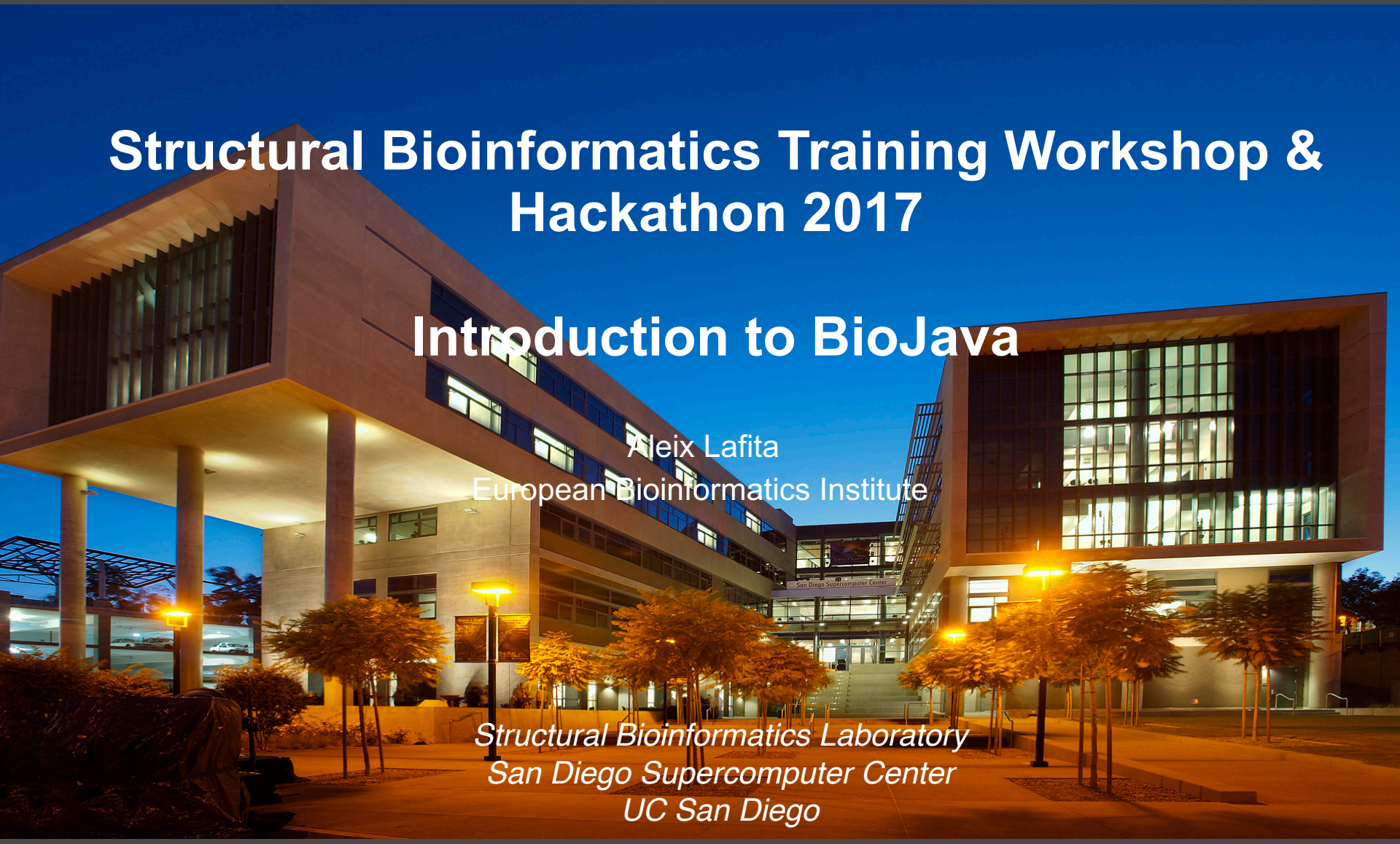
Structural Bioinformatics Training Workshop & Hackathon 2017

Introduction to BioJava

Aleix Lafita
European Bioinformatics Institute

Structural Bioinformatics Laboratory
San Diego Supercomputer Center
UC San Diego



SDSC SAN DIEGO SUPERCOMPUTER CENTER RCSB PDB UC San Diego

PART 1
BioJava Project

History

Resources

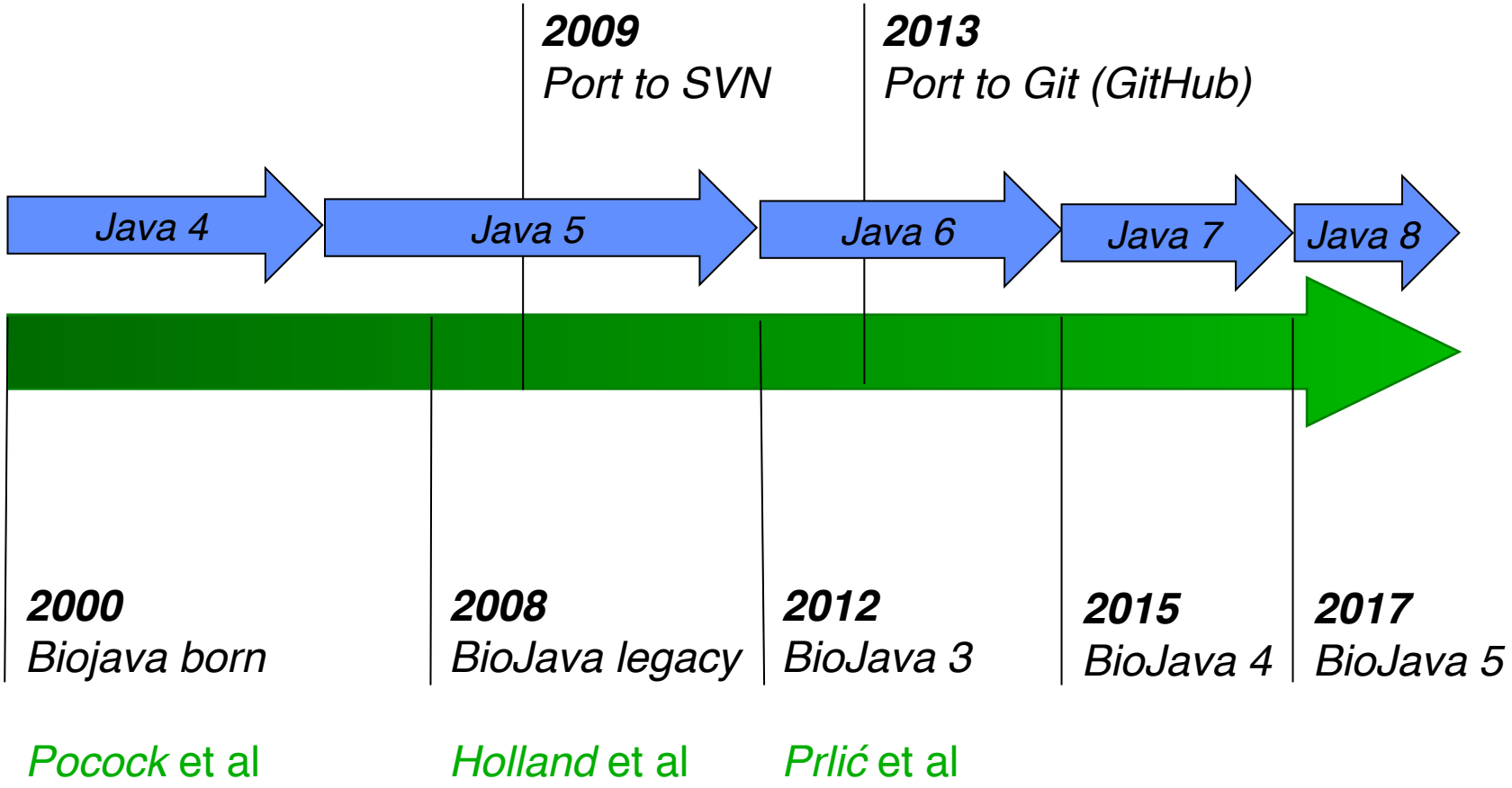
Setup

BioJava

BioJava is an **open-source** project dedicated to providing a **Java framework** for processing **biological data**

File parsers	Data models	Algorithms	Resources
• FASTA • PDB • MMCIF • MMTF	• Biological sequences • Protein structures	• Sequence and structure alignment • DSSP • Symmetry	• BLAST • Protein domains: SCOP, CATH, ECOD

History of BioJava



BioJava Resources

Website: https://biojava.org

Source code: https://github.com/biojava/biojava/

Wikipedia: https://en.wikipedia.org/wiki/BioJava

General information about the project

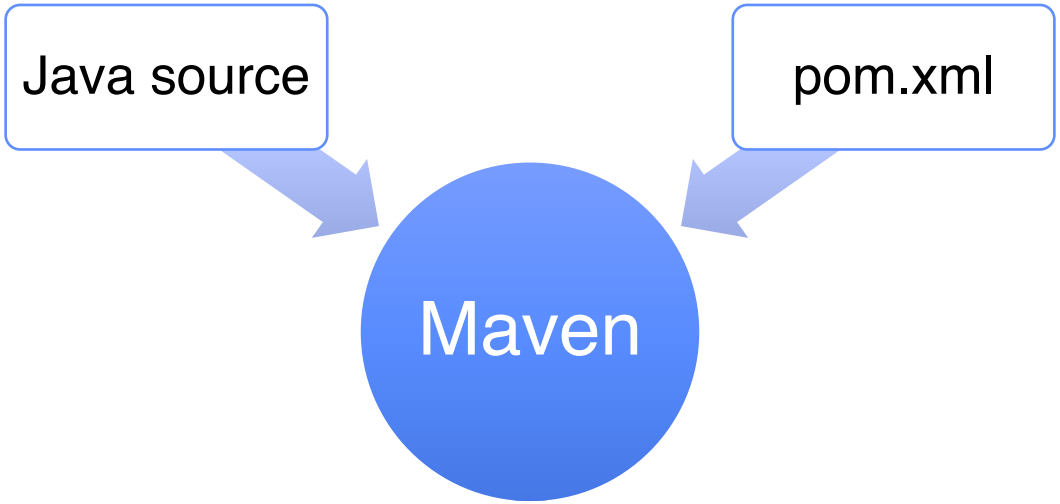
Tutorial: https://github.com/biojava/biojava-tutorial

Educational introduction into the tools provided by BioJava

Cookbook: http://biojava.org/wiki/BioJava:CookBook

Collection of “How do I…?” recipes for common tasks

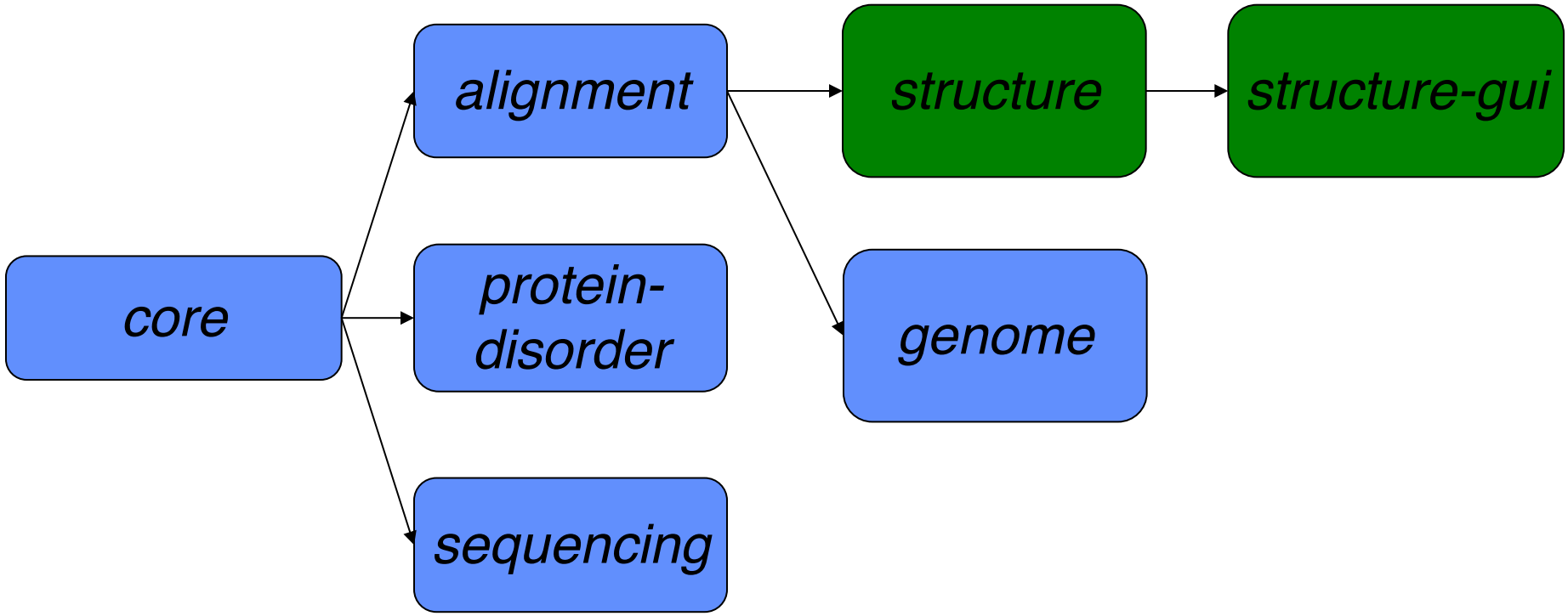
Setting up BioJava



- **Project Object Model (POM):** XML file that contains information about the project and configuration details

Setting up BioJava

- BioJava modules

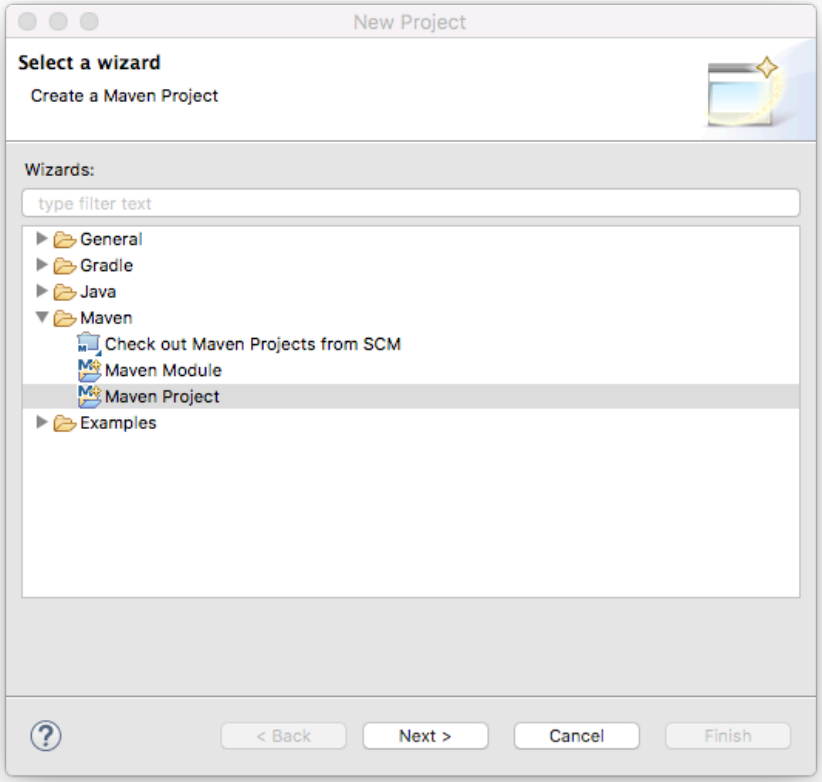


Demo 1

- Create a new Maven repository in Eclipse

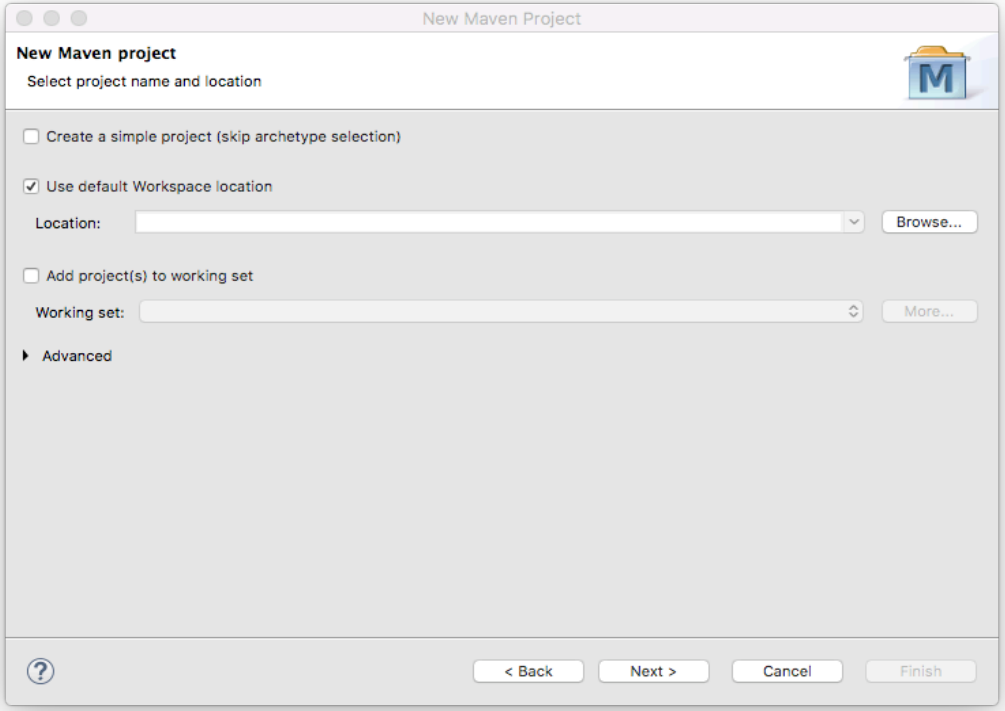
Demo 1

- **Create a new Maven repository in Eclipse**
  1. File > New > Project
  2. Maven Project



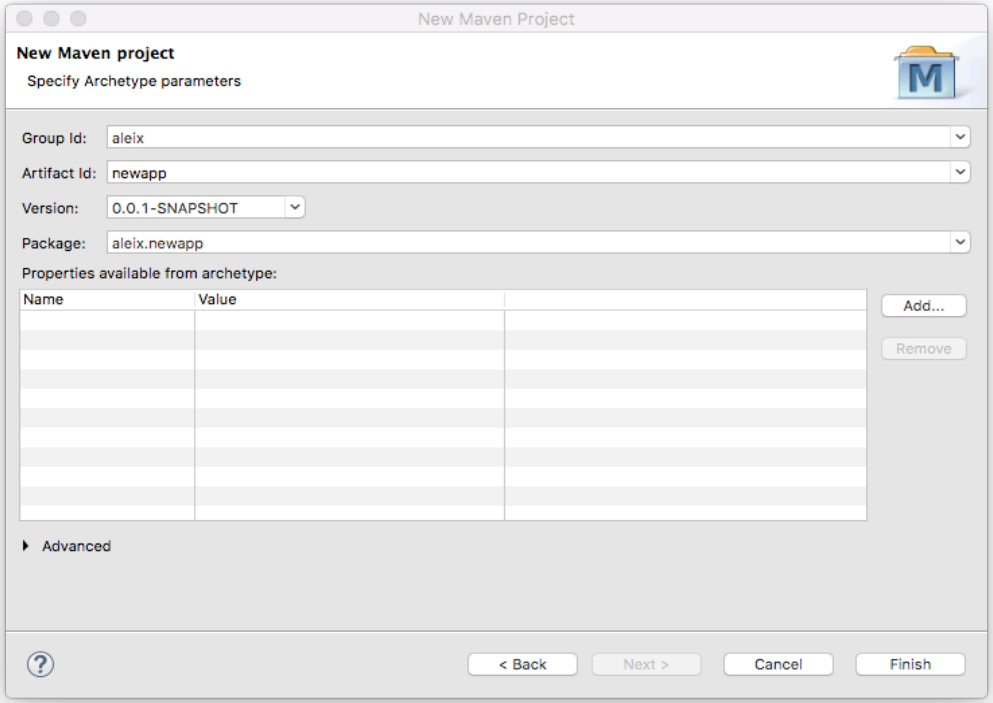
Demo 1

- **Create a new Maven repository in Eclipse**
  1. File > New > Project
  2. Maven Project
  3. Use defaults



Demo 1

- **Create a new Maven repository in Eclipse**
  1. File > New > Project
  2. Maven Project
  3. Use defaults
  4. Choose your group and project id



Problem 1

- **Configure the new Maven repository and add the BioJava dependency**

  - **Task 1**: fill in the URL field (replace the default URL)
  - **Task 2**: configure the Java 8 JDK. Insert these properties in the POM:

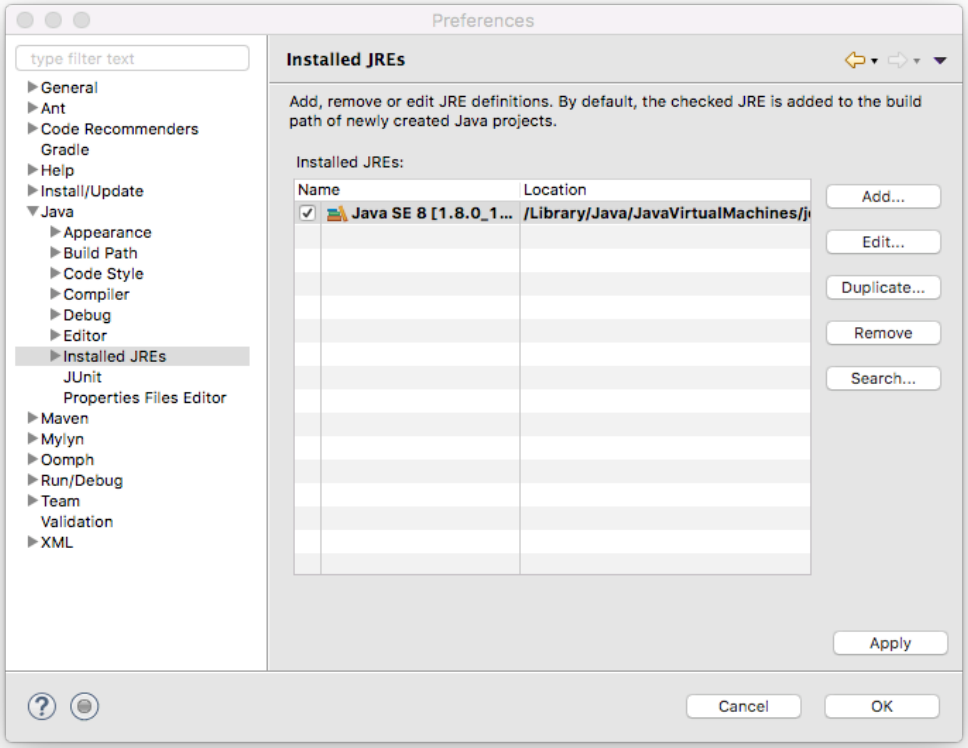
    <jdk.version>1.8</jdk.version>
    <maven.enforcer.jdk-version>1.8</maven.enforcer.jdk-version>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>

  - **Task 3**: insert the *biojava-stucture-gui* dependency (hint: https://github.com/biojava/biojava-tutorial/blob/master/installation.md)

  - **Task 4:** ensure BioJava version is the latest (5.0.0-alpha8)

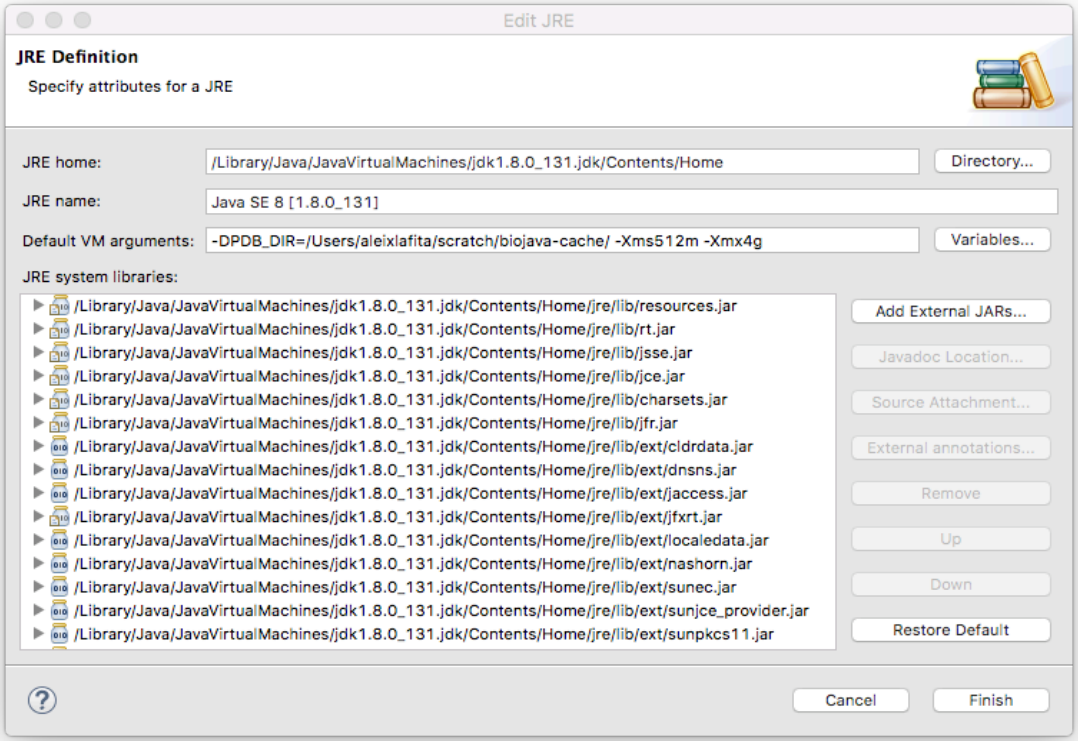
Demo 1

- **Configure the BioJava environment**
  1. Eclipse > Preferences > Java > Installed JREs
  2. Edit…



Demo 1

- **Configure the BioJava environment**
  1. Eclipse > Preferences > Java > Installed JREs
  2. Edit…
  3. Add -DPDB_DIR= directory



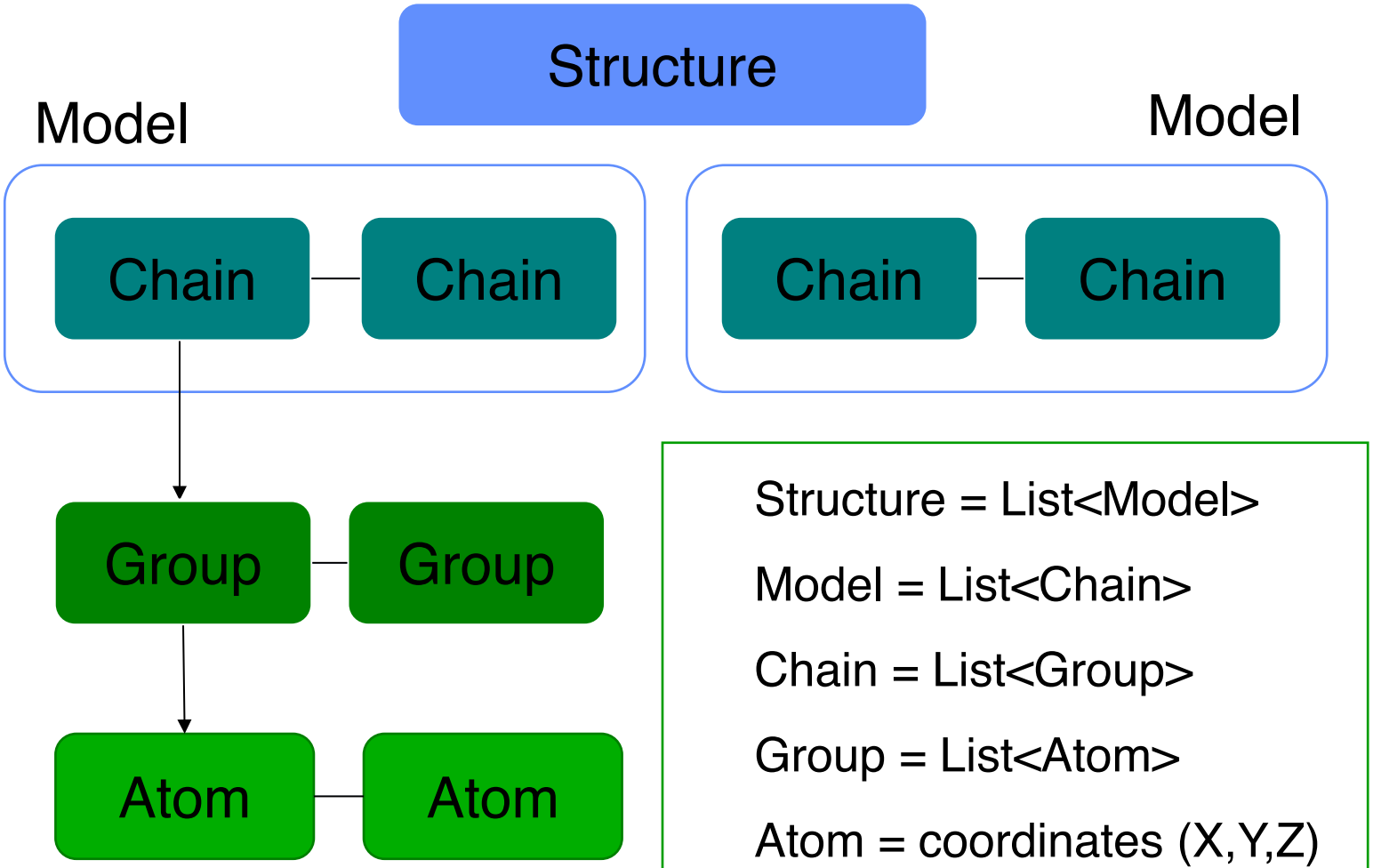
PART 2
BioJava Structures

Structure objects

Loading and writing

Operations

Structure Objects



Group Types

- **BioJava** uses the **Chemical Component Dictionary** to assign group types
  - Amino Acid (L-peptides)
  - Nucleotide (DNA or RNA)
  - Waters
  - Hetatoms

https://github.com/biojava/biojava-tutorial/blob/master/structure/chemcomp.md

Loading Structure Objects

- **AtomCache**
  - PDB ids (2HHB)
  - Structural ranges (2HHB.A:1-20)
  - SCOP, CATH & ECOD ids (d2hhba_)
  - Many configuration options
- **StructureIO**

Demo 2

- Load a structure in different formats
- Write a structure from BioJava to a file

Problem 2

- **Traverse BioJava structures**
  - **Task 1**: number of models. Hint: *nrModels*
  - **Task 2**: number of polymer chains. Hint: *PolyChains*
  - **Task 3**: number of amino acids in a chain.
    - Take a look at the *GroupType* class first
    - Hint: use one method from the Chain object
  - **Task 4**: number of oxygens in the amino acids of a chain.
    - Take a look at the *Element* class
    - Hint: use a double loop iteration over List of Groups and Atoms

PART 3
BioJava Applications

Biological assemblies
Visualization in Jmol

Biological Assemblies

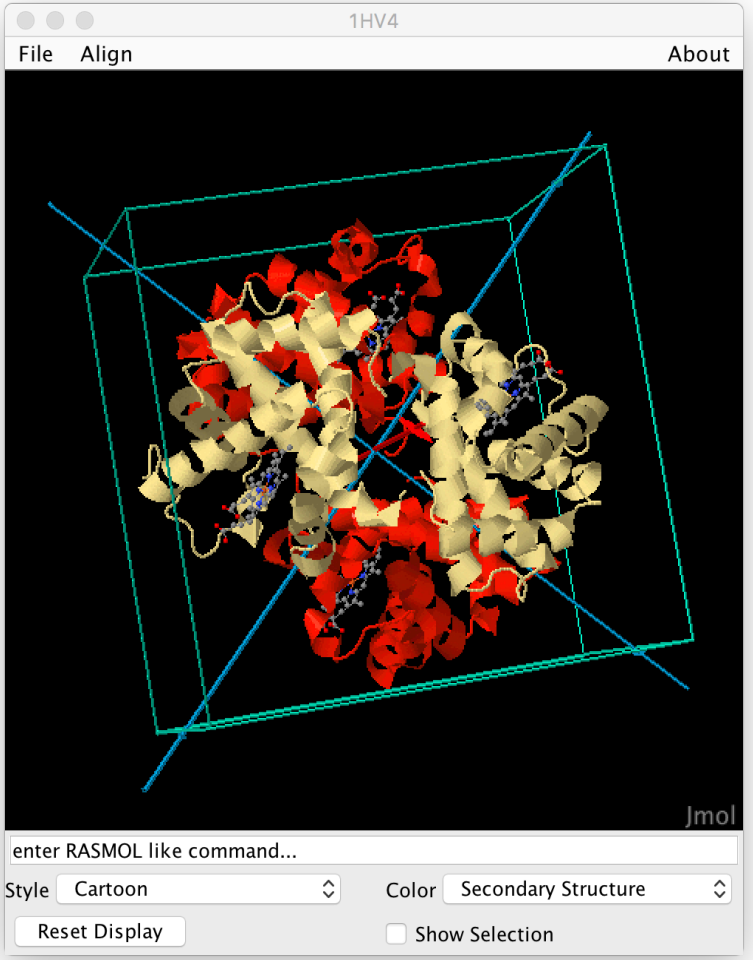
- **Important concepts**
  - Asymmetric Unit (AU)
  - Unit cell
  - Crystal lattice

- **Resource:**
  - https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/biological-assemblies

Visualization in Jmol

- Interface for **quick** and **simple** visualization of structures in **Jmol**

1. Start a **JFrame**
2. Insert a **Jmol** panel
3. Convert the **BioJava** structure to a structure file (PDB, MMTF)
4. Send the structure file to the **Jmol** panel



Problem 3

- **Display the symmetry of a biological assembly**
  - **Task 1**: download a biological assembly.
    - Hint: use the StructureIO class directly.
    - Warning: set the multiModel option to true (false is default) to be able to visualize the results (due to limitation of single letter chains in PDB format)
  - **Task 2**: change the parameters for pseudo-symmetry analysis.
    - Hint: structural clustering instead of sequence.
  - **Task 3**: obtain the symmetry and stoichiometry from the result.
    - Hint: use getter methods.

Thank you!

BioJava

Contributing

- **Do you want to use BioJava for your project?**
- **Do you have any ideas?**
- **Would you like to stay tuned?**

- **Join us at:**
  - GitHub issues: https://github.com/biojava/biojava/issues
  - Mailing list: http://biojava.org/wiki/BioJava%3AMailingLists
  - Gitter: https://gitter.im/biojava/biojava

BioJava