ALIDA–Automatic Generation of User Interfaces for Data Analysis Algorithms

Stefan Posch, Birgit Moeller

Institute of Computer Science, University of Halle

Germany

stefan.posch@informatik.uni-halle.de http://www.informatik.uni-halle.de/alida/

Abstract

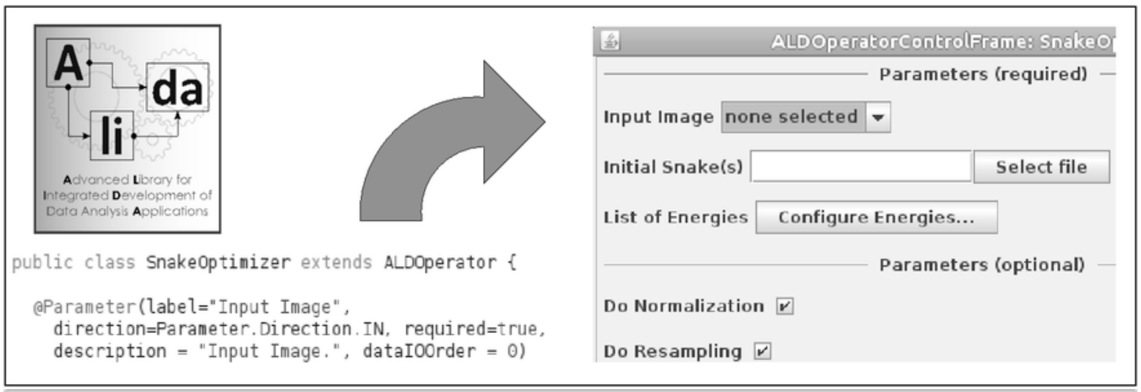
Analysis of biomedical data may be interpreted as a flow of objects through an analysis pipeline. The Java framework Alida defines the concept of operators as the single places of these manipulations. Typically, operators may be invoked sequentially or in parallel, and often also nested. Besides invocation on the programming level their functionality should also be available directly to users, including developers of algorithms and non-experts. This calls for graphical as well as command line interfaces. Eliminating the need to explicitly code these interfaces, Alida features fully automated generation of graphical and command line user interfaces for each operator implemented in the Alida framework.

The basis is a formalism for an operator to define all input and output data objects and parameters to control processing. Automatic generation of interfaces is based on the model view presenter design pattern to achieve maximal independence between the operators, interfaces, and I/O of data objects. For implementation Java's annotation mechanism is used. The programmer is only required to properly annotate classes and member variables. Out of the box this facilitates I/O for a wide variety of Java objects including primitive data types, enumerations, arrays, and collections. In a generic way Alida handles also operators as parameters of other operators and inheritance. Only specialized classes like images require additional data providers to be implemented.

While Alida is devised for data processing in general, it is used in our image analysis toolbox MiToBo (http://www.informatik.uni-halle.de/mitobo) for biomedical image analysis which is based on ImageJ and compatible to it.

Keywords

User interface, automatic generation, data analysis, ImageJ


A
da
li
Advanced Library for Integrated Development of Data Analysis Applications
ALDOperatorControlFrame: SnakeO
Select file
Configure Energies...

```
public class SnakeOptimizer extends ALDOperator {

  @Parameter(label="Input Image",
    direction=Parameter.Direction.IN, required=true,
    description = "Input Image.", dataIOOrder = 0)
```

Parameters (required)

Input Image none selected

Initial Snake(s)

List of Energies

Parameters (optional)

Do Normalization ✓

Do Resampling ✓