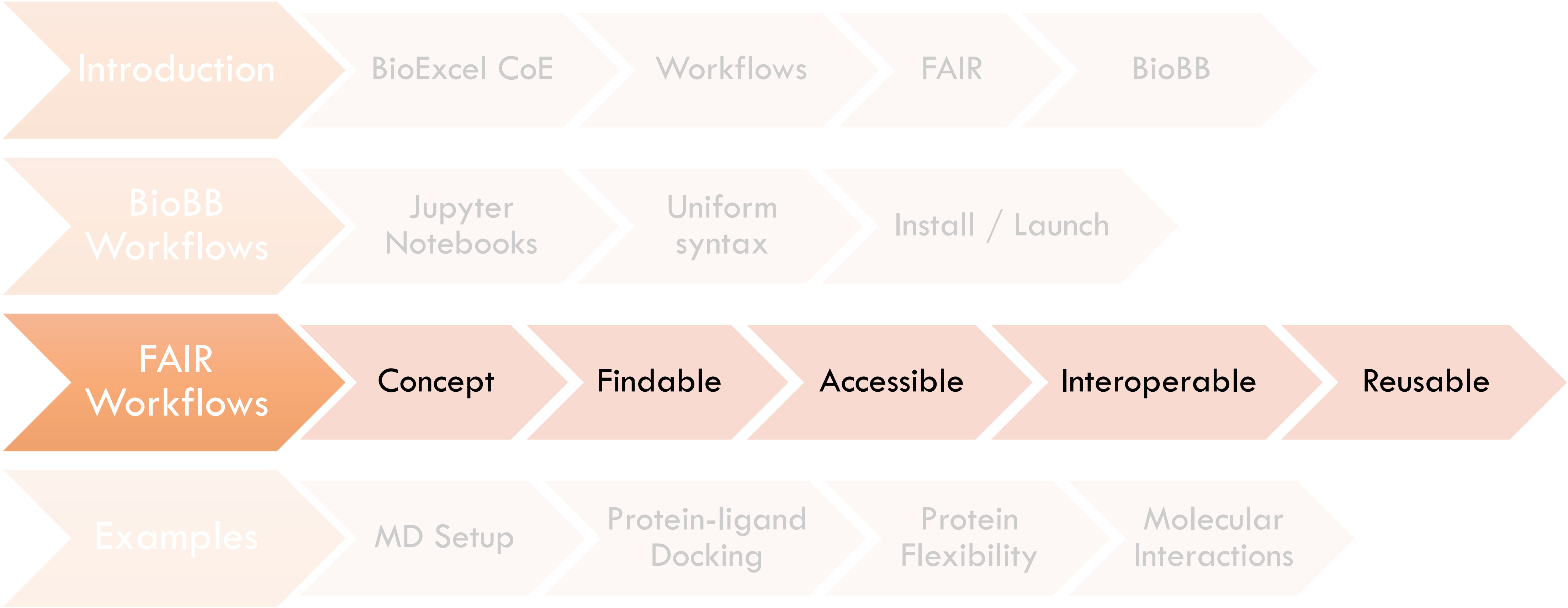
GROMACS PROTEIN MD SETUP

2024.1

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).

(*) **MyBinder** provides a **free**, online ount that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and dc at the same time.

Index

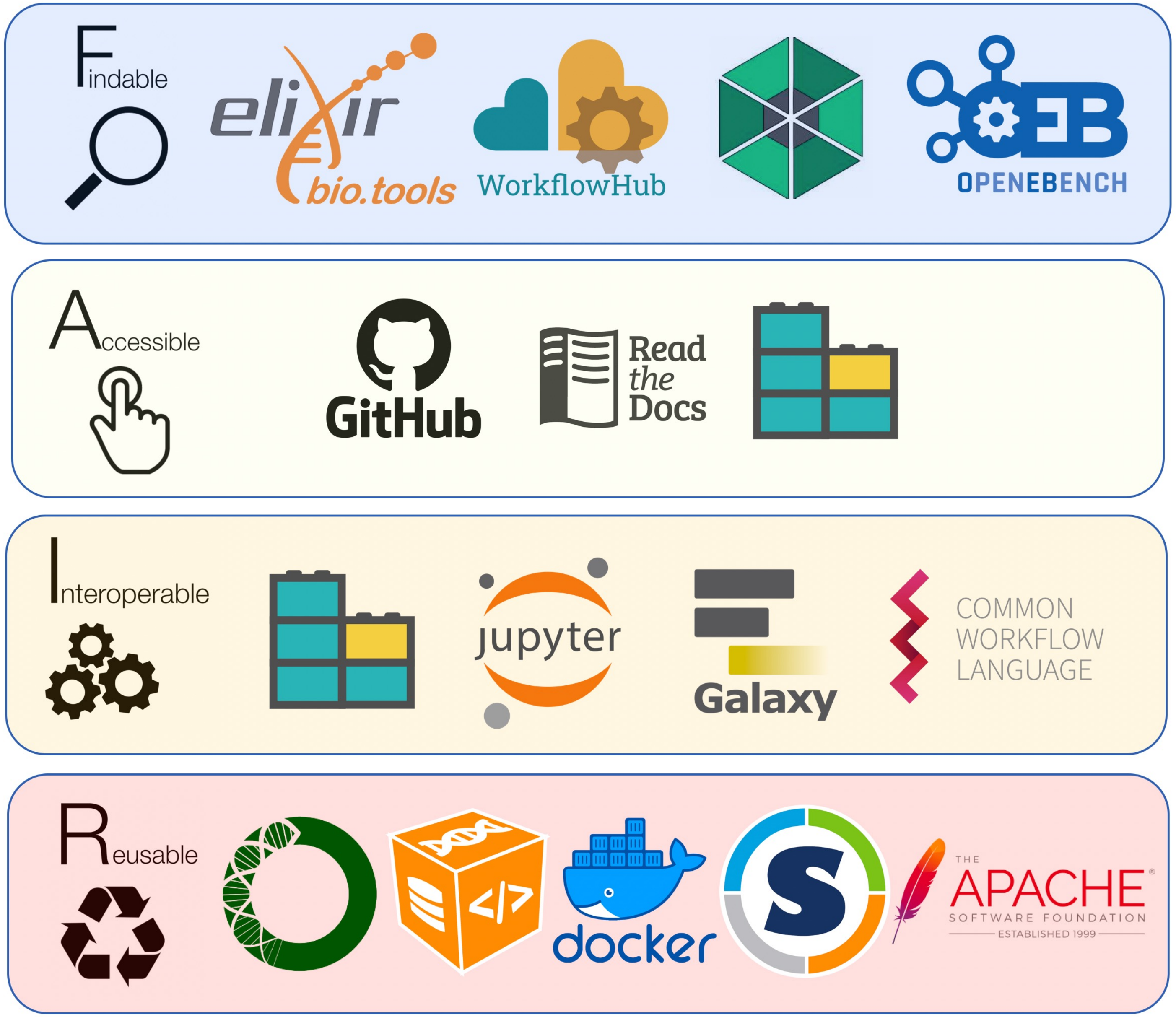


Introduction – BioExcel CoE – Workflows – FAIR – BioBB

BioBB Workflows – Jupyter Notebooks – Uniform syntax – Install / Launch

FAIR Workflows – Concept – Findable – Accessible – Interoperable – Reusable

Examples – MD Setup – Protein-ligand Docking – Protein Flexibility – Molecular Interactions



F: Software, and its associated metadata, is easy for both humans and machines to find.

A: Software, and its metadata, is retrievable via standardised protocols.

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

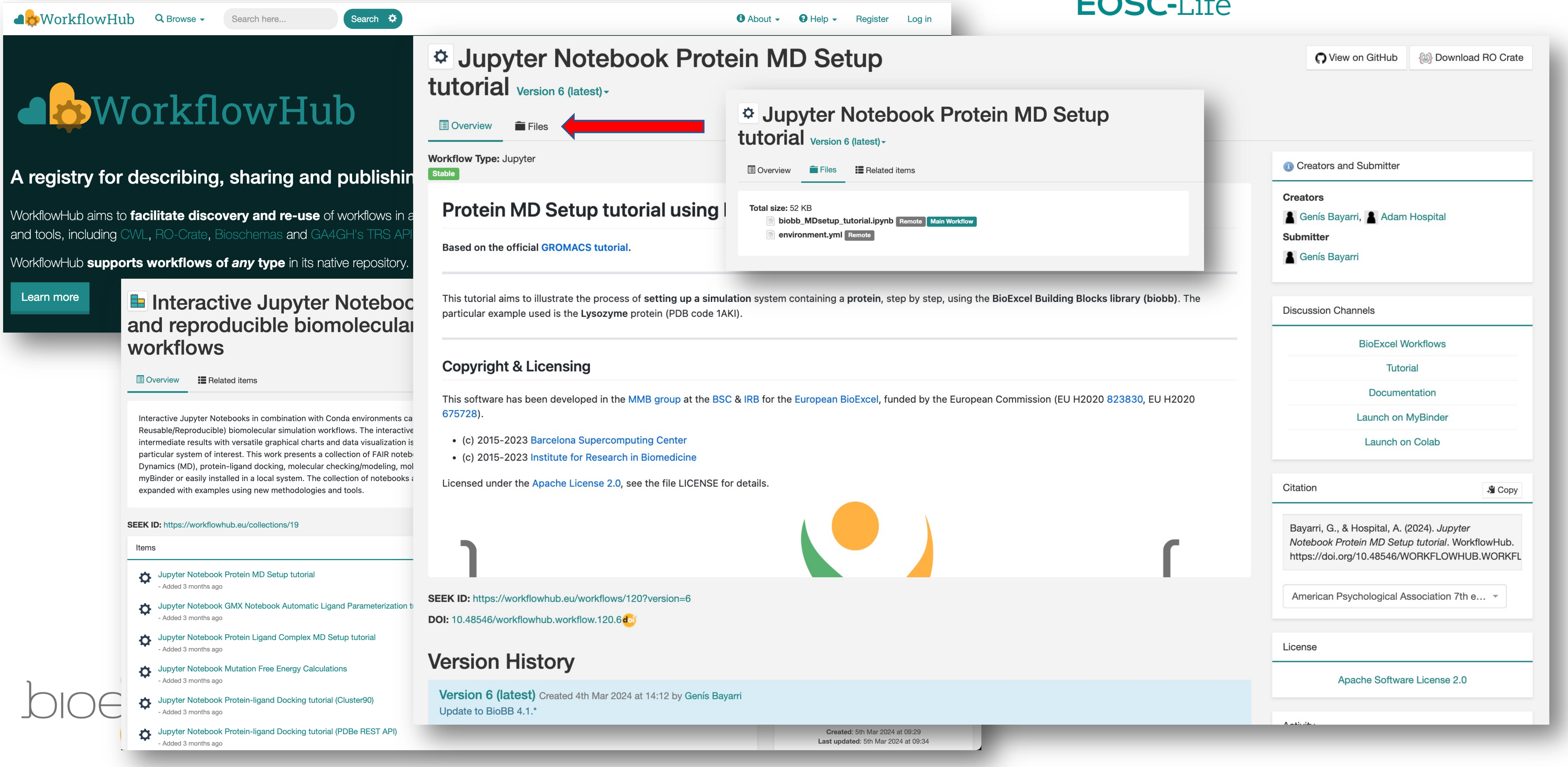
R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).

Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* **9**, 622 (2022).

F: Software, and its associated metadata, is easy for both humans and machines to find.

Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* **9**, 622 (2022).

https://workflowhub.eu/



WorkflowHub

A registry for describing, sharing and publishin

WorkflowHub aims to **facilitate discovery and re-use** of workflows in a
and tools, including CWL, RO-Crate, Bioschemas and GA4GH's TRS API

WorkflowHub **supports workflows of *any* type** in its native repository.

Interactive Jupyter Noteboo and reproducible biomolecular workflows

Interactive Jupyter Notebooks in combination with Conda environments ca
Reusable/Reproducible) biomolecular simulation workflows. The interactive
intermediate results with versatile graphical charts and data visualization is
particular system of interest. This work presents a collection of FAIR noteb
Dynamics (MD), protein-ligand docking, molecular checking/modeling, mol
myBinder or easily installed in a local system. The collection of notebooks a
expanded with examples using new methodologies and tools.

SEEK ID: https://workflowhub.eu/collections/19

Items

- Jupyter Notebook Protein MD Setup tutorial - Added 3 months ago
- Jupyter Notebook GMX Notebook Automatic Ligand Parameterization t - Added 3 months ago
- Jupyter Notebook Protein Ligand Complex MD Setup tutorial - Added 3 months ago
- Jupyter Notebook Mutation Free Energy Calculations - Added 3 months ago
- Jupyter Notebook Protein-ligand Docking tutorial (Cluster90) - Added 3 months ago
- Jupyter Notebook Protein-ligand Docking tutorial (PDBe REST API) - Added 3 months ago

Created: 5th Mar 2024 at 09:29
Last updated: 5th Mar 2024 at 09:34

Jupyter Notebook Protein MD Setup tutorial Version 6 (latest)

Workflow Type: Jupyter
Stable

Protein MD Setup tutorial using

Based on the official GROMACS tutorial.

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).

Copyright & Licensing

This software has been developed in the MMB group at the BSC & IRB for the European BioExcel, funded by the European Commission (EU H2020 823830, EU H2020 675728).

- (c) 2015-2023 Barcelona Supercomputing Center
- (c) 2015-2023 Institute for Research in Biomedicine

Licensed under the Apache License 2.0, see the file LICENSE for details.

SEEK ID: https://workflowhub.eu/workflows/120?version=6

DOI: 10.48546/workflowhub.workflow.120.6

Version History

Version 6 (latest) Created 4th Mar 2024 at 14:12 by Genís Bayarri
Update to BioBB 4.1.*

Jupyter Notebook Protein MD Setup tutorial Version 6 (latest)

Total size: 52 KB
- biobb_MDsetup_tutorial.ipynb Remote Main Workflow
- environment.yml Remote

Creators and Submitter

Creators
Genís Bayarri, Adam Hospital

Submitter
Genís Bayarri

Discussion Channels

- BioExcel Workflows
- Tutorial
- Documentation
- Launch on MyBinder
- Launch on Colab

Citation

Bayarri, G., & Hospital, A. (2024). *Jupyter Notebook Protein MD Setup tutorial*. WorkflowHub. https://doi.org/10.48546/WORKFLOWHUB.WORKFL

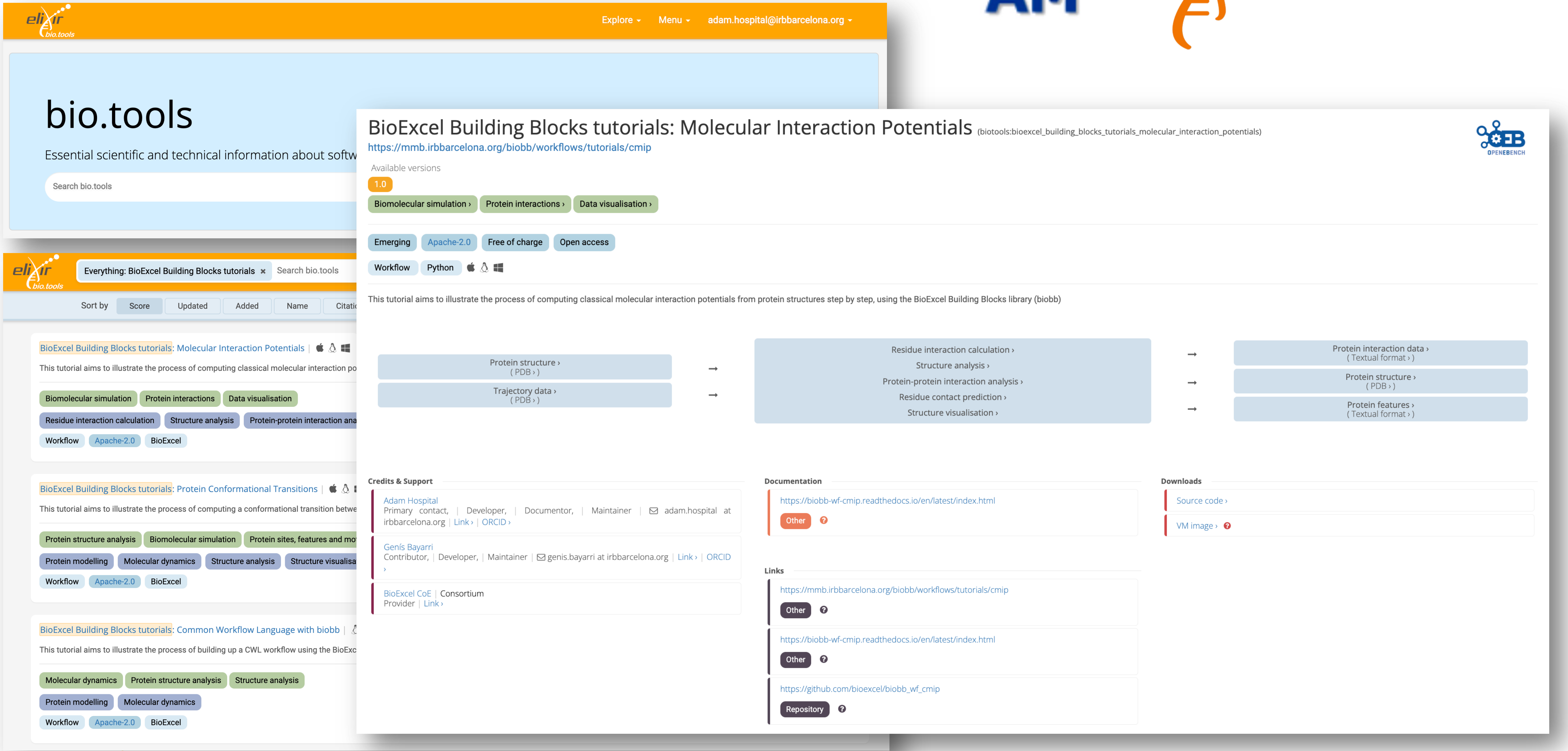
American Psychological Association 7th e...

License

Apache Software License 2.0

Findable

https://bio.tools/



Explore ▾ Menu ▾ adam.hospital@irbbarcelona.org ▾

bio.tools

Essential scientific and technical information about softw

Search bio.tools

Everything: BioExcel Building Blocks tutorials × Search bio.tools

Sort by Score Updated Added Name Citati

BioExcel Building Blocks tutorials: Molecular Interaction Potentials |

This tutorial aims to illustrate the process of computing classical molecular interaction po

Biomolecular simulation | Protein interactions | Data visualisation

Residue interaction calculation | Structure analysis | Protein-protein interaction ana

Workflow | Apache-2.0 | BioExcel

BioExcel Building Blocks tutorials: Protein Conformational Transitions |

This tutorial aims to illustrate the process of computing a conformational transition betwe

Protein structure analysis | Biomolecular simulation | Protein sites, features and mo

Protein modelling | Molecular dynamics | Structure analysis | Structure visualisa

Workflow | Apache-2.0 | BioExcel

BioExcel Building Blocks tutorials: Common Workflow Language with biobb |

This tutorial aims to illustrate the process of building up a CWL workflow using the BioExc

Molecular dynamics | Protein structure analysis | Structure analysis

Protein modelling | Molecular dynamics

Workflow | Apache-2.0 | BioExcel

BioExcel Building Blocks tutorials: Molecular Interaction Potentials (biotools:bioexcel_building_blocks_tutorials_molecular_interaction_potentials)

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

Available versions

1.0

Biomolecular simulation › | Protein interactions › | Data visualisation ›

Emerging | Apache-2.0 | Free of charge | Open access

Workflow | Python

This tutorial aims to illustrate the process of computing classical molecular interaction potentials from protein structures step by step, using the BioExcel Building Blocks library (biobb)

Protein structure › (PDB ›) →
Trajectory data › (PDB ›) →

Residue interaction calculation ›
Structure analysis ›
Protein-protein interaction analysis ›
Residue contact prediction ›
Structure visualisation ›

→ Protein interaction data › (Textual format ›)
→ Protein structure › (PDB ›)
→ Protein features › (Textual format ›)

Credits & Support

Adam Hospital
Primary contact, | Developer, | Documentor, | Maintainer | ✉ adam.hospital at irbbarcelona.org | Link › | ORCID ›

Genís Bayarri
Contributor, | Developer, | Maintainer | ✉ genis.bayarri at irbbarcelona.org | Link › | ORCID ›

BioExcel CoE | Consortium
Provider | Link ›

Documentation

https://biobb-wf-cmip.readthedocs.io/en/latest/index.html
Other

Links

https://mmb.irbbarcelona.org/biobb/workflows/tutorials/cmip
Other

https://biobb-wf-cmip.readthedocs.io/en/latest/index.html
Other

https://github.com/bioexcel/biobb_wf_cmip
Repository

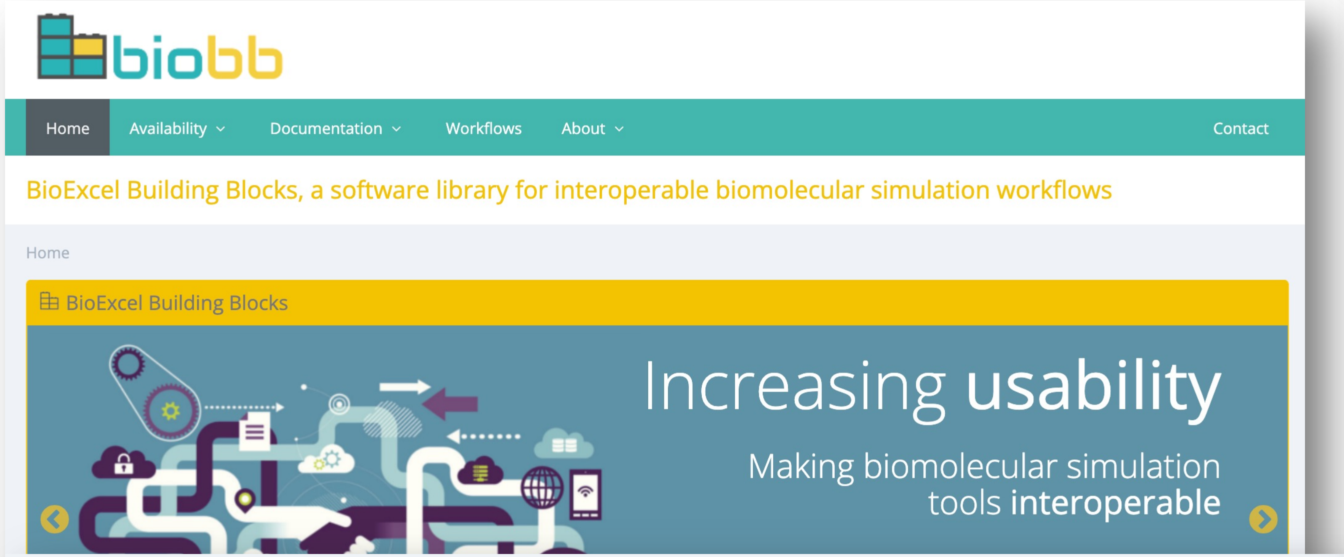
Downloads

Source code ›

VM image ›

Bioschemas

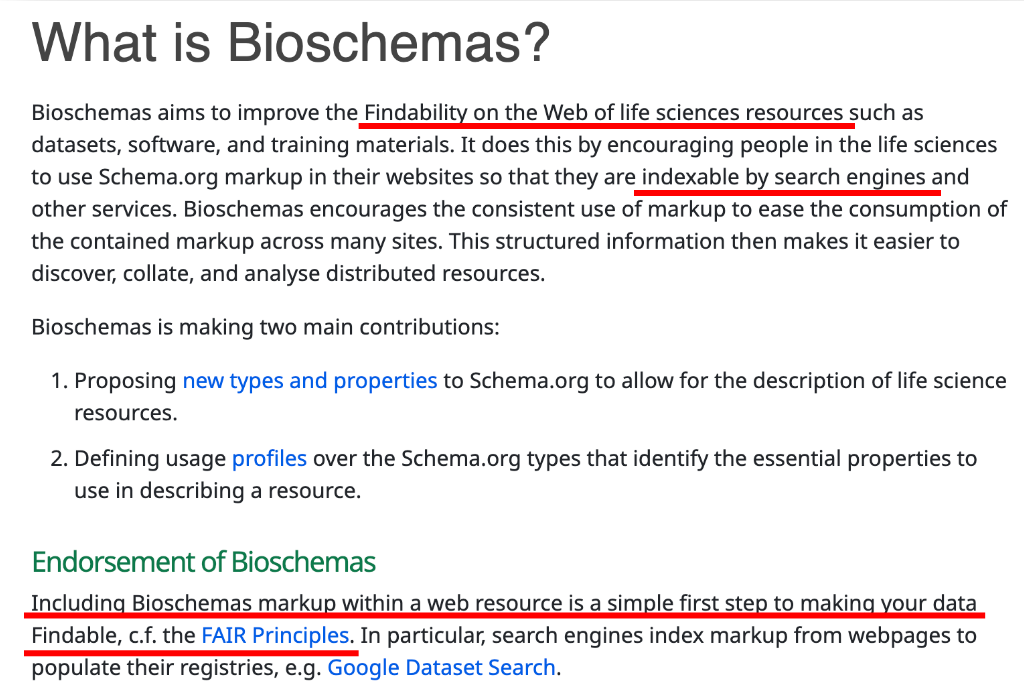
https://mmb.irbbarcelona.org/biobb/



BioExcel Building Blocks, a software library for interoperable biomolecular simulation workflows

Increasing usability

Making biomolecular simulation tools interoperable



What is Bioschemas?

Bioschemas aims to improve the Findability on the Web of life sciences resources such as datasets, software, and training materials. It does this by encouraging people in the life sciences to use Schema.org markup in their websites so that they are indexable by search engines and other services. Bioschemas encourages the consistent use of markup to ease the consumption of the contained markup across many sites. This structured information then makes it easier to discover, collate, and analyse distributed resources.

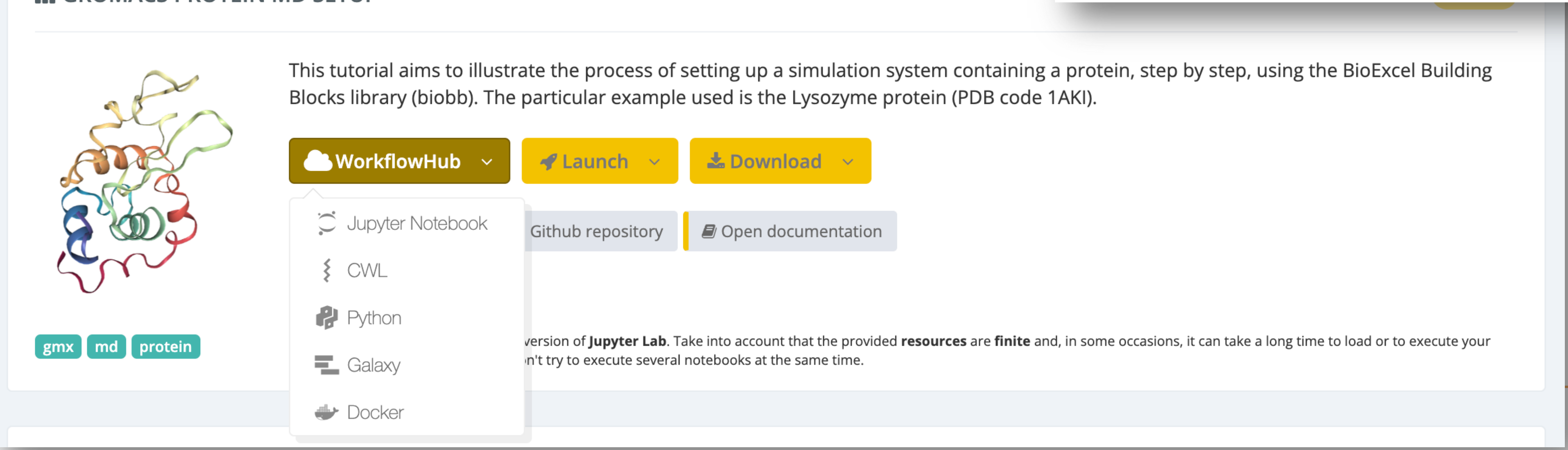
Bioschemas is making two main contributions:

1. Proposing new types and properties to Schema.org to allow for the description of life science resources.
2. Defining usage profiles over the Schema.org types that identify the essential properties to use in describing a resource.

Endorsement of Bioschemas

Including Bioschemas markup within a web resource is a simple first step to making your data Findable, c.f. the FAIR Principles. In particular, search engines index markup from webpages to populate their registries, e.g. Google Dataset Search.

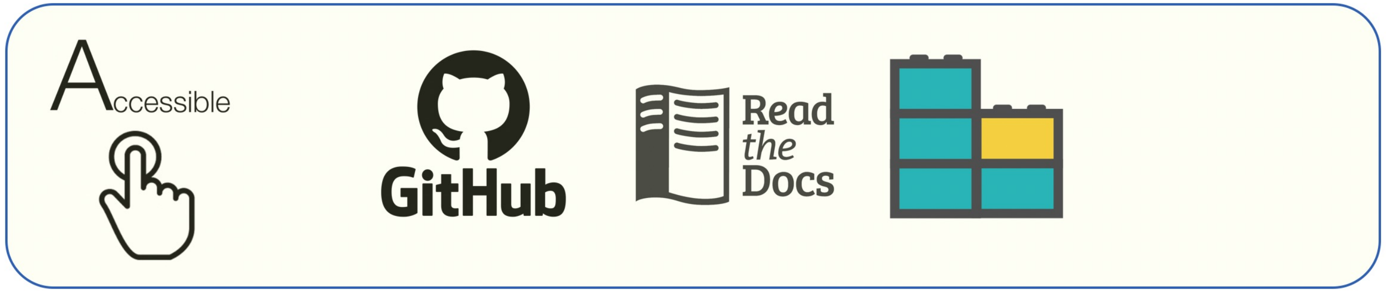
GROMACS PROTEIN MD SETUP



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).

version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your

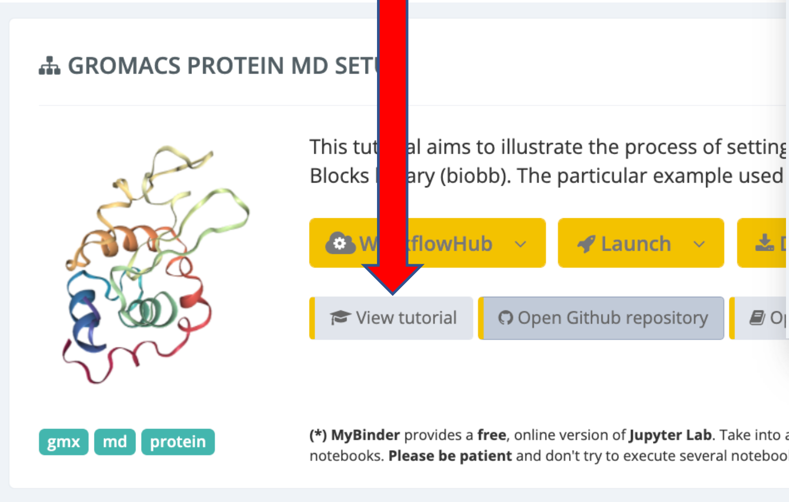
n't try to execute several notebooks at the same time.



A: Software, and its metadata, is retrievable via standardised protocols.

Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* **9**, 622 (2022).

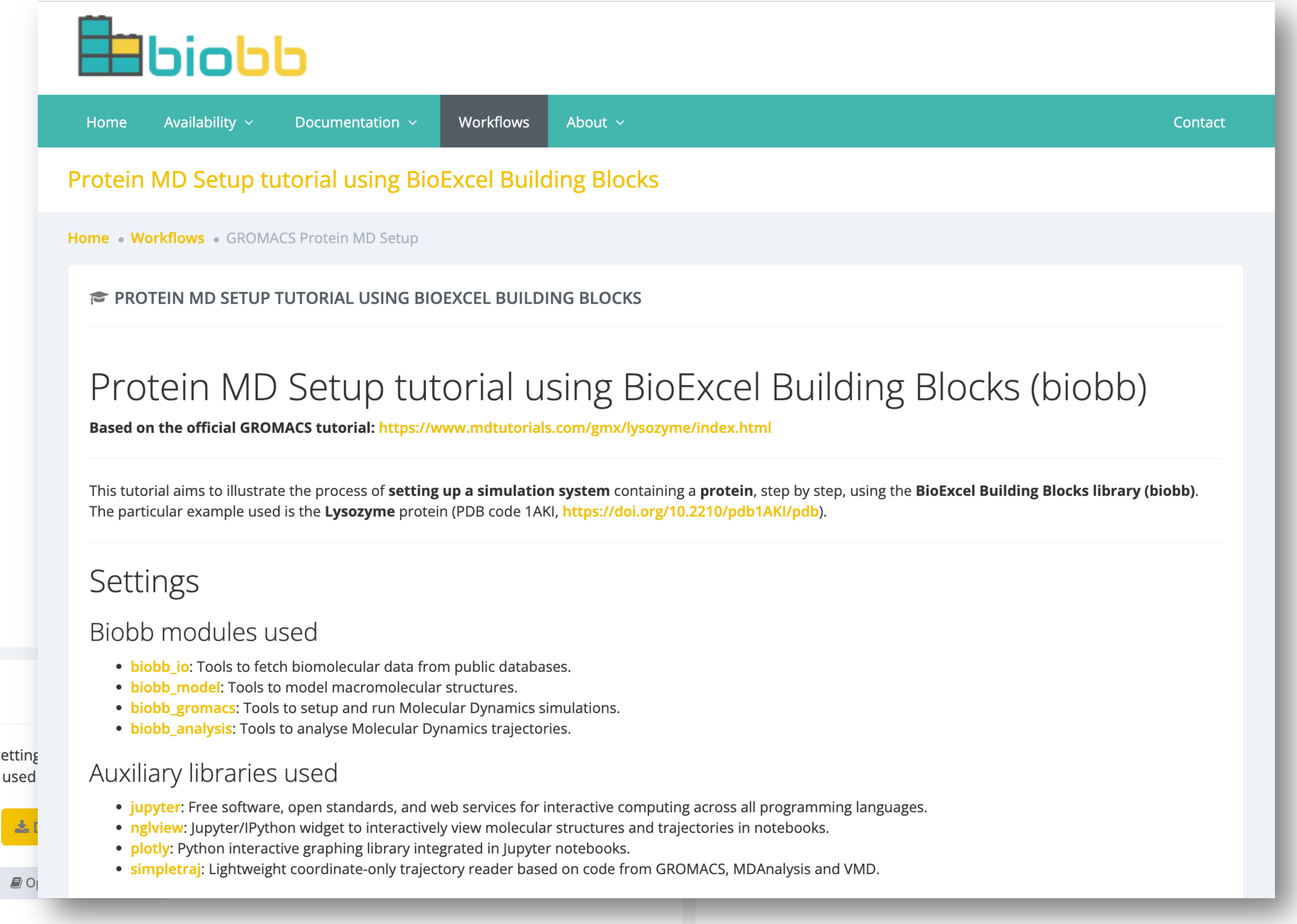
Accessible GitHub Read the Docs



GROMACS PROTEIN MD SET

This tut al aims to illustrate the process of setting Blocks ary (biobb). The particular example used

(*) MyBinder provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.



Protein MD Setup tutorial using BioExcel Building Blocks

Home • Workflows • GROMACS Protein MD Setup

PROTEIN MD SETUP TUTORIAL USING BIOEXCEL BUILDING BLOCKS

Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Based on the official GROMACS tutorial: https://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).

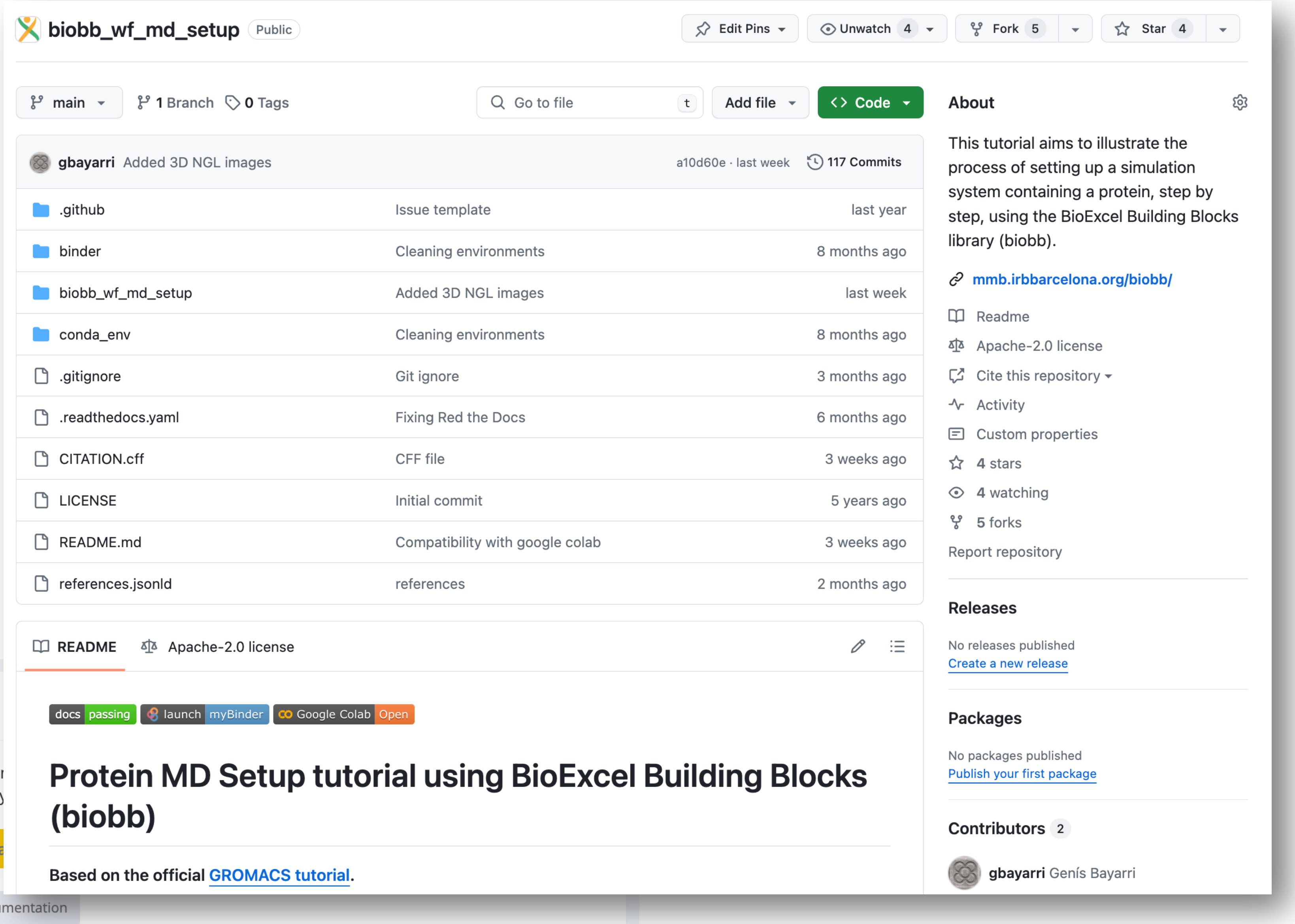
Settings

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.



biobb_wf_md_setup Public

gbayarri Added 3D NGL images a10d60e · last week 117 Commits

.github	Issue template	last year
binder	Cleaning environments	8 months ago
biobb_wf_md_setup	Added 3D NGL images	last week
conda_env	Cleaning environments	8 months ago
.gitignore	Git ignore	3 months ago
.readthedocs.yaml	Fixing Red the Docs	6 months ago
CITATION.cff	CFF file	3 weeks ago
LICENSE	Initial commit	5 years ago
README.md	Compatibility with google colab	3 weeks ago
references.jsonld	references	2 months ago

Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Based on the official GROMACS tutorial.

About

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).

mmb.irbbarcelona.org/biobb/

Releases

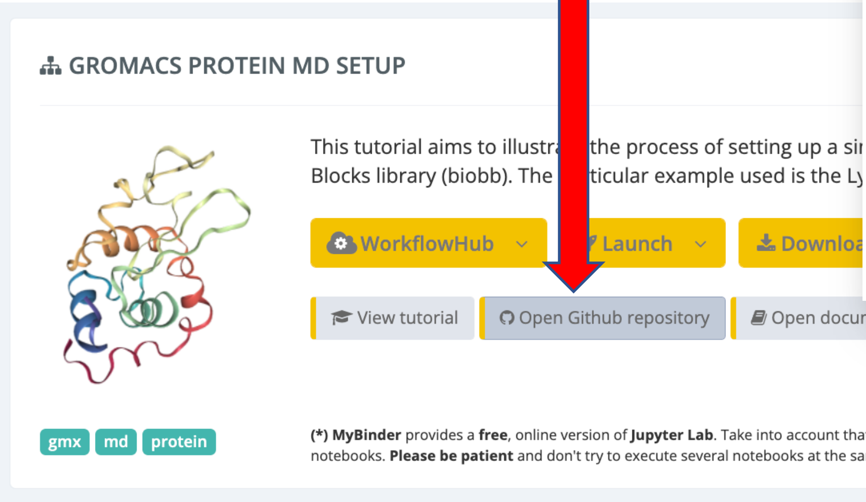
No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

gbayarri Genís Bayarri

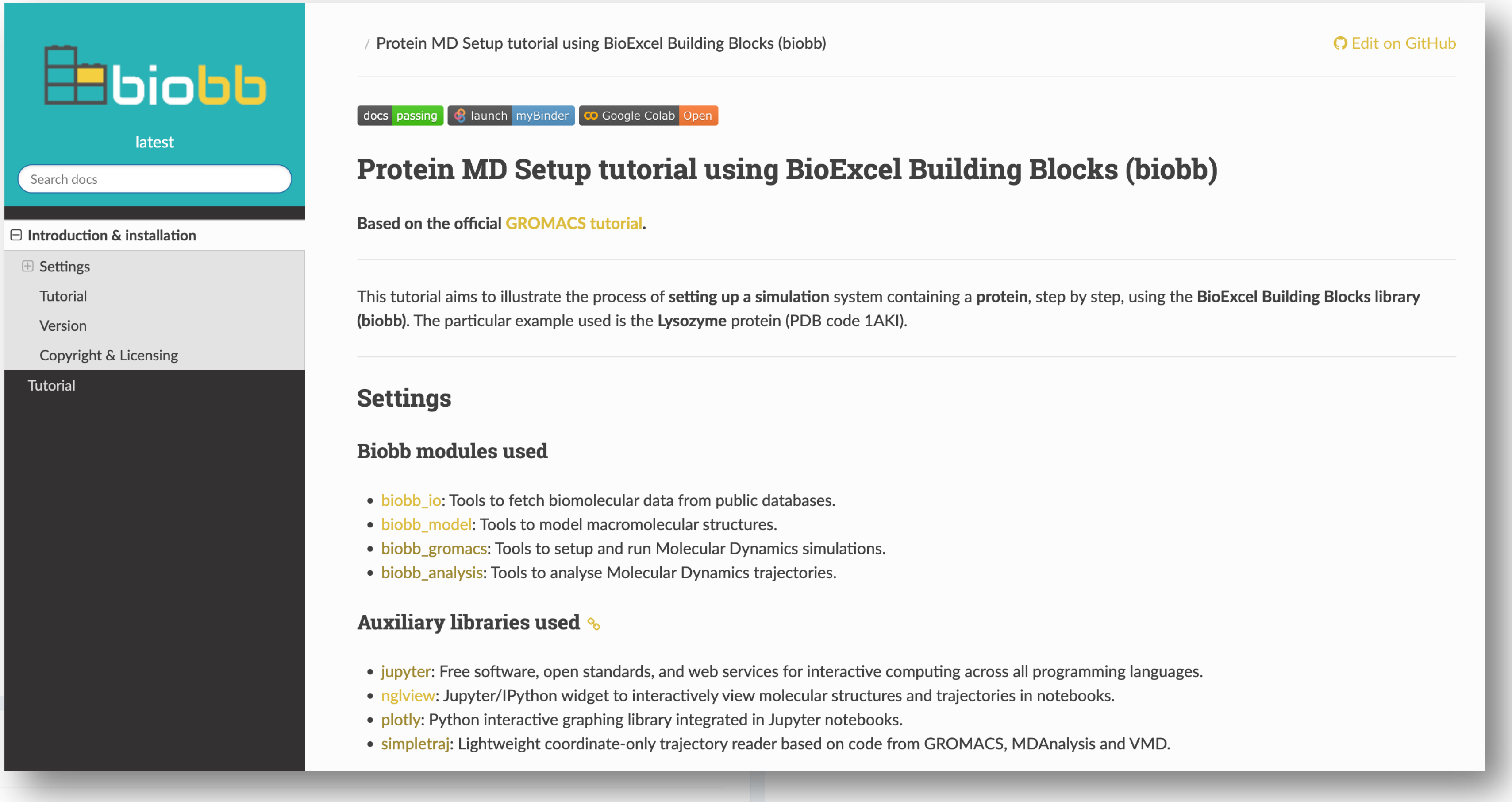


GROMACS PROTEIN MD SETUP

This tutorial aims to illustra the process of setting up a si
Blocks library (biobb). The ticular example used is the Ly

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided resources are finite and, in some occasions, it can take a long time to load or to execute your notebooks. Please be patient and don't try to execute several notebooks at the same time.

Accessible GitHub Read the Docs



/ Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Based on the official GROMACS tutorial.

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).

Settings

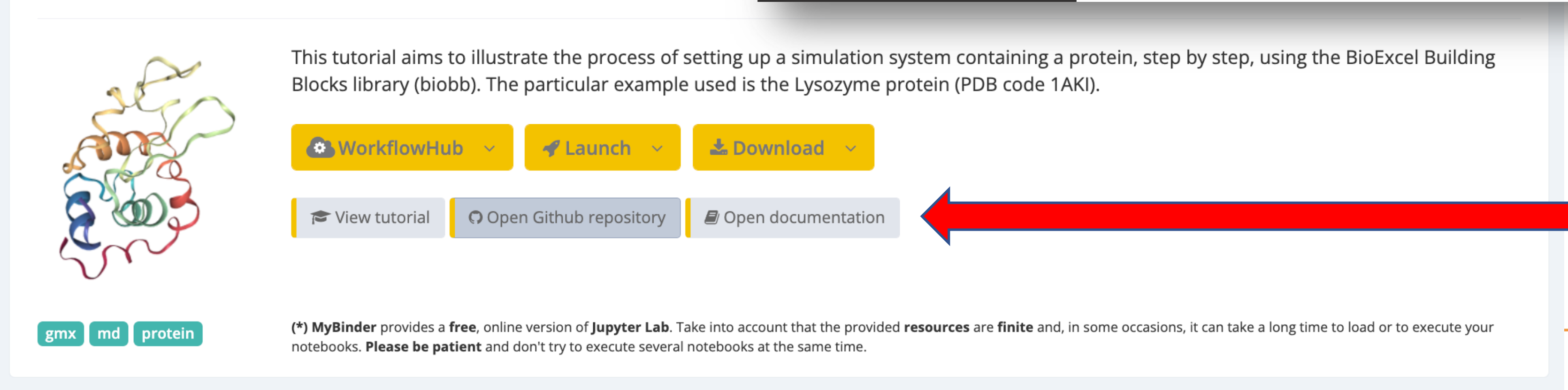
Biobb modules used

- 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.

GROMACS PROTEIN MD SETUP



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).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

Read the Docs

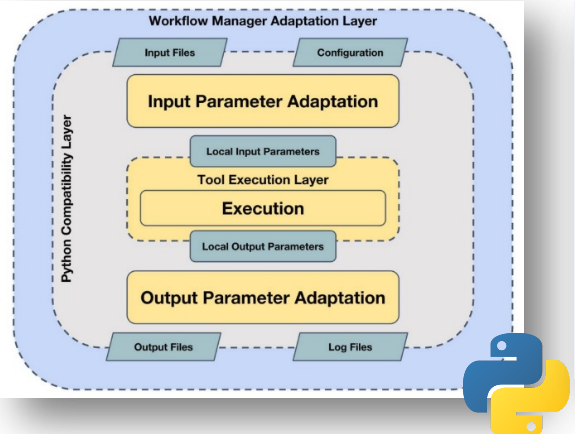
I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

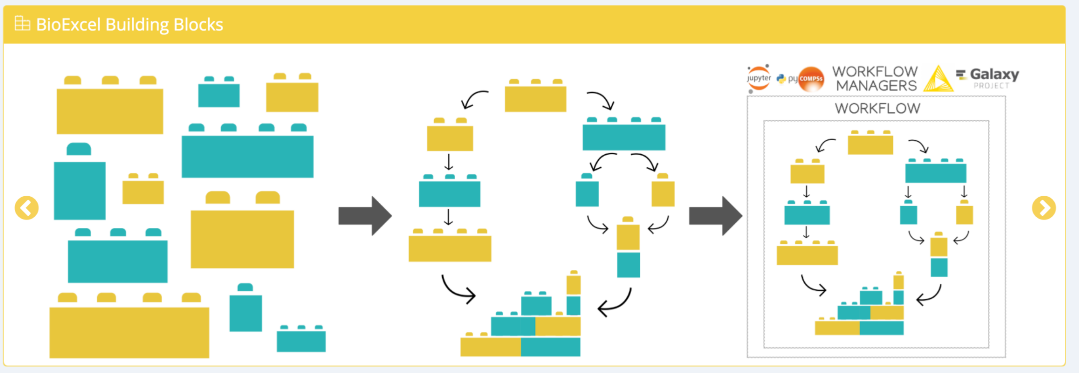
Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* **9**, 622 (2022).

Interoperable

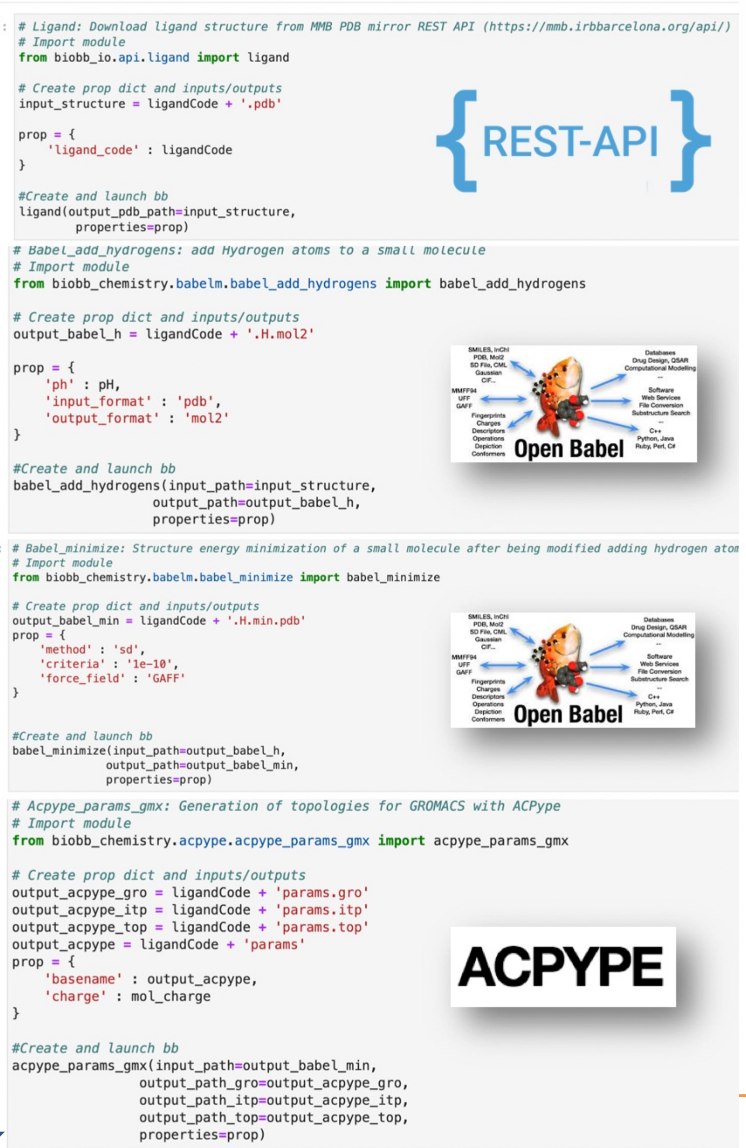
Galaxy

COMMON WORKFLOW LANGUAGE





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



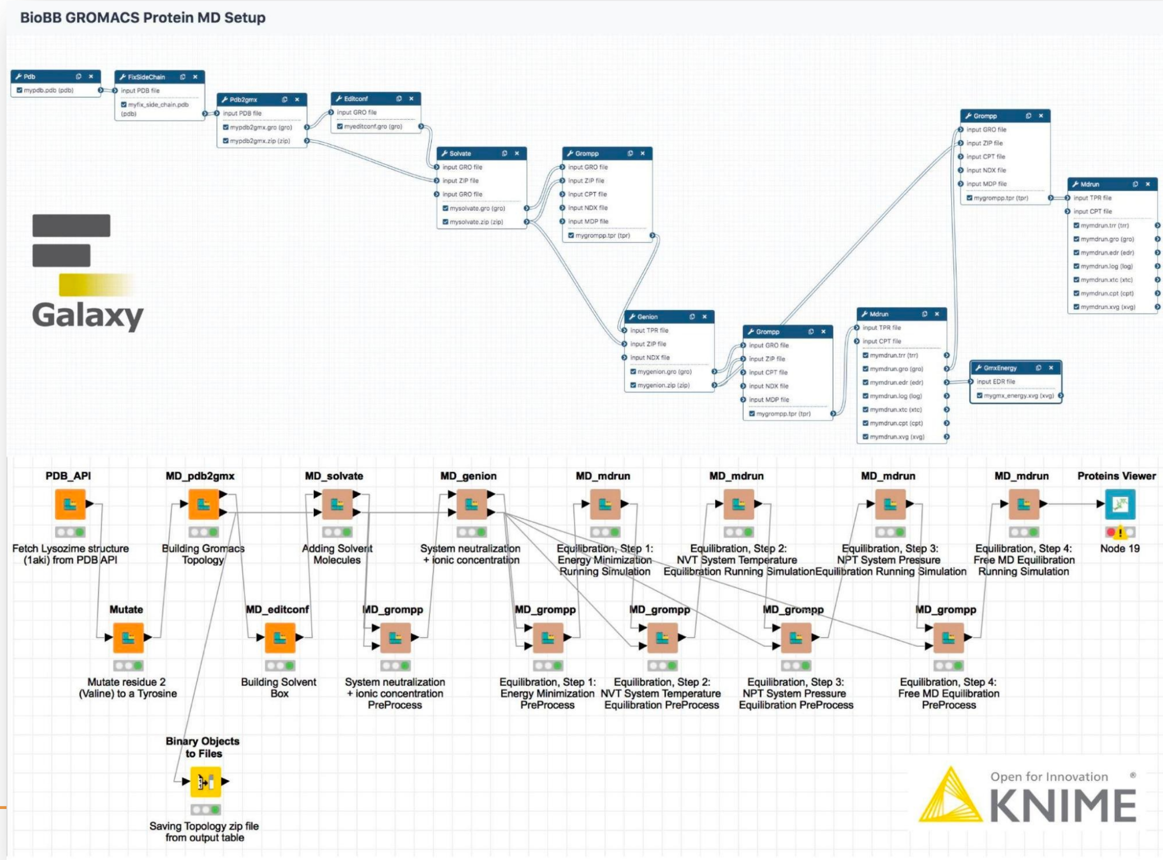
```
# Ligand: Download ligand structure from MMB PDB mirror REST API (https://mmb.irbbarcelona.org/api/)
# Import module
from biobb_io.api.ligand import ligand

# 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)
```

```
# 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)
```

```
# Babel_minimize: Structure energy minimization of a small molecule after being modified adding hydrogen atom
# 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)
```

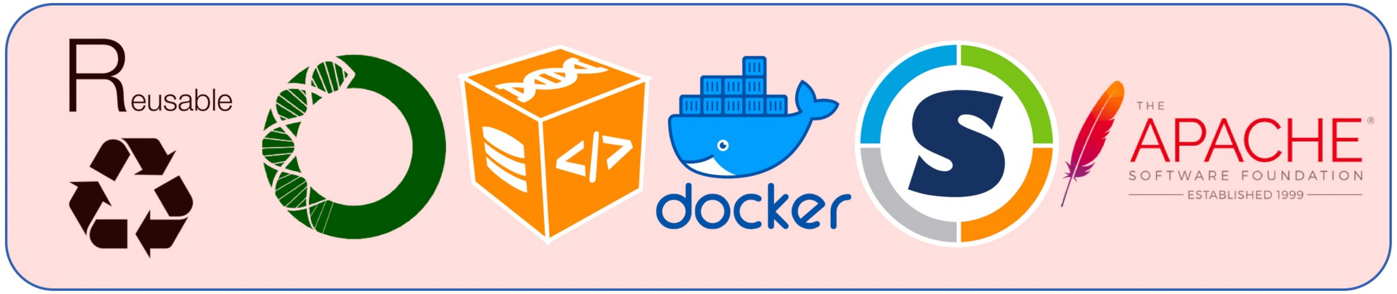
```
# Acpype_params_gmx: Generation of topologies for GROMACS with ACPype
# Import module
from biobb_chemistry.acpype.acpype_params_gmx import acpype_params_gmx

# 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)
```

jupyter



BioBB GROMACS Protein MD Setup

Pdb
FixSideChain
Pdb2gmx
Editconf
Solvate
Grompp
Genion
Grompp
Mdrun
GmxEnergy
Grompp
Mdrun
Galaxy
PDB_API
Fetch Lysozime structure (1aki) from PDB API
MD_pdb2gmx
Building Gromacs Topology
MD_solvate
Adding Solvent Molecules
MD_genion
System neutralization + ionic concentration
MD_mdrun
Equilibration, Step 1: Energy Minimization Running Simulation
MD_mdrun
Equilibration, Step 2: NVT System Temperature Equilibration Running Simulation
MD_mdrun
Equilibration, Step 3: NPT System Pressure Equilibration Running Simulation
MD_mdrun
Equilibration, Step 4: Free MD Equilibration Running Simulation
Proteins Viewer
Node 19
Mutate
Mutate residue 2 (Valine) to a Tyrosine
MD_editconf
Building Solvent Box
MD_grompp
System neutralization + ionic concentration PreProcess
MD_grompp
Equilibration, Step 1: Energy Minimization PreProcess
MD_grompp
Equilibration, Step 2: NVT System Temperature Equilibration PreProcess
MD_grompp
Equilibration, Step 3: NPT System Pressure Equilibration PreProcess
MD_grompp
Equilibration, Step 4: Free MD Equilibration PreProcess
Binary Objects to Files
Saving Topology zip file from output table
Open for Innovation
KNIME



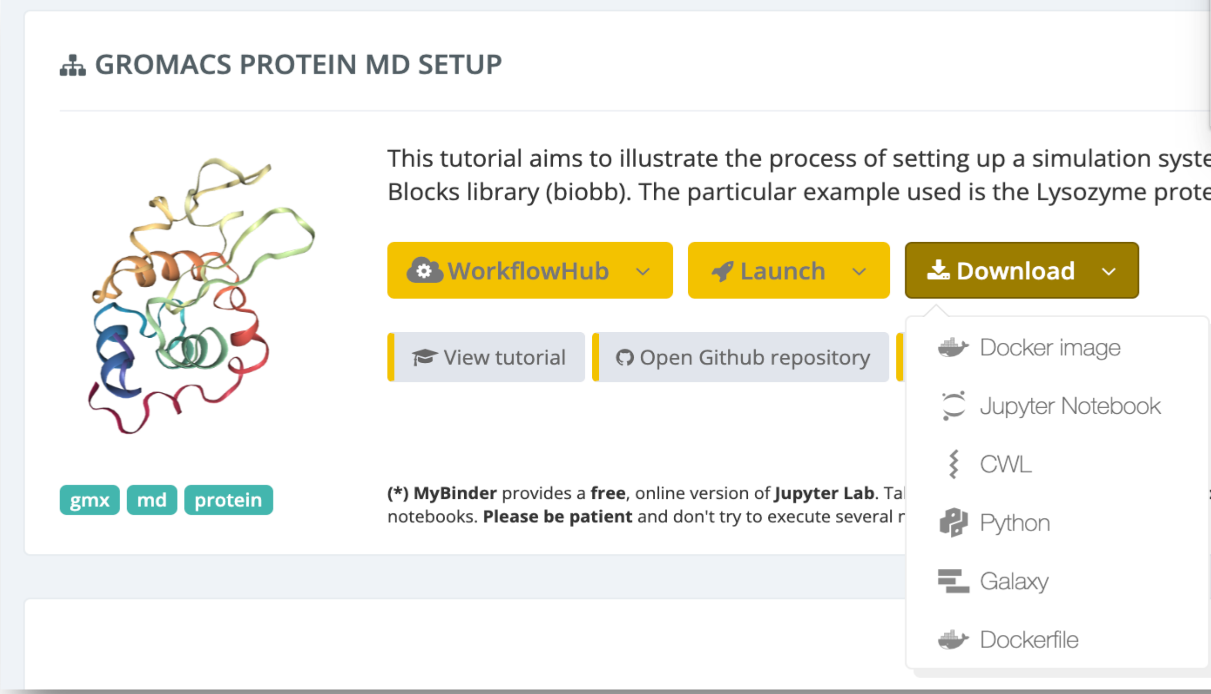
R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).

Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* **9**, 622 (2022).

Reusable

CONDA® → *Reproducibility*

biobb → *Use / Reuse*

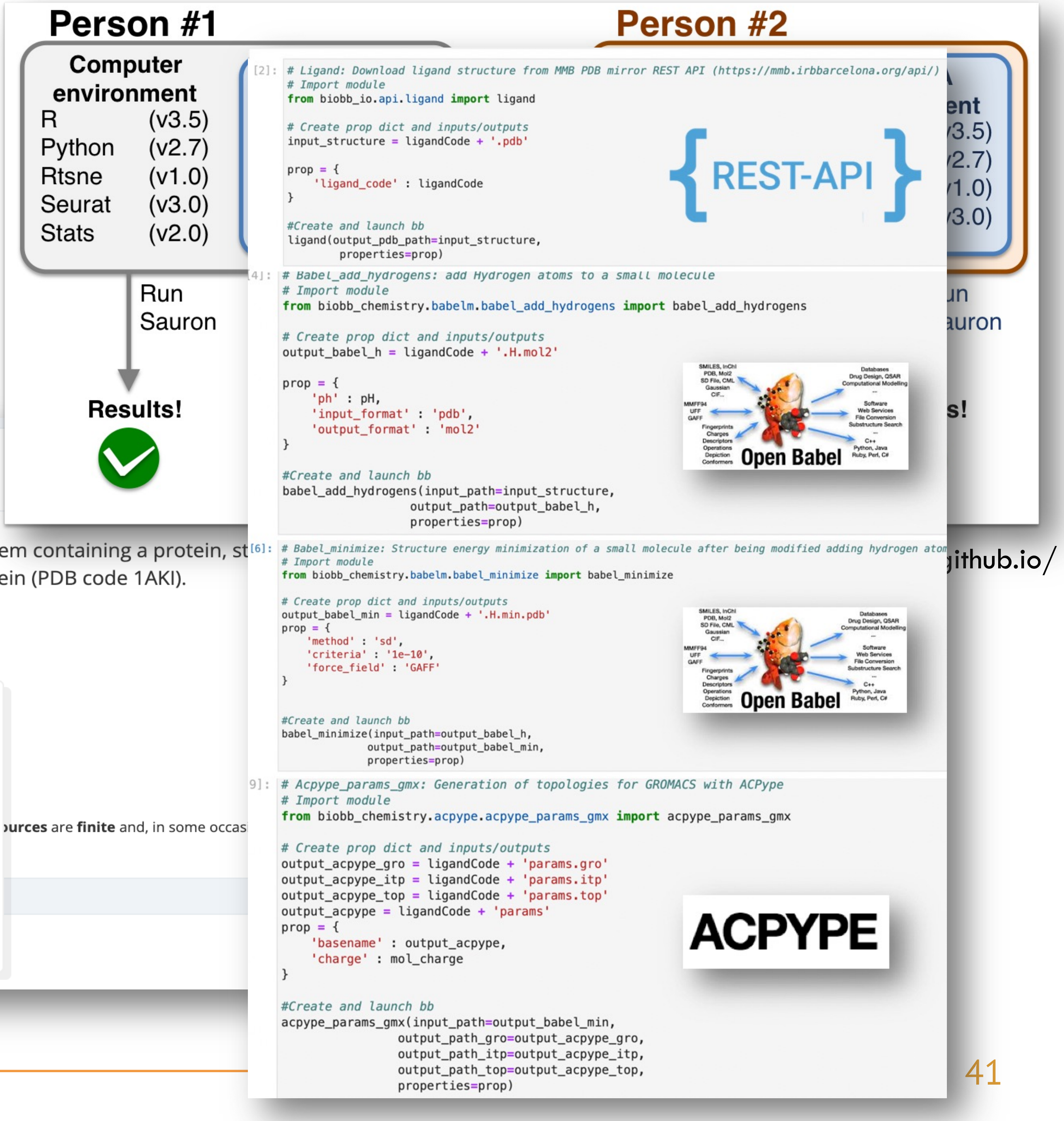


GROMACS PROTEIN MD SETUP

This tutorial aims to illustrate the process of setting up a simulation system containing a protein, st
Blocks library (biobb). The particular example used is the Lysozyme protein (PDB code 1AKI).

(*) MyBinder provides a **free**, online version of **Jupyter Lab**. Ta …urces are **finite** and, in some occas
notebooks. **Please be patient** and don't try to execute several r

…github.io/



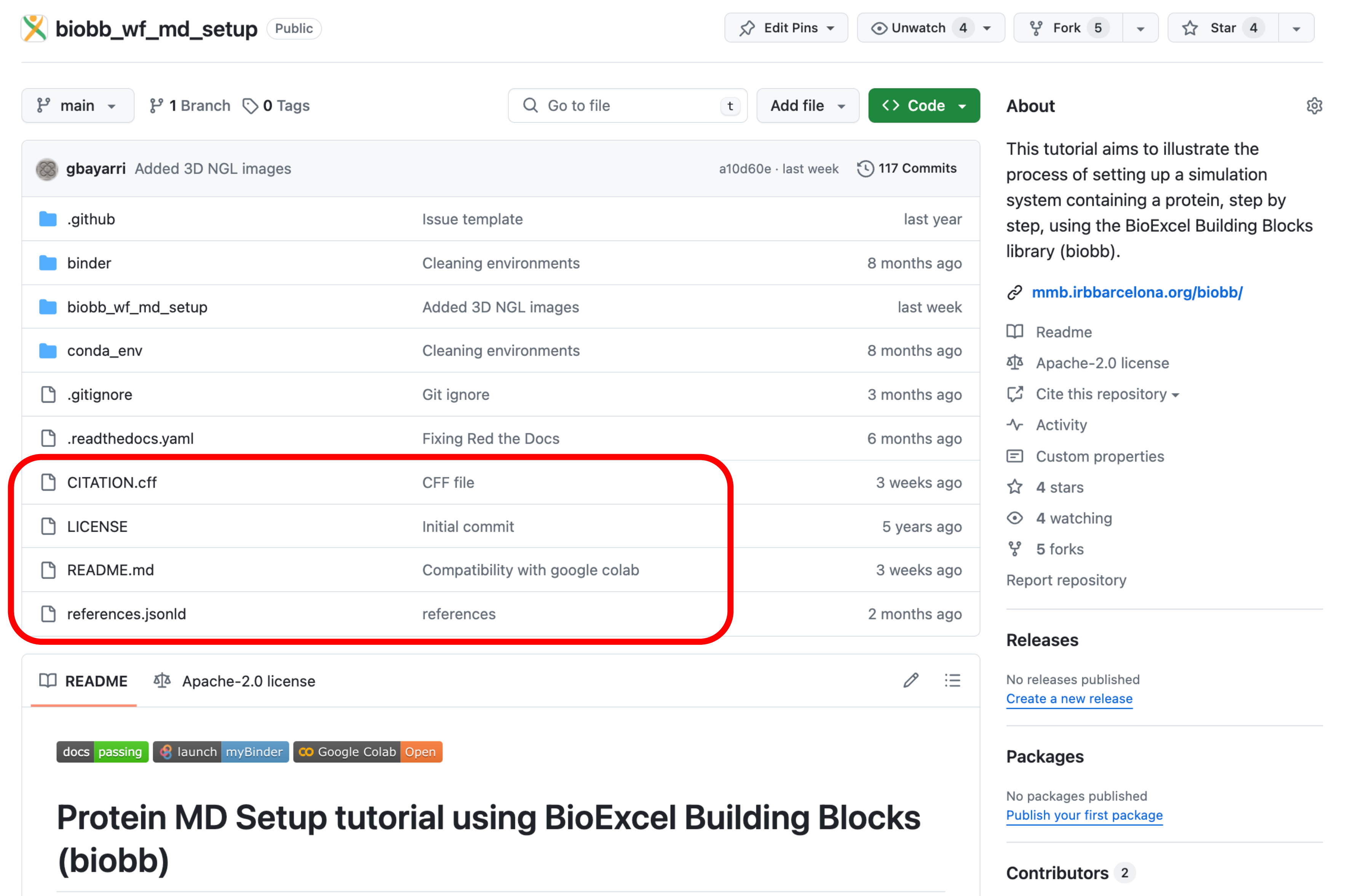
```
[2]: # Ligand: Download ligand structure from MMB PDB mirror REST API (https://mmb.irbbarcelona.org/api/)
     # Import module
     from biobb_io.api.ligand import ligand

     # 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)
```

```
[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)
```

```
[6]: # Babel_minimize: Structure energy minimization of a small molecule after being modified adding hydrogen atom
     # 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)
```

```
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)
```

Use / Reuse Attribution



biobb_wf_md_setup Public

Edit Pins | Unwatch 4 | Fork 5 | Star 4

main | 1 Branch | 0 Tags | Go to file | Add file | Code

gbayarri Added 3D NGL images a10d60e · last week 117 Commits

.github	Issue template	last year
binder	Cleaning environments	8 months ago
biobb_wf_md_setup	Added 3D NGL images	last week
conda_env	Cleaning environments	8 months ago
.gitignore	Git ignore	3 months ago
.readthedocs.yaml	Fixing Red the Docs	6 months ago
CITATION.cff	CFF file	3 weeks ago
LICENSE	Initial commit	5 years ago
README.md	Compatibility with google colab	3 weeks ago
references.jsonld	references	2 months ago

README | Apache-2.0 license

docs passing | launch myBinder | Google Colab Open

Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

About

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).

mmb.irbbarcelona.org/biobb/

Readme

Apache-2.0 license

Cite this repository

Activity

Custom properties

4 stars

4 watching

5 forks

Report repository

Releases

No releases published

Create a new release

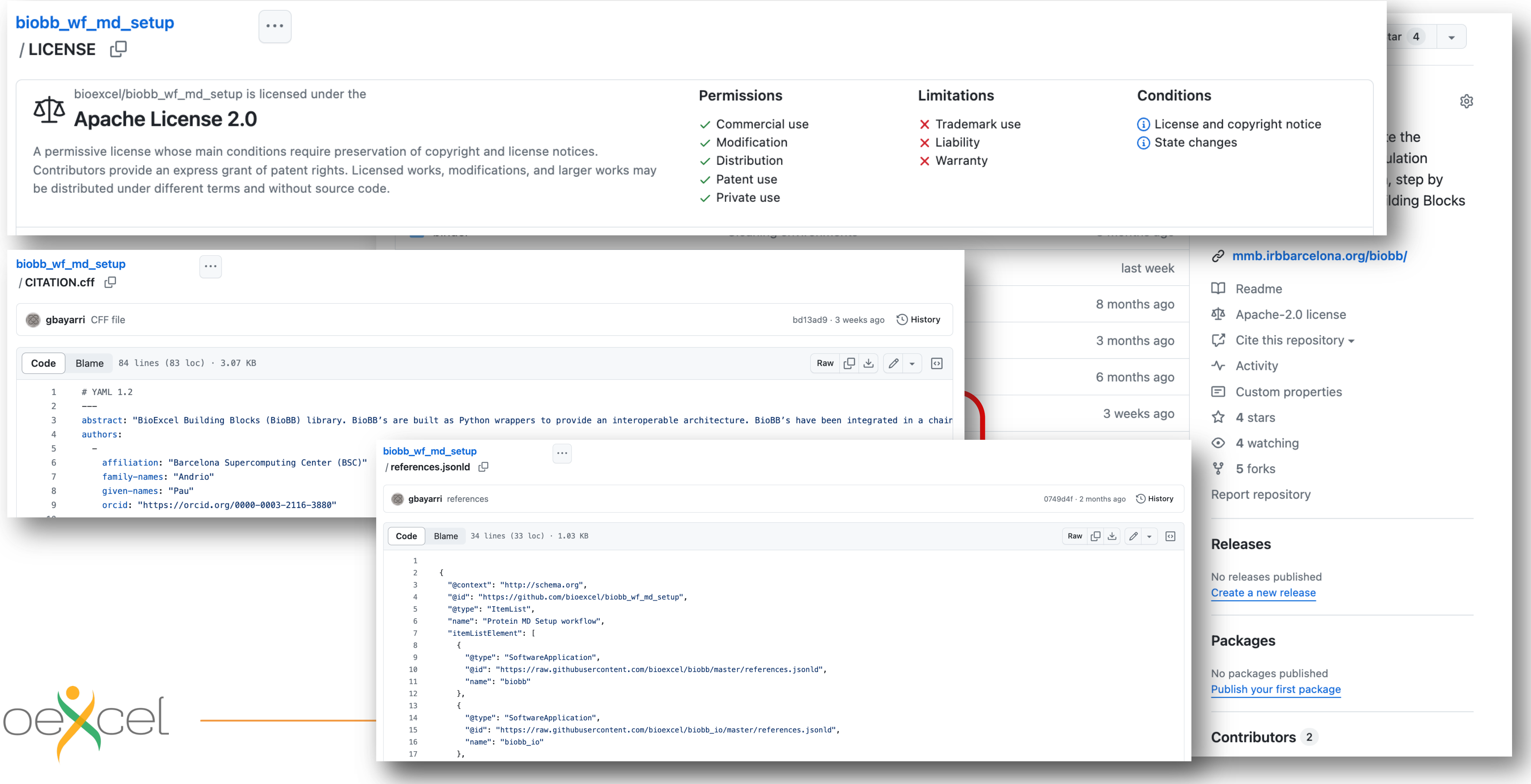
Packages

No packages published

Publish your first package

Contributors 2

Reusable



biobb_wf_md_setup

/ LICENSE

bioexcel/biobb_wf_md_setup is licensed under the

Apache License 2.0

A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions
- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Patent use
- ✓ Private use

Limitations
- ✗ Trademark use
- ✗ Liability
- ✗ Warranty

Conditions
- ⓘ License and copyright notice
- ⓘ State changes

biobb_wf_md_setup

/ CITATION.cff

gbayarri CFF file — bd13ad9 · 3 weeks ago — History

Code | Blame — 84 lines (83 loc) · 3.07 KB — Raw

```yaml
# YAML 1.2
---
abstract: "BioExcel Building Blocks (BioBB) library. BioBB's are built as Python wrappers to provide an interoperable architecture. BioBB's have been integrated in a chair
authors:
  -
    affiliation: "Barcelona Supercomputing Center (BSC)"
    family-names: "Andrio"
    given-names: "Pau"
    orcid: "https://orcid.org/0000-0003-2116-3880"
```

biobb_wf_md_setup

/ references.jsonld

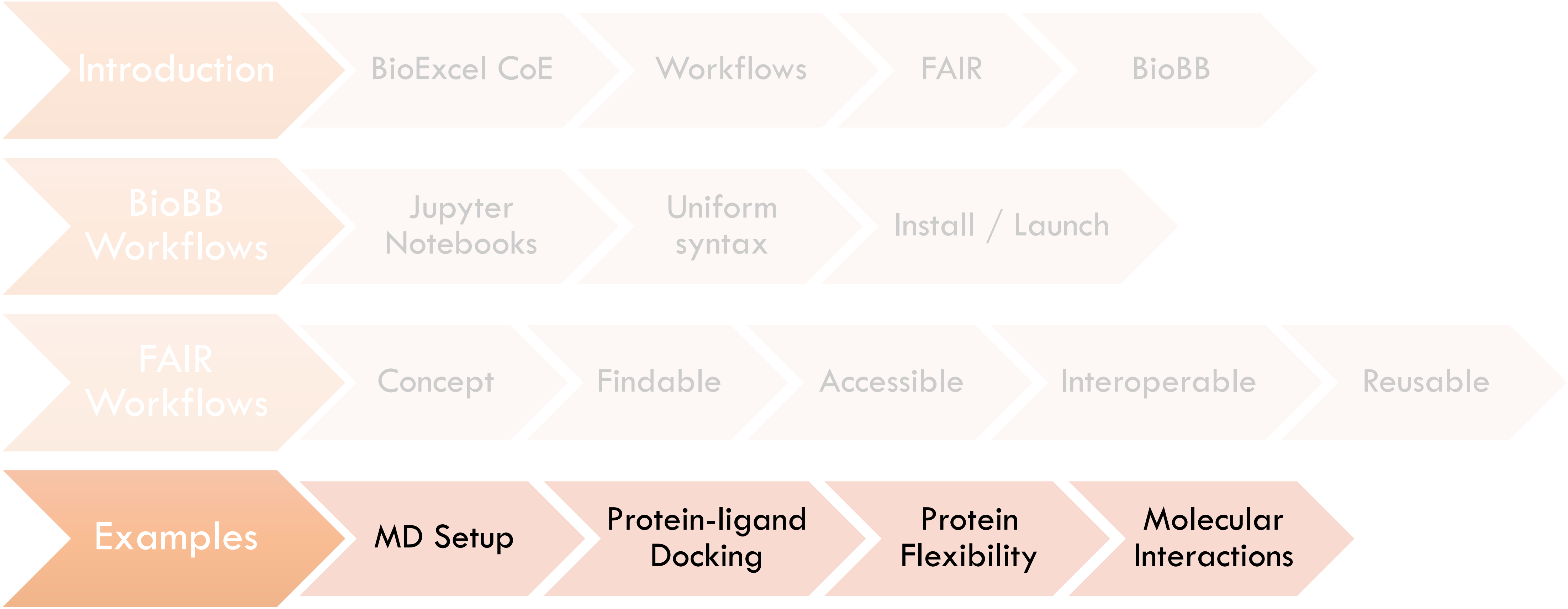
gbayarri references — 0749d4f · 2 months ago — History

Code | Blame — 34 lines (33 loc) · 1.03 KB — Raw

```json

{
  "@context": "http://schema.org",
  "@id": "https://github.com/bioexcel/biobb_wf_md_setup",
  "@type": "ItemList",
  "name": "Protein MD Setup workflow",
  "itemListElement": [
    {
      "@type": "SoftwareApplication",
      "@id": "https://raw.githubusercontent.com/bioexcel/biobb/master/references.jsonld",
      "name": "biobb"
    },
    {
      "@type": "SoftwareApplication",
      "@id": "https://raw.githubusercontent.com/bioexcel/biobb_io/master/references.jsonld",
      "name": "biobb_io"
    },
```

bioexcel

Index

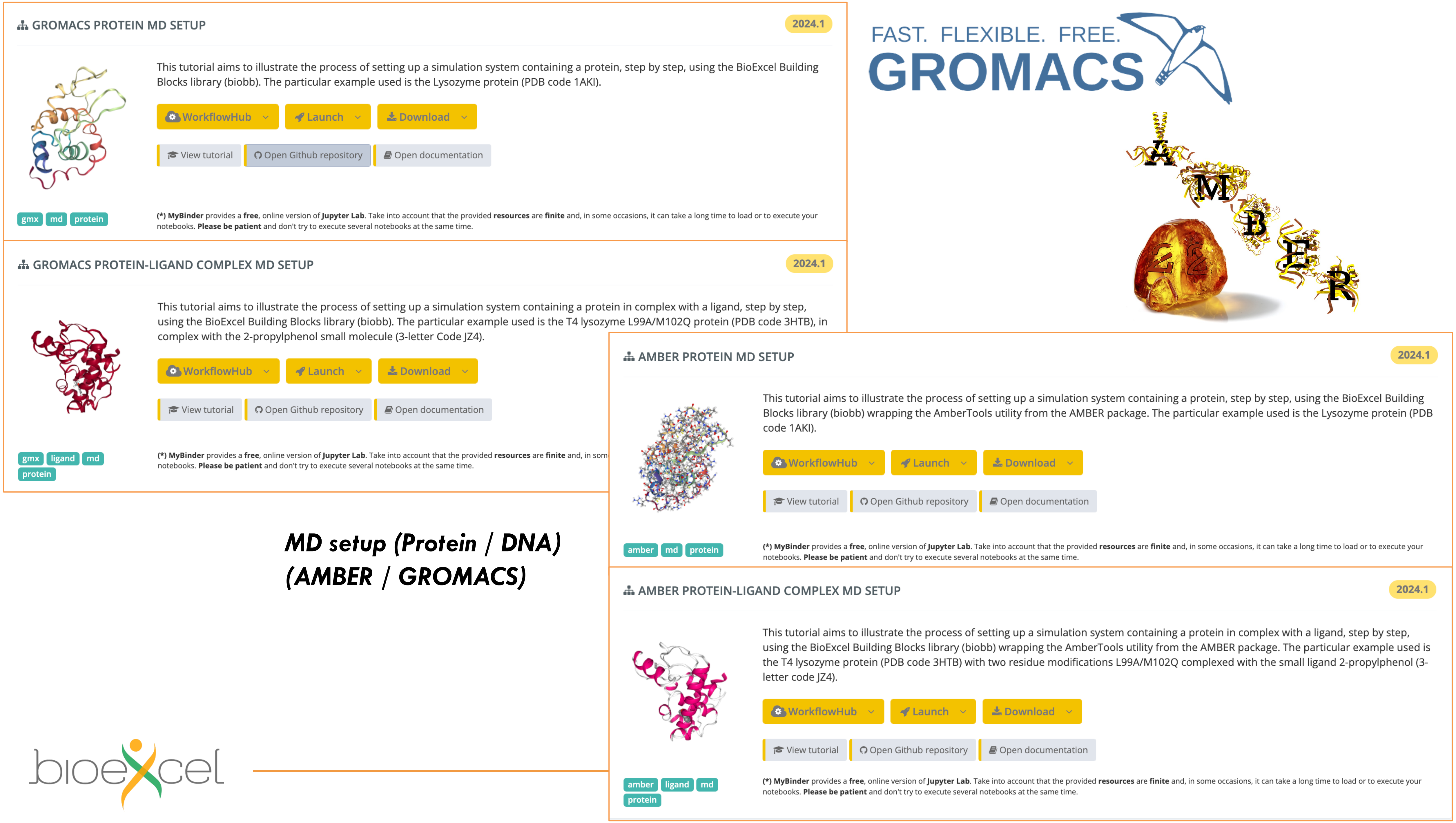


Introduction — BioExcel CoE — Workflows — FAIR — BioBB

BioBB Workflows — Jupyter Notebooks — Uniform syntax — Install / Launch

FAIR Workflows — Concept — Findable — Accessible — Interoperable — Reusable

Examples — MD Setup — Protein-ligand Docking — Protein Flexibility — Molecular Interactions



GROMACS PROTEIN MD SETUP

2024.1

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).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

GROMACS PROTEIN-LIGAND COMPLEX MD SETUP

2024.1

This tutorial aims to illustrate the process of setting up a simulation system containing a protein in complex with a ligand, step by step, using the BioExcel Building Blocks library (biobb). The particular example used is the T4 lysozyme L99A/M102Q protein (PDB code 3HTB), in complex with the 2-propylphenol small molecule (3-letter Code JZ4).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in som
notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

AMBER PROTEIN MD SETUP

2024.1

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) wrapping the AmberTools utility from the AMBER package. The particular example used is the Lysozyme protein (PDB code 1AKI).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

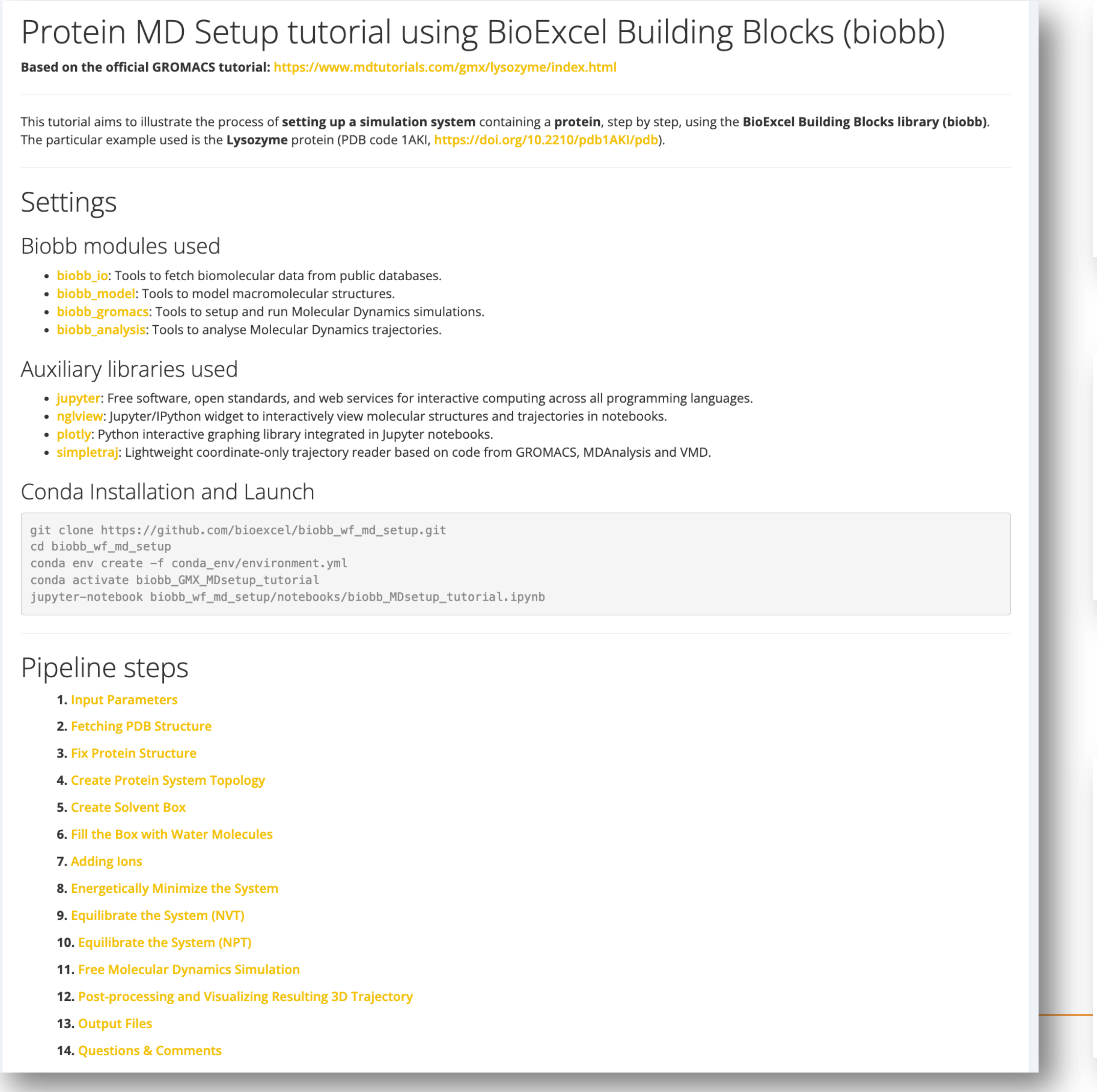
AMBER PROTEIN-LIGAND COMPLEX MD SETUP

2024.1

This tutorial aims to illustrate the process of setting up a simulation system containing a protein in complex with a ligand, step by step, using the BioExcel Building Blocks library (biobb) wrapping the AmberTools utility from the AMBER package. The particular example used is the T4 lysozyme protein (PDB code 3HTB) with two residue modifications L99A/M102Q complexed with the small ligand 2-propylphenol (3-letter code JZ4).

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

***MD setup (Protein / DNA)
(AMBER / GROMACS)***



Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Based on the official GROMACS tutorial: https://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.

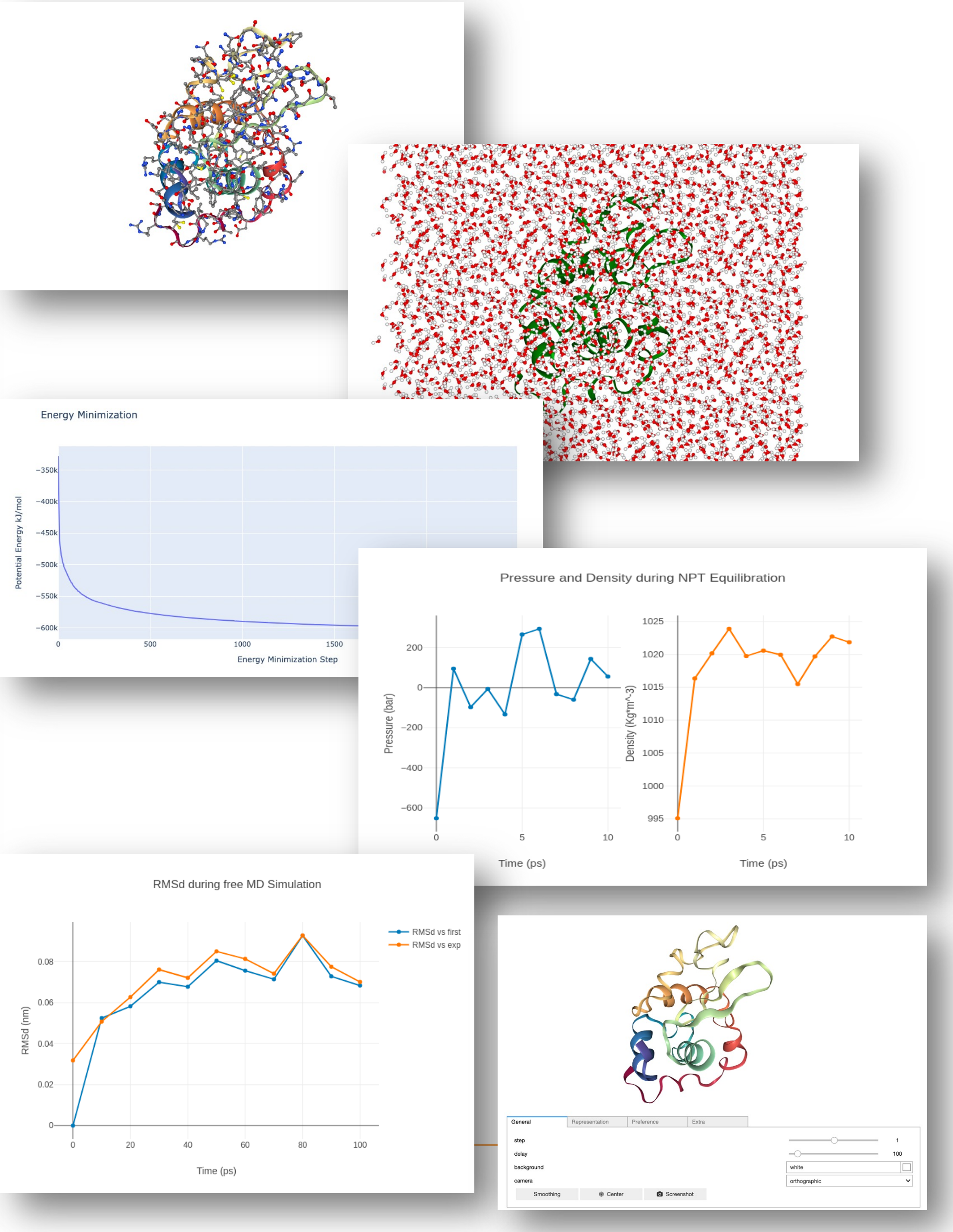
Auxiliary libraries used

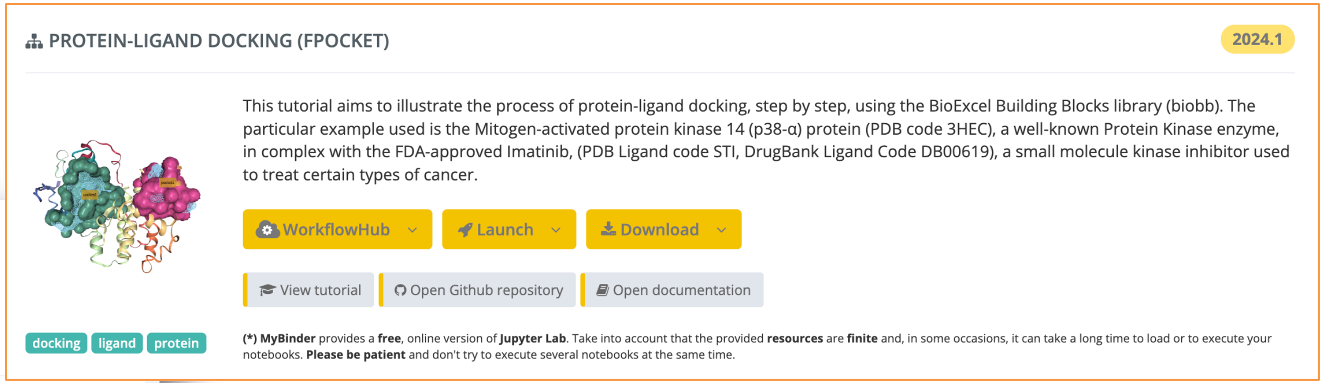
- jupyter: Free software, open standards, and web services for interactive computing across all programming languages.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in notebooks.
- plotly: Python interactive graphing library integrated in Jupyter notebooks.
- 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
conda activate biobb_GMX_MDsetup_tutorial
jupyter-notebook biobb_wf_md_setup/notebooks/biobb_MDsetup_tutorial.ipynb
```

Pipeline steps





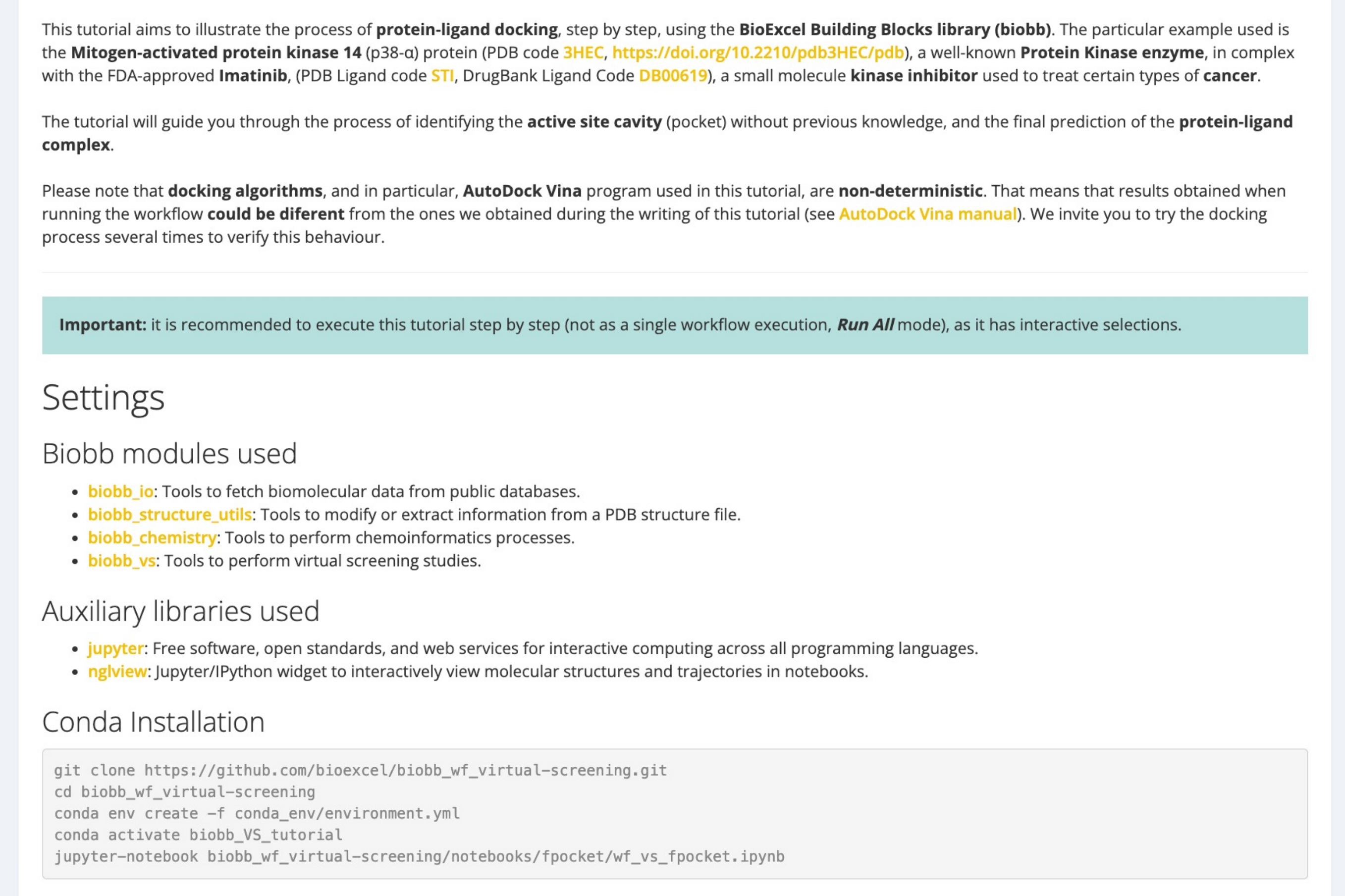
PROTEIN-LIGAND DOCKING (FPOCKET) 2024.1

This tutorial aims to illustrate the process of protein-ligand docking, step by step, using the BioExcel Building Blocks library (biobb). The particular example used is the Mitogen-activated protein kinase 14 (p38-α) protein (PDB code 3HEC), a well-known Protein Kinase enzyme, in complex with the FDA-approved Imatinib, (PDB Ligand code STI, DrugBank Ligand Code DB00619), a small molecule kinase inhibitor used to treat certain types of cancer.

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

Protein-ligand Docking tutorial using BioExcel Building Blocks (biobb)

-- Fpocket Version --



This tutorial aims to illustrate the process of **protein-ligand docking**, step by step, using the **BioExcel Building Blocks library (biobb)**. The particular example used is the **Mitogen-activated protein kinase 14** (p38-α) protein (PDB code 3HEC, https://doi.org/10.2210/pdb3HEC/pdb), a well-known **Protein Kinase enzyme**, in complex with the FDA-approved **Imatinib**, (PDB Ligand code STI, DrugBank Ligand Code DB00619), a small molecule **kinase inhibitor** used to treat certain types of **cancer**.

The tutorial will guide you through the process of identifying the **active site cavity** (pocket) without previous knowledge, and the final prediction of the **protein-ligand complex**.

Please note that **docking algorithms**, and in particular, **AutoDock Vina** program used in this tutorial, are **non-deterministic**. That means that results obtained when running the workflow **could be diferent** from the ones we obtained during the writing of this tutorial (see AutoDock Vina manual). We invite you to try the docking process several times to verify this behaviour.

Important: it is recommended to execute this tutorial step by step (not as a single workflow execution, ***Run All*** mode), as it has interactive selections.

Settings

Biobb modules used

- biobb_io: Tools to fetch biomolecular data from public databases.
- biobb_structure_utils: Tools to modify or extract information from a PDB structure file.
- biobb_chemistry: Tools to perform chemoinformatics processes.
- biobb_vs: Tools to perform virtual screening studies.

Auxiliary libraries used

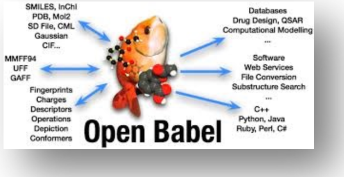
- jupyter: Free software, open standards, and web services for interactive computing across all programming languages.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in notebooks.

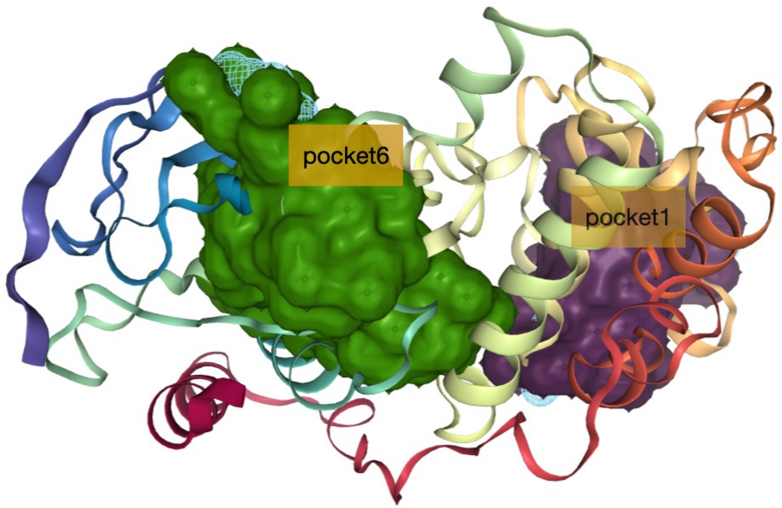
Conda Installation

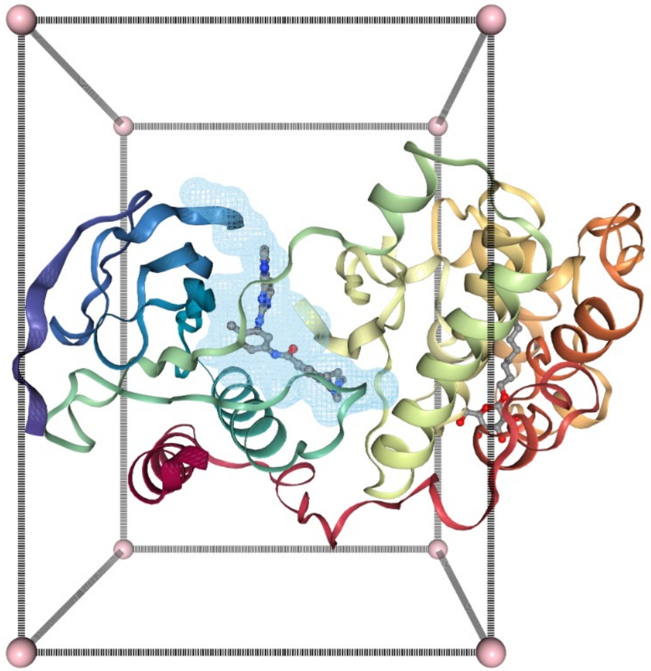
```
git clone https://github.com/bioexcel/biobb_wf_virtual-screening.git
cd biobb_wf_virtual-screening
conda env create -f conda_env/environment.yml
conda activate biobb_VS_tutorial
jupyter-notebook biobb_wf_virtual-screening/notebooks/fpocket/wf_vs_fpocket.ipynb
```

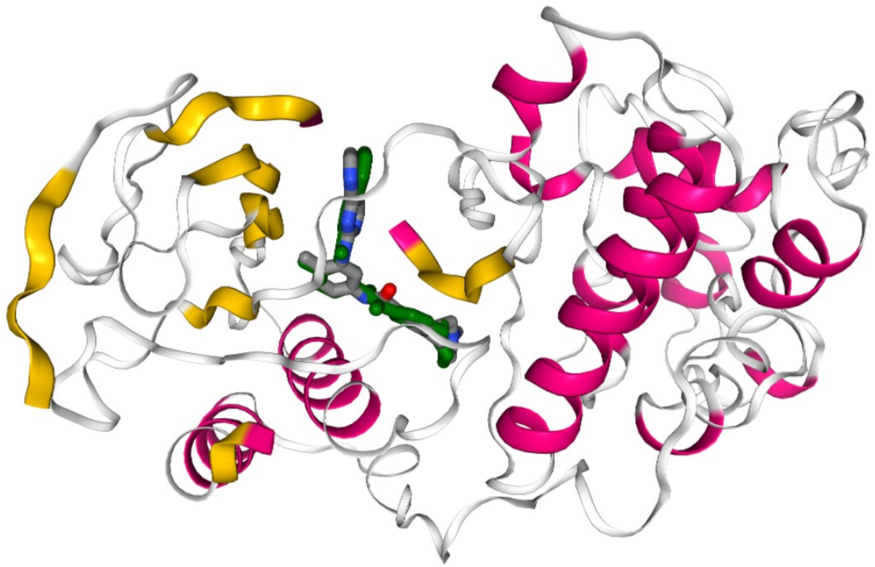
Pipeline steps

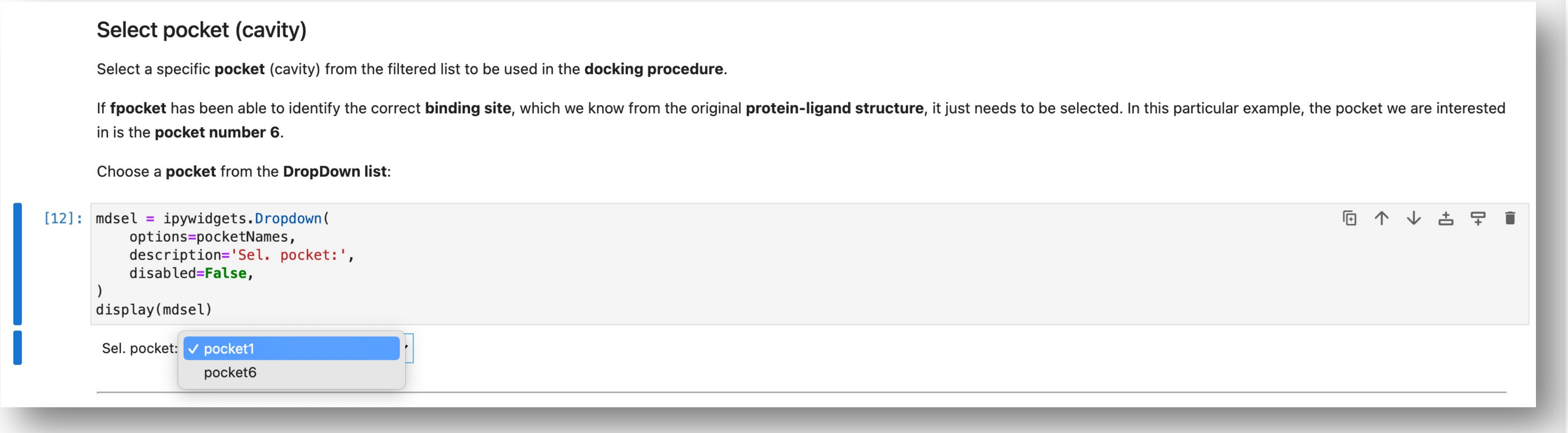
1. Input Parameters
2. Fetching PDB Structure
3. Extract Protein Structure
4. Computing Protein Cavities (fpocket)
5. Filtering Protein Cavities (fpocket output)
6. Extract Pocket Cavity
7. Generating Cavity Box
8. Downloading Small Molecule
9. Converting Small Molecule
10. Preparing Small Molecule (ligand) for Docking
11. Preparing Target Protein for Docking
12. Running the Docking
13. Extract a Docking Pose
14. Converting Ligand Pose to PDB format
15. Superposing Ligand Pose to the Target Protein Structure
16. Comparing final result with experimental structure
17. Questions & Comments











Select pocket (cavity)

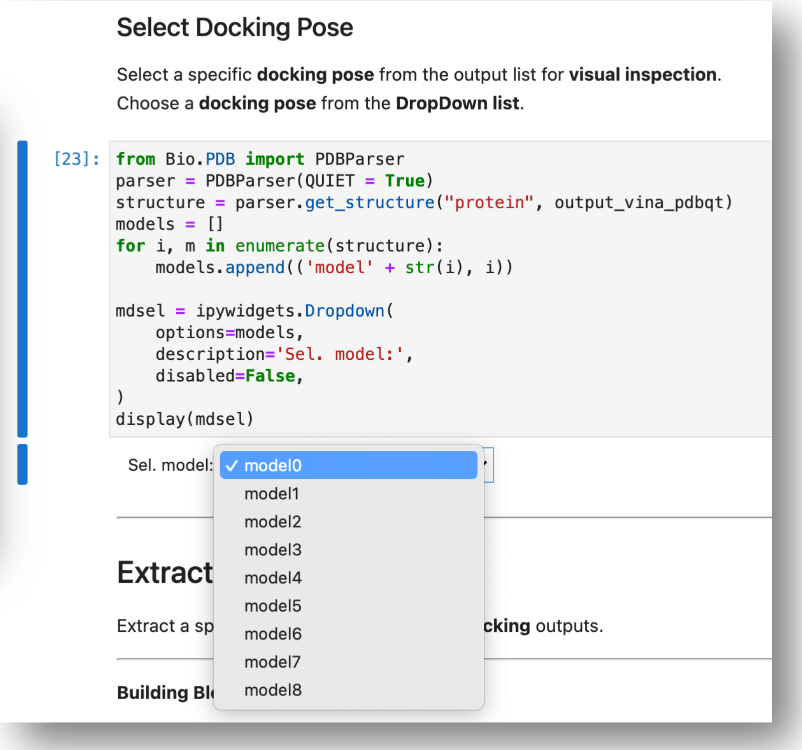
Select a specific **pocket** (cavity) from the filtered list to be used in the **docking procedure**.

If **fpocket** has been able to identify the correct **binding site**, which we know from the original **protein-ligand structure**, it just needs to be selected. In this particular example, the pocket we are interested in is the **pocket number 6**.

Choose a **pocket** from the **DropDown list**:

```
[12]: mdsel = ipywidgets.Dropdown(
          options=pocketNames,
          description='Sel. pocket:',
          disabled=False,
      )
      display(mdsel)
```

Sel. pocket: ✓ pocket1



Select Docking Pose

Select a specific **docking pose** from the output list for **visual inspection**.
Choose a **docking pose** from the **DropDown list**.

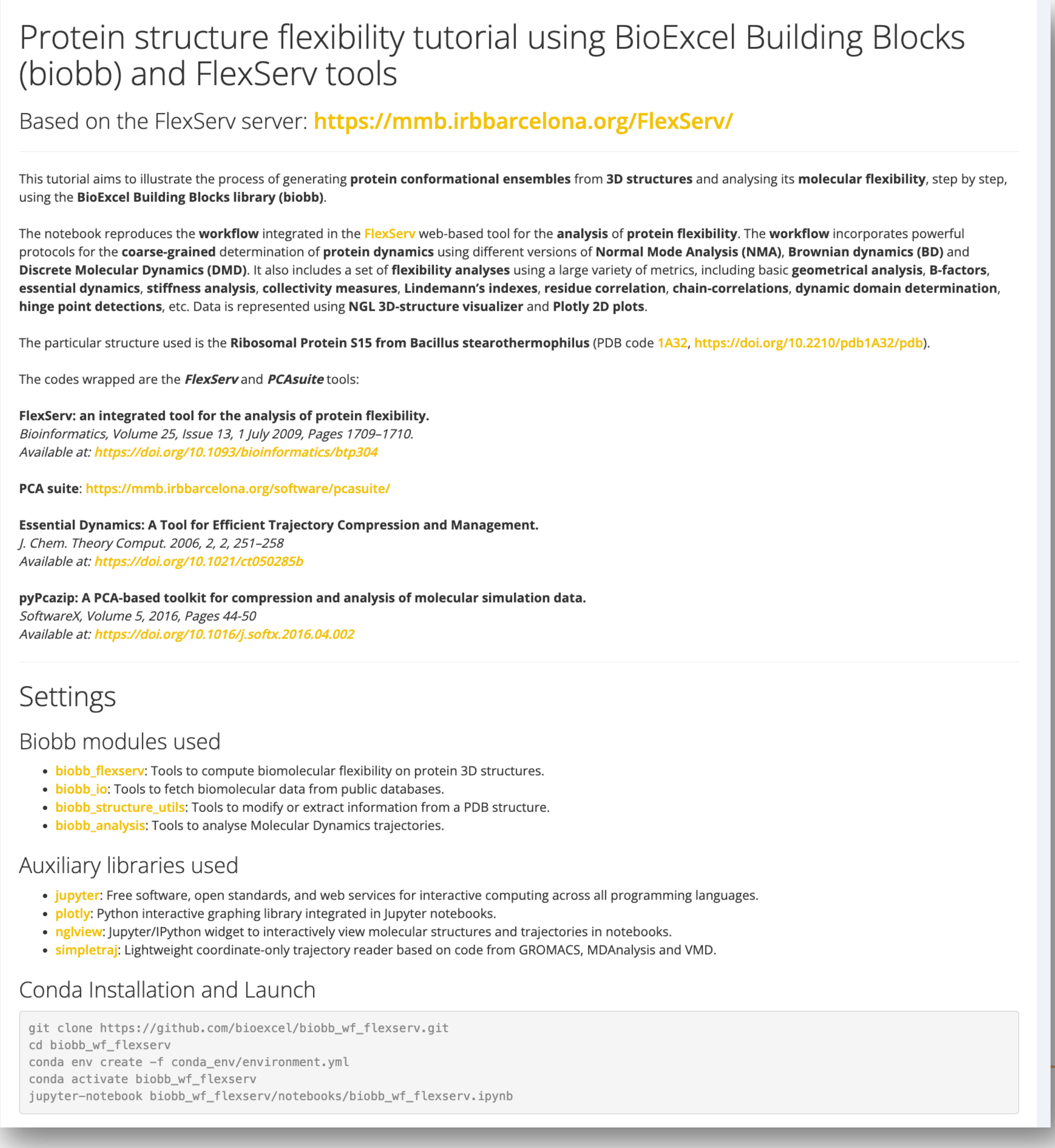
```
[23]: from Bio.PDB import PDBParser
      parser = PDBParser(QUIET = True)
      structure = parser.get_structure("protein", output_vina_pdbqt)
      models = []
      for i, m in enumerate(structure):
          models.append(('model' + str(i), i))

      mdsel = ipywidgets.Dropdown(
          options=models,
          description='Sel. model:',
          disabled=False,
      )
      display(mdsel)
```

Sel. model: ✓ model0

Extract

Extract a sp cking outputs.

Building Bl



Protein structure flexibility tutorial using BioExcel Building Blocks (biobb) and FlexServ tools

Based on the FlexServ server: https://mmb.irbbarcelona.org/FlexServ/

This tutorial aims to illustrate the process of generating **protein conformational ensembles** from **3D structures** and analysing its **molecular flexibility**, step by step, using the **BioExcel Building Blocks library (biobb)**.

The notebook reproduces the **workflow** integrated in the FlexServ web-based tool for the **analysis** of **protein flexibility**. The **workflow** incorporates powerful protocols for the **coarse-grained** determination of **protein dynamics** using different versions of **Normal Mode Analysis (NMA)**, **Brownian dynamics (BD)** and **Discrete Molecular Dynamics (DMD)**. It also includes a set of **flexibility analyses** using a large variety of metrics, including basic **geometrical analysis**, **B-factors**, **essential dynamics**, **stiffness analysis**, **collectivity measures**, **Lindemann's indexes**, **residue correlation**, **chain-correlations**, **dynamic domain determination**, **hinge point detections**, etc. Data is represented using **NGL 3D-structure visualizer** and **Plotly 2D plots**.

The particular structure used is the **Ribosomal Protein S15 from Bacillus stearothermophilus** (PDB code 1A32, https://doi.org/10.2210/pdb1A32/pdb).

The codes wrapped are the ***FlexServ*** and ***PCAsuite*** tools:

FlexServ: an integrated tool for the analysis of protein flexibility.
Bioinformatics, Volume 25, Issue 13, 1 July 2009, Pages 1709–1710.
Available at: https://doi.org/10.1093/bioinformatics/btp304

PCA suite: https://mmb.irbbarcelona.org/software/pcasuite/

Essential Dynamics: A Tool for Efficient Trajectory Compression and Management.
J. Chem. Theory Comput. 2006, 2, 2, 251–258
Available at: https://doi.org/10.1021/ct050285b

pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data.
SoftwareX, Volume 5, 2016, Pages 44-50
Available at: https://doi.org/10.1016/j.softx.2016.04.002

Settings

Biobb modules used

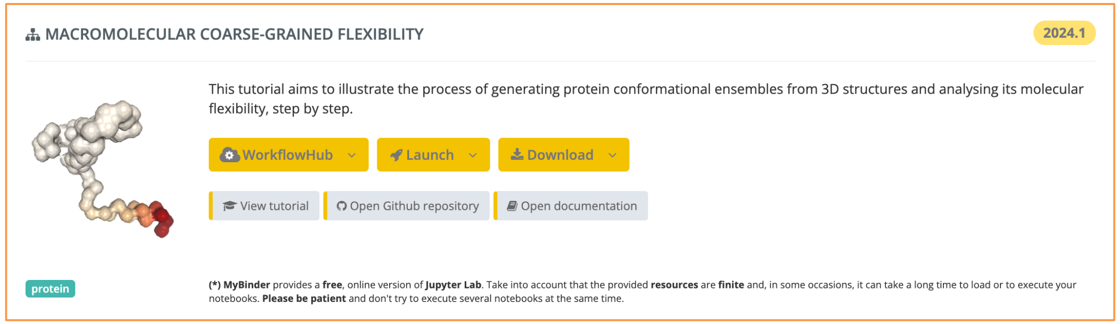
- biobb_flexserv: Tools to compute biomolecular flexibility on protein 3D structures.
- biobb_io: Tools to fetch biomolecular data from public databases.
- biobb_structure_utils: Tools to modify or extract information from a PDB structure.
- biobb_analysis: Tools to analyse Molecular Dynamics trajectories.

Auxiliary libraries used

- jupyter: Free software, open standards, and web services for interactive computing across all programming languages.
- plotly: Python interactive graphing library integrated in Jupyter notebooks.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in 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_flexserv.git
cd biobb_wf_flexserv
conda env create -f conda_env/environment.yml
conda activate biobb_wf_flexserv
jupyter-notebook biobb_wf_flexserv/notebooks/biobb_wf_flexserv.ipynb
```



MACROMOLECULAR COARSE-GRAINED FLEXIBILITY

2024.1

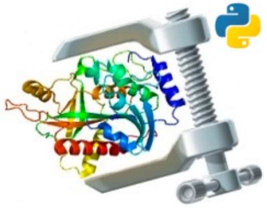
This tutorial aims to illustrate the process of generating protein conformational ensembles from 3D structures and analysing its molecular flexibility, step by step.

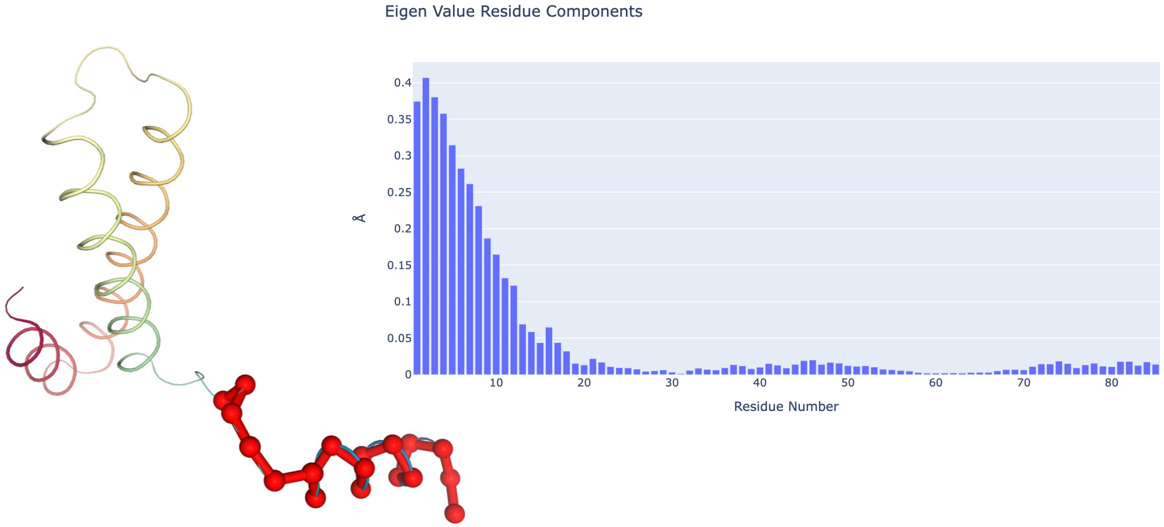
(*) MyBinder provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.

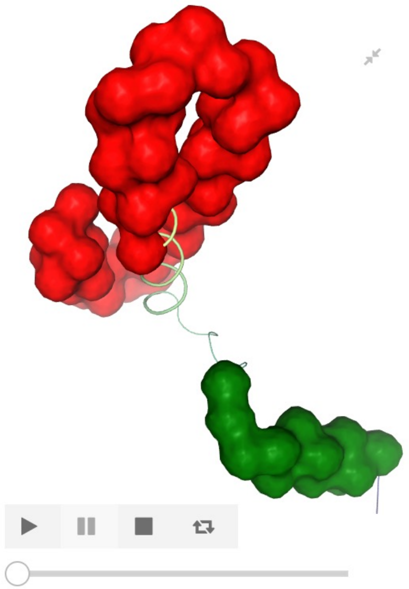
Pipeline steps

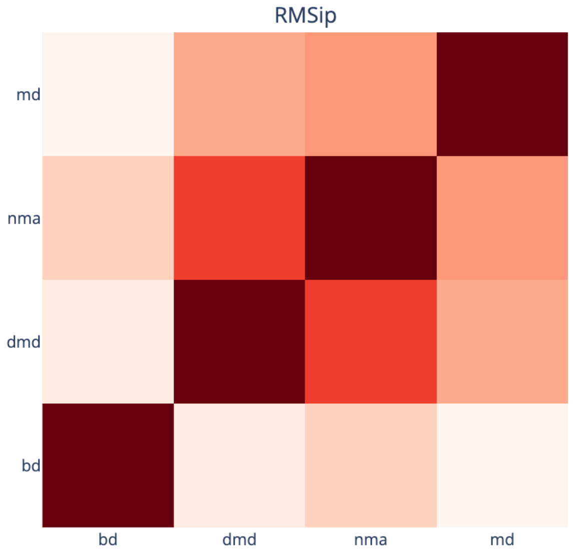
1. Input Parameters
2. Molecular Flexibility Representation/Generation
    1. Fetching PDB structure
    2. Generate Coarse Grain structure
    3. Brownian Dynamics (BD)
    4. Discrete Molecular Dynamics (DMD)
    5. Normal Mode Analysis (NMA)
3. Molecular Flexibility Analyses
    1. PCA Zip compression
    2. PCA Zip uncompression
    3. PCA report
    4. PCA eigenvectors
    5. PCA mode animations
    6. PCA Bfactors
    7. PCA hinge points
    8. PCA Stiffness
    9. PCA collectivity index
    10. PCA similarity index
4. Questions & Comments

FlexServ









Brownian Dynamics

The **Brownian Dynamics** (BD) method introduces the protein in an **stochastic bath** that keeps the **temperature constant** and modulates the otherwise extreme oscillations of the residues. This bath is simulated with two terms accounting for a **velocity-dependent friction** and **stochastic forces** due to the **solvent environment**. **Velocity Verlet** algorithm is used to solve the **stochastic differential equation** (equation of motion) for **alpha-carbons** ($C\alpha$):

$$m\dot{v}_i = \gamma v_i + F_i + \eta_i$$

where **m** stands for the **effective mass** of $C\alpha$ (see below), v and $\dot{v}$ stands for velocity and acceleration, **F** represent the force, γ is the inverse of a characteristic **time at which the particle loses its energy in a given solvent**, and finally the **random term** is considered a Robust white noise $\eta(t)$ with autocorrelation given by:

$$\langle \eta_l(t)\ \eta_n(t') \rangle = 2mk_BT\gamma\delta_{ln}\delta(t-t')$$

where k_B is the **Boltzmann constant**, and **t** is the **temperature of the stochastic bath**. The **Dirac functions** δ_{ln} and $\delta(t-t')$ force the **independence of the components of the noise vector**.

The **equation of motion** is integrated using **Verlet's algorithm**, giving for the **velocities** and **positions** after time:

$$\vec{v}_i = e^{-\frac{\Delta t}{\tau}}\vec{v}_i^0 + \frac{1}{\gamma}\left(1 - e^{-\frac{\Delta t}{\tau}}\right)\vec{F}_i^0 + \Delta\vec{v}_i^G$$

and

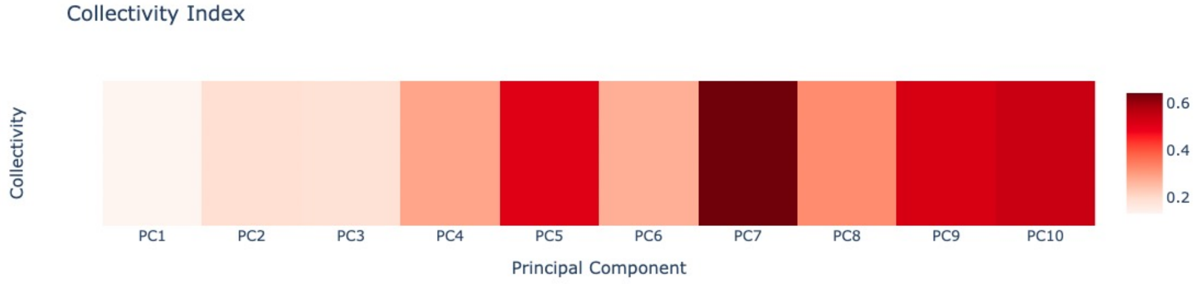
$$\vec{r}_i = \vec{r}_i^0 + \tau\left(1 - e^{-\frac{\Delta t}{\tau}}\right)\vec{v}_i^0 + \frac{\Delta t}{\gamma}\left(1 - \frac{\tau}{\Delta t}\left(1 - e^{-\frac{\Delta t}{\tau}}\right)\right)\vec{F}_i + \Delta\vec{r}_i^G$$

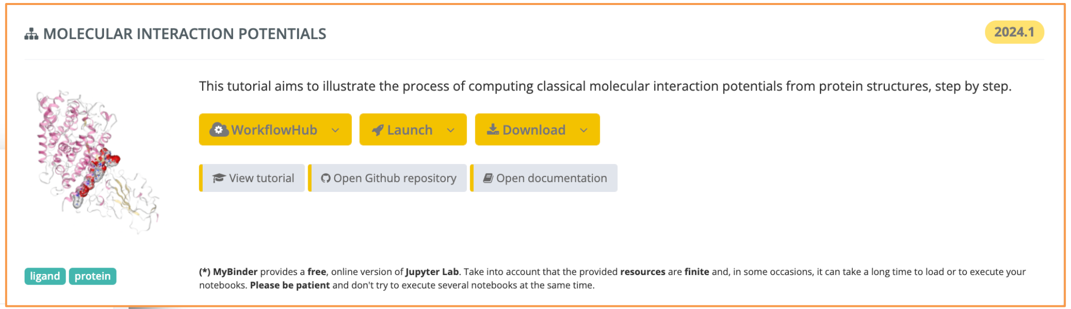
where $\tau = m\gamma^{-1}$ is the **characteristic time**, and $\Delta\vec{r}_i^G$, $\Delta\vec{v}_i^G$ are the **changes** in **position** and **velocity** induced by the **stochastic term**.

The **potential energy** used to compute **forces** in the **equation of motion** assumes a **coarse-grained representation of the protein** ($C\alpha$-only) and a **quasi-harmonic representation** of the **interactions** (similar to that suggested by Kovacs et al. 2004):

$$U_{ij} = \frac{1}{2}C\left(\frac{r^*}{|\vec{r}_{ij}^0|}\right)^6\left(\vec{r}_{ij} - \vec{r}_{ij}^0\right)^2$$

where $r_{ij} = r_i - r_j$ stands for the vector connecting $C\alpha$ atoms i and j.



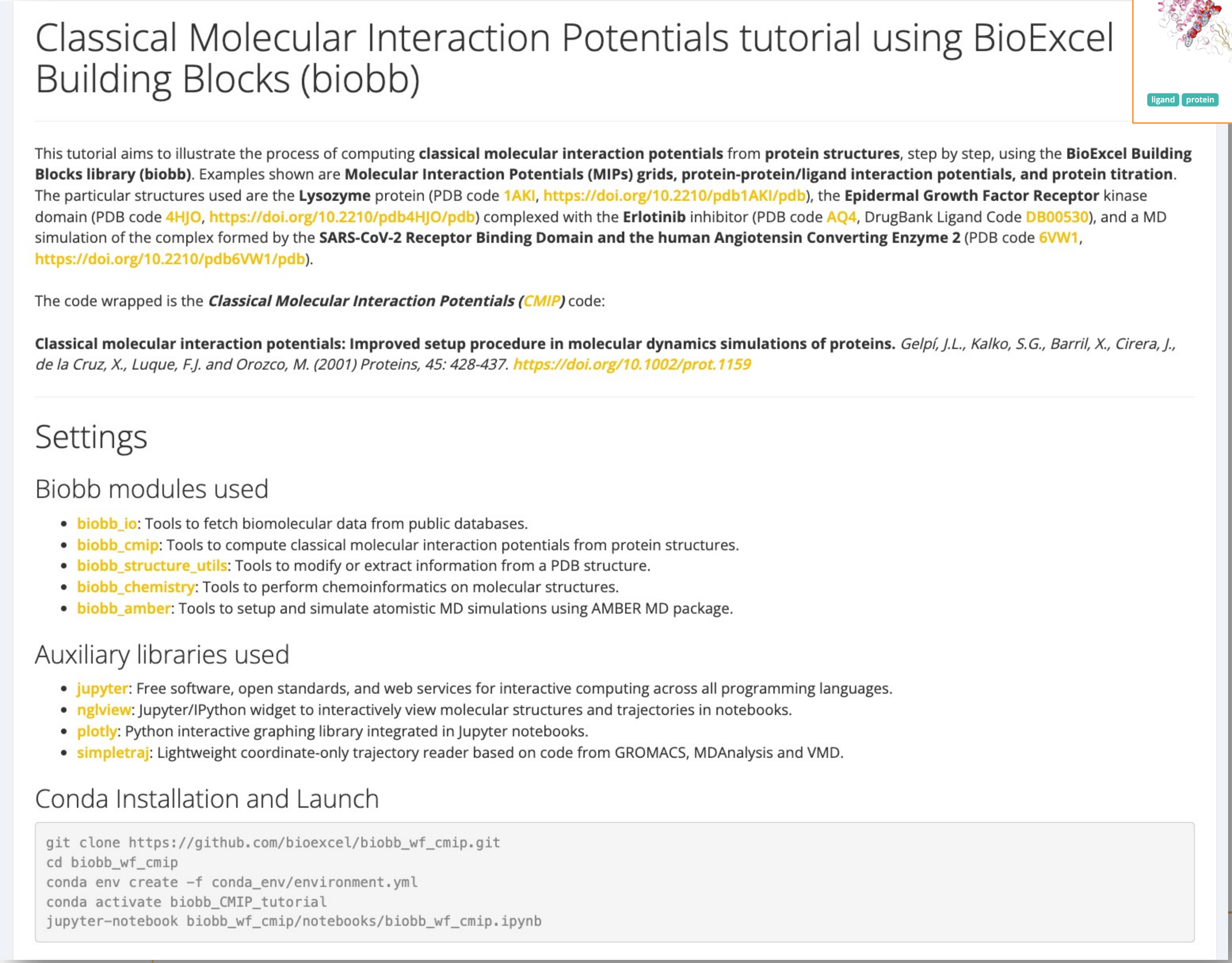


MOLECULAR INTERACTION POTENTIALS

2024.1

This tutorial aims to illustrate the process of computing classical molecular interaction potentials from protein structures, step by step.

(*) **MyBinder** provides a **free**, online version of **Jupyter Lab**. Take into account that the provided **resources** are **finite** and, in some occasions, it can take a long time to load or to execute your notebooks. **Please be patient** and don't try to execute several notebooks at the same time.



Classical Molecular Interaction Potentials tutorial using BioExcel Building Blocks (biobb)

This tutorial aims to illustrate the process of computing **classical molecular interaction potentials** from **protein structures**, step by step, using the **BioExcel Building Blocks library (biobb)**. Examples shown are **Molecular Interaction Potentials (MIPs) grids, protein-protein/ligand interaction potentials, and protein titration**. The particular structures used are the **Lysozyme** protein (PDB code 1AKI, https://doi.org/10.2210/pdb1AKI/pdb), the **Epidermal Growth Factor Receptor** kinase domain (PDB code 4HJO, https://doi.org/10.2210/pdb4HJO/pdb) complexed with the **Erlotinib** inhibitor (PDB code AQ4, DrugBank Ligand Code DB00530), and a MD simulation of the complex formed by the **SARS-CoV-2 Receptor Binding Domain and the human Angiotensin Converting Enzyme 2** (PDB code 6VW1, https://doi.org/10.2210/pdb6VW1/pdb).

The code wrapped is the ***Classical Molecular Interaction Potentials (CMIP)*** code:

Classical molecular interaction potentials: Improved setup procedure in molecular dynamics simulations of proteins. *Gelpí, J.L., Kalko, S.G., Barril, X., Cirera, J., de la Cruz, X., Luque, F.J. and Orozco, M. (2001) Proteins, 45: 428-437. https://doi.org/10.1002/prot.1159*

Settings

Biobb modules used

- biobb_io: Tools to fetch biomolecular data from public databases.
- biobb_cmip: Tools to compute classical molecular interaction potentials from protein structures.
- biobb_structure_utils: Tools to modify or extract information from a PDB structure.
- biobb_chemistry: Tools to perform chemoinformatics on molecular structures.
- biobb_amber: Tools to setup and simulate atomistic MD simulations using AMBER MD package.

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_cmip.git
cd biobb_wf_cmip
conda env create -f conda_env/environment.yml
conda activate biobb_CMIP_tutorial
jupyter-notebook biobb_wf_cmip/notebooks/biobb_wf_cmip.ipynb
```

Pipeline steps

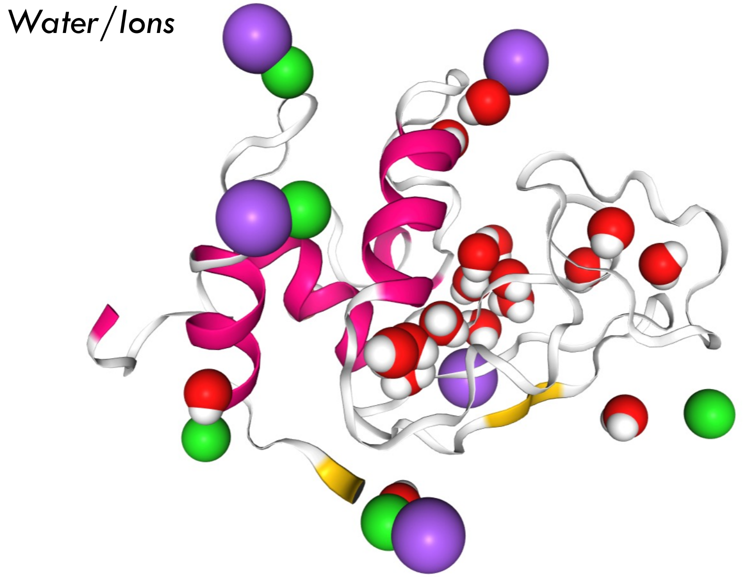
1. Input Parameters
2. Fetching PDB structure
3. CMIP PDB preparation (from PDB structure)
4. Structural water molecules & ions
5. Molecular Interaction Potentials
    1. Positive MIP (MIP+)
    2. Negative MIP (MIP-)
    3. Neutral MIP (MIPn)
6. Interaction Potential Energies
    1. Protein-Ligand Interaction Energies
        1. Fetching PDB structure
        2. Removing water molecules
        3. Creating ligand topology
        4. Generating system topology
        5. Minimizing the energy of the system
        6. Preparing the structures for CMIP
        7. Computing the Protein-Ligand interaction energies
    2. Protein-Protein Interaction Energies
        1. CMIP PDB preparation (from MD)
        2. CMIP Boxes
        3. RDB Interaction Potential Energies
        4. hACE2 Interaction Potential Energies
7. Questions & Comments

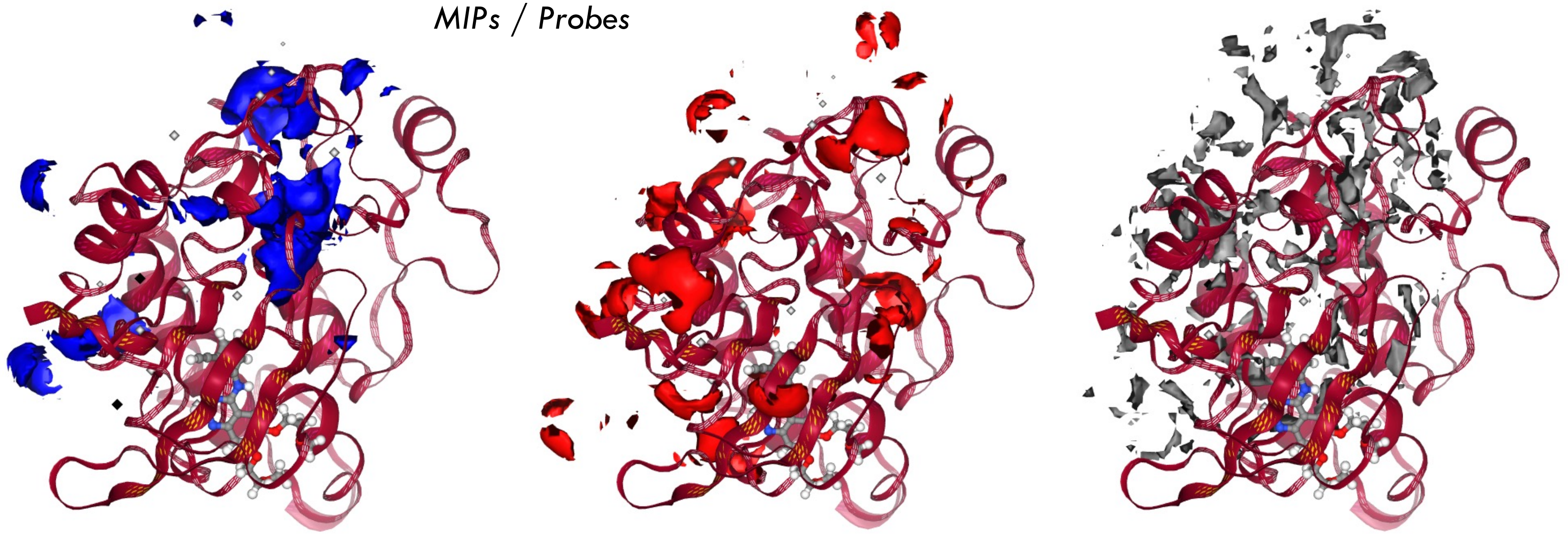
AmberTools19

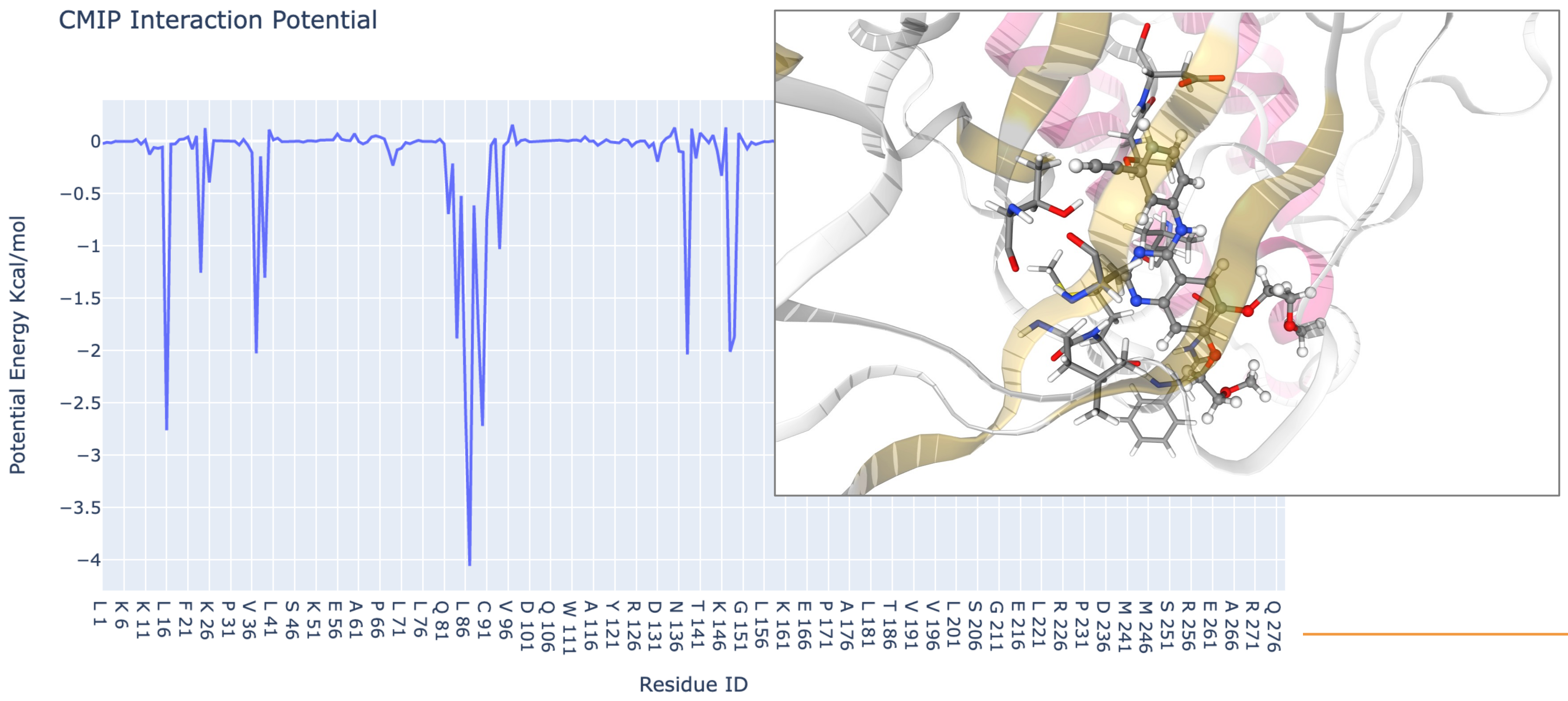
ACPYPE

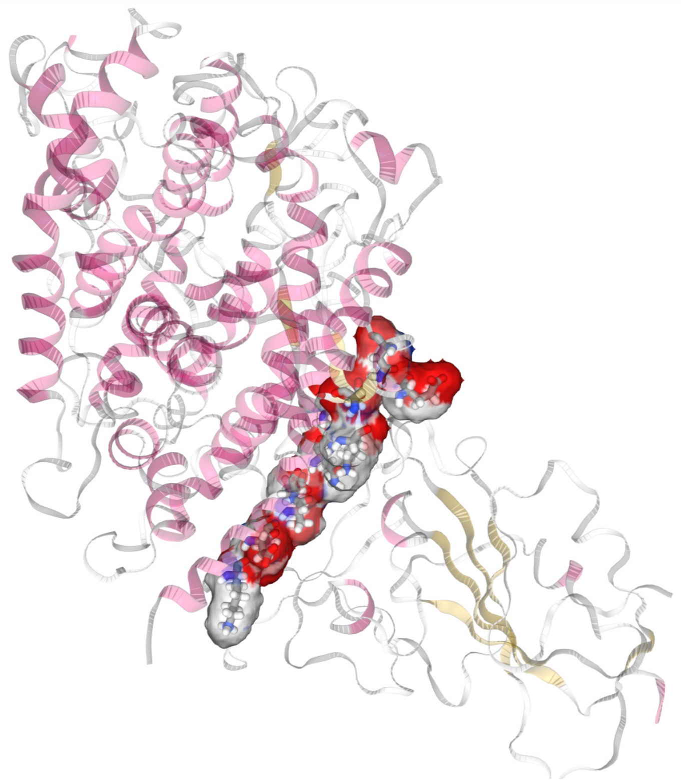
Open Babel

CMIP


Water/Ions


MIPs / Probes


CMIP Interaction Potential
Potential Energy Kcal/mol
0
−0.5
−1
−1.5
−2
−2.5
−3
−3.5
−4
L 1
K 6
K 11
L 16
F 21
K 26
P 31
V 36
L 41
S 46
K 51
E 56
A 61
P 66
L 71
L 76
Q 81
L 86
C 91
V 96
D 101
Q 106
W 111
A 116
Y 121
R 126
D 131
N 136
T 141
K 146
G 151
L 156
K 161
E 166
P 171
A 176
L 181
T 186
V 191
V 196
L 201
S 206
G 211
E 216
L 221
R 226
P 231
D 236
M 241
M 246
S 251
R 256
E 261
A 266
R 271
Q 276
Residue ID



BioBB Demonstration Workflows

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

python™

jupyter

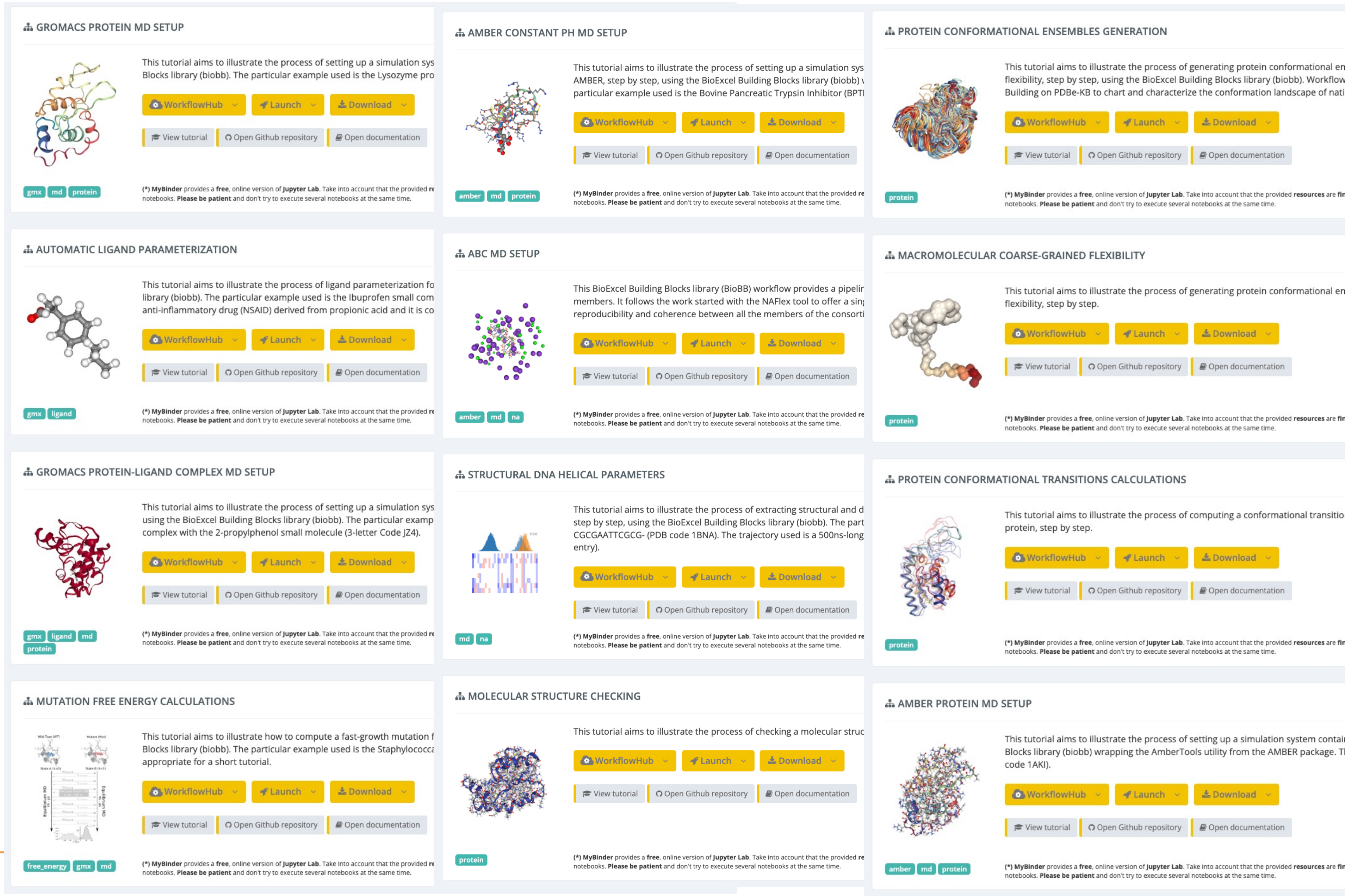
COMMON WORKFLOW LANGUAGE

Google colab

Galaxy

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

bioexcel



GROMACS PROTEIN MD SETUP

This tutorial aims to illustrate the process of setting up a simulation sys
Blocks library (biobb). The particular example used is the Lysozyme pro

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

AUTOMATIC LIGAND PARAMETERIZATION

This tutorial aims to illustrate the process of ligand parameterization fo
library (biobb). The particular example used is the Ibuprofen small com
anti-inflammatory drug (NSAID) derived from propionic acid and it is co

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

GROMACS PROTEIN-LIGAND COMPLEX MD SETUP

This tutorial aims to illustrate the process of setting up a simulation sys
using the BioExcel Building Blocks library (biobb). The particular examp
complex with the 2-propylphenol small molecule (3-letter Code JZ4).

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

MUTATION FREE ENERGY CALCULATIONS

This tutorial aims to illustrate how to compute a fast-growth mutation f
Blocks library (biobb). The particular example used is the Staphylococc
appropriate for a short tutorial.

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

AMBER CONSTANT PH MD SETUP

This tutorial aims to illustrate the process of setting up a simulation sys
AMBER, step by step, using the BioExcel Building Blocks library (biobb) v
particular example used is the Bovine Pancreatic Trypsin Inhibitor (BPTI

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

ABC MD SETUP

This BioExcel Building Blocks library (BioBB) workflow provides a pipelir
members. It follows the work started with the NAFlex tool to offer a sin
reproducibility and coherence between all the members of the consorti

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

STRUCTURAL DNA HELICAL PARAMETERS

This tutorial aims to illustrate the process of extracting structural and d
step by step, using the BioExcel Building Blocks library (biobb). The part
CGCGAATTCGCG- (PDB code 1BNA). The trajectory used is a 500ns-long
entry).

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

MOLECULAR STRUCTURE CHECKING

This tutorial aims to illustrate the process of checking a molecular struc

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided re
notebooks. Please be patient and don't try to execute several notebooks at the same time.

PROTEIN CONFORMATIONAL ENSEMBLES GENERATION

This tutorial aims to illustrate the process of generating protein conformational en
flexibility, step by step, using the BioExcel Building Blocks library (biobb). Workflow
Building on PDBe-KB to chart and characterize the conformation landscape of nati

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided resources are fir
notebooks. Please be patient and don't try to execute several notebooks at the same time.

MACROMOLECULAR COARSE-GRAINED FLEXIBILITY

This tutorial aims to illustrate the process of generating protein conformational en
flexibility, step by step.

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided resources are fir
notebooks. Please be patient and don't try to execute several notebooks at the same time.

PROTEIN CONFORMATIONAL TRANSITIONS CALCULATIONS

This tutorial aims to illustrate the process of computing a conformational transitio
protein, step by step.

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided resources are fir
notebooks. Please be patient and don't try to execute several notebooks at the same time.

AMBER PROTEIN MD SETUP

This tutorial aims to illustrate the process of setting up a simulation system contai
Blocks library (biobb) wrapping the AmberTools utility from the AMBER package. T
code 1AKI).

(*) MyBinder provides a free, online version of Jupyter Lab. Take into account that the provided resources are fir
notebooks. Please be patient and don't try to execute several notebooks at the same time.

Conclusions

- Science is going towards **FAIR**.
- **Research software** and **workflows** should follow the trend.

- **New technologies** are helping (GitHub, packages, containers, software registries, etc.).

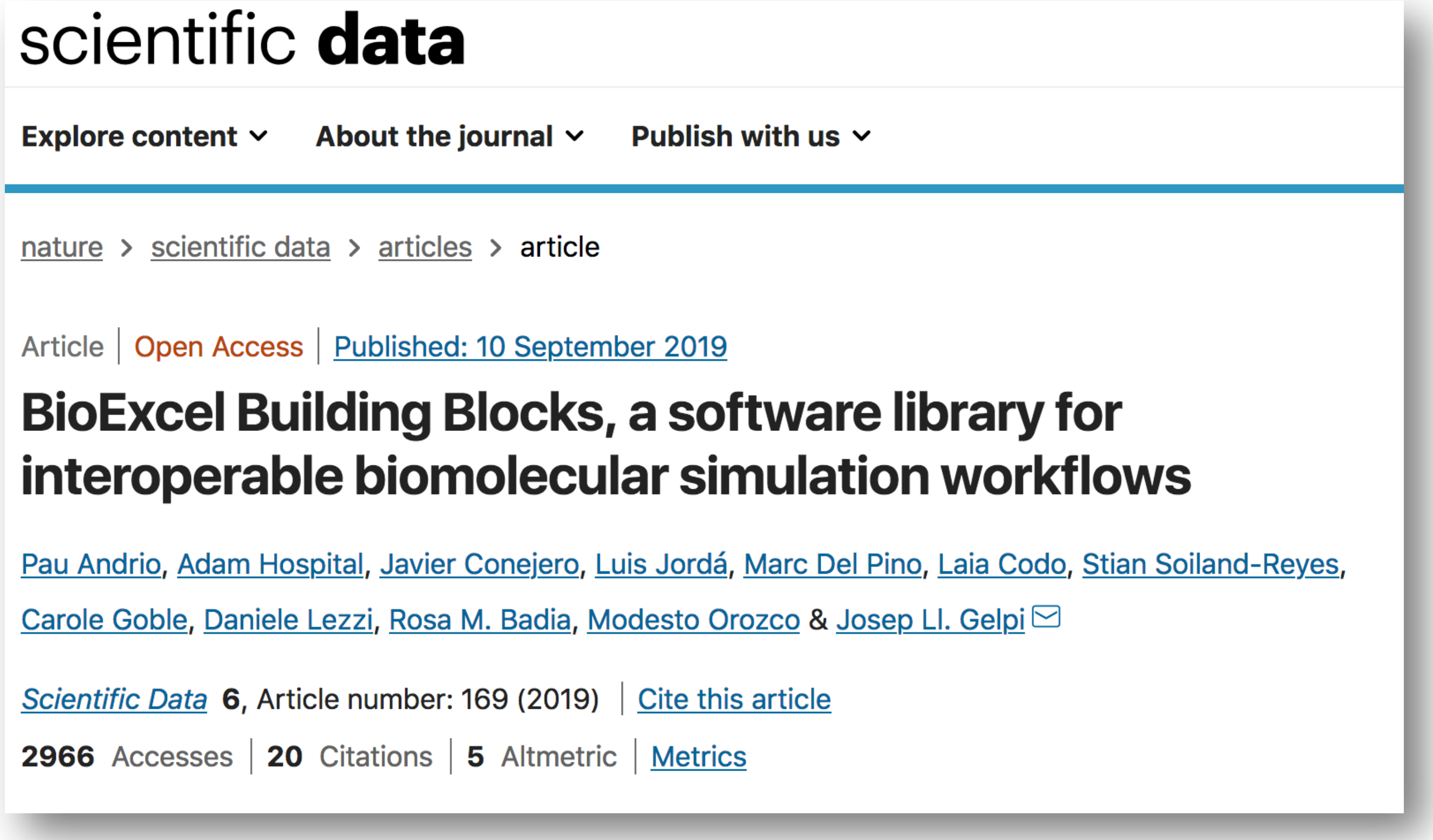
- The collection of **BioBB workflows** is a **proof of concept**.

More about BioBB & BioExcel

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

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

- **BioBB Webinar:** https://bioexcel.eu/webinar-computational-biomolecular-simulation-workflows-with-bioexcel-building-blocks-2020-09-10/



Article | Open Access | Published: 10 September 2019

BioExcel Building Blocks, a software library for interoperable biomolecular simulation workflows

Pau Andrio, Adam Hospital, Javier Conejero, Luis Jordá, Marc Del Pino, Laia Codo, Stian Soiland-Reyes, Carole Goble, Daniele Lezzi, Rosa M. Badia, Modesto Orozco & Josep Ll. Gelpi

Scientific Data **6**, Article number: 169 (2019) | Cite this article

2966 Accesses | **20** Citations | **5** Altmetric | Metrics

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

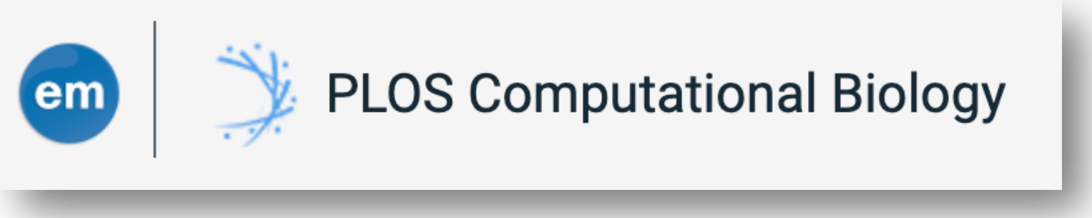
Genís Bayarri[1], Pau Andrio[3], Josep Lluís Gelpí[2,3], Adam Hospital[1]* and Modesto Orozco[1,2]*

*adam.hospital@irbbarcelona; *modesto.orozco@irbbarcelona.org

[1] Institute for Research in Biomedicine (IRB Barcelona), the Barcelona Institute of Science and Technology, Barcelona, Spain

[2] Department of Biochemistry and Biomedicine. University of Barcelona. Barcelona, Spain

[3] Barcelona Supercomputing Center (BSC). Barcelona, Spain



em

PLOS Computational Biology

Just accepted!

Acknowledgments & Questions

INSTITUTE FOR RESEARCH IN BIOMEDICINE

IRB BARCELONA

Federica Battistini

Genís Bayarri

Modesto Orozco

thank you

Barcelona Supercomputing Center

Centro Nacional de Supercomputación

Pau Andrio

Josep Ll. Gelpí

bioexcel

Centre of Excellence for Computational Biomolecular Research

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).

Co-funded by the European Union

EuroHPC Joint Undertaking

bioexcel