BicOverlapper

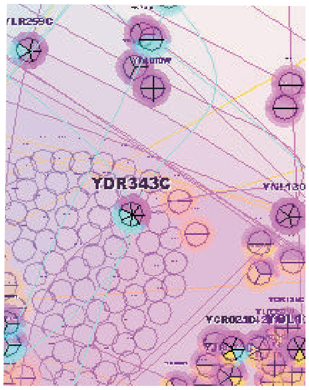
Developer's Guide

July 8, 2009

Rodrigo Santamaría, Roberto Therón and Luis Quintales
Departamento de Informática y Automática

Contents

List of Figures



Chapter 1

Overview

This document is aimed to java developers that want to use some or all of the BicOverlapper java classes. Therefore, this document contains information about the engineering, design and architecture of these classes. Further documentation about the classes is available as a javadoc document in the downloads section at http://code.google.com/p/bicoverlapper.

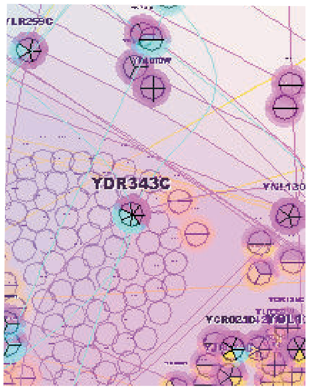
1.1 License

The software referred in this guide is distributed under MIT License:

1.2 BicOverlapper

BicOverlapper is a framework designed to analyze biclustering results by visualizing them and other biological information in different, linked displays. These displays and their linkage, along with other functionalities, have been implemented in Java under JDK 1.6.0, with the help of some additional packages as Prefuse [1], Substance and VectorGraphics.

The software has been developed by Rodrigo Santamaría and Roberto Therón, with the help of Javier Molpeceres, at the VisUsal group of the Department of Informatics and Automatics, University of Salamanca (Spain). For any question, suggestion or request for collaboration, please address to Rodrigo Santamaría at rodri@usal.es



Chapter 2
Architecture

2.1 External packages

Although the core of BicOverlapper is original code, it makes use of existing packages:

- *Prefuse*[1]: this toolkit provides a framework to develop information visualization displays that has been used in order to implement the heatmap, bubblemap and TRN graph visualizations.

- *JRI*[2]: the JRI allows to run R inside Java applications as a single thread. It is now distributed as part of *rJava* package for R. BicOverlapper uses this library to request for applications available in R and BioConductor, specially the packages *biclust*, *annotate* and *GOstats*. Also it requests for other in-house R codes integrated in the tool.

- *Axis*[3]: is an implementation of the SOAP submission to W3C that allows the exchange of structured information in a distributed environment. It was used to request for different biological information to several repositories, although the current version only uses it to support eUtils.

- *NCBI eUtils*[4]: is a set of tools to interact with NCBI resources, BicOverlapper makes used of it to query *Entrez Gene* database.

- *Substance*[5]: is a library that provides visually appealing swing interfaces. BicOverlapper external interface style is one of the several skins that this library offers.

[1] *http://prefuse.org*
[2] *http://rosuda.org/JRI*
[3] *http://ws.apache.org/axis*
[4] *http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html*
[5] *https://substance.dev.java.net*

- *FreeHEP*[6]: is a library developed to encourage the sharing and reuse of Java code in HEP. BicOverlapper use of this library is obviously marginal, focused in its subsection VectorGraphics to export vectorial snapshots of Overlapper.

- *SplineFactory*[7]: this single class, developed by Jacob Dreyer to generate splines, is also used by Overlapper to draw circular shapes (hulls).

In addition, *Processing*[8] was used in the prototype of Overlapper, but was previously substituted for an internal class (*JProcessingPanel*) that emulates most of the Processing functionality. Also, *Gishur*[9] class library was previously used to implement some geometric algorithms, and will be possibly used again in the future, specially in the Overlapper visualization. Finally, *QuickGO*[10] web service is used in order to request for GO terms.

2.2 BicOverlapper structure

BicOverlapper has been designed focused in two main entities: *data* and *visualizations*. Both entities are related, so a kind of data is visualized by one or more visualizations, and viceversa. To control the communication between data and visualizations, a third entity, the *kernel*, is introduced. This kernel manages the vertical (data to visualization or visualization to data) communication and the horizontal (between visualizations or between data) communication (see Fig. 2.1).

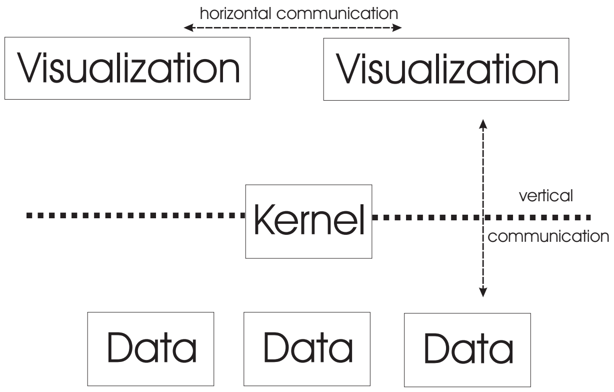


Figure 2.1. *BicOverlapper structure is based in tree entities: data, visualization and kernel. Data and Visualization can have multiple instances, while only one instance of the kernel is at each moment, linking visualization and data to allow communication between visualizations or between data (horizontal communication), and between visualizations and data (vertical communication).*

[6] *http://java.freehep.org*

[7] *http://geosoft.no/software/spline/docs/no/geosoft/cc/ geometry/spline/SplineFactory.html*

[8] *http://processing.org*

[9] *http://www.lupinho.de/gishur.html*

[10] *http://www.ebi.ac.uk/QuickGO*

Some of the implemented communications include:

- *Data to visualization*: tell the visualization that a new set of data has been loaded, or that there are data loaded.
- *Visualization to data*: tell the data that some of the data has been selected, filtered or searched by the user's interaction.
- *Visualization to visualization*: inform or detect the selection or filtering of linked elements.

Finally, two other packages act as ancillary repositories:

- *utils*: contains utility classes for the main entities
- *resources*: contains other utilities as images, temporary files, etc

2.3 BicOverlapper packages

The detailed package structure is shown at Fig 2.2. Here is described each of the packages and, briefly, the classes they contain. For full description of attributes and methods, please refer to the javadoc. All the package names start with *es.usal.bicoverlapper*, following java package naming.

2.3.1 es.usal.bicoverlapper.kernel

Kernel package contains the fundamental classes for BicOverlapper: the main class, the main window and desktop classes and the data/visualization communication classes. Additionally, it contains packages *kernel.managers* with the classes that manage menu options of the main window and *kernel.configuration*, that have the configuration management classes.

BicOverlapper

Main class of the program, it just sets the Substance look and feel and opens a new *BicOverlapperWindow*.

BicOverlapperWindow

Main window, with a desktop in which diagrams will be added (see 2.3.6 and a menu bar.

BiclusterSelection

This important class maintains the current selection for the biological entities. Designed as a bicluster, that is, a set of genes and conditions, you can use it to select only genes, only conditions or both of them (biclusters).

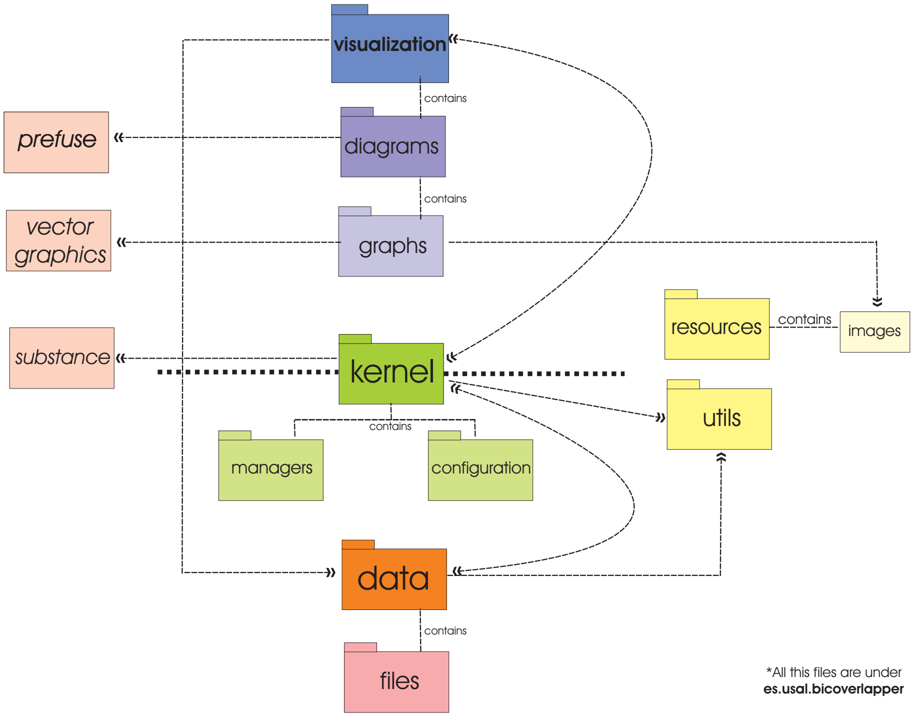


Figure 2.2. *BicOverlapper package structure. At the left, the external packages used. At the center, the main package kernel that communicates visualization (top) and data (bottom). At the left, two ancillary packages. An arrow pointing form* a *to* b *means that* a *uses classes contained in* b*. Non arrowed lines are containment relationships.*

Configuration

This class only contains field with information of default sizes and positions of diagrams, and is to be deprecated.

DiagramWindow

A *JInternalFrame* in which Diagrams are to be embedded.

Session

One of the most important classes of the *BicOverlapper*, the Session class establishes a *session layer* where information about the data loaded, diagrams visualized and interactions done in diagrams is stored and distributed between data and diagrams, in the form of *BiclusterSelection* and *TupleSelection*.

TupleSelection

This important class maintains the current selection for the general purpose entities, that is, multidimensional variables or tuples.

WorkDesktop

JDesktopPanel for the *BicOverlapperWindow*, containing the session instance or session layer

2.3.2 es.usal.bicoverlapper.kernel.managers

Each one of the classes in the managers (*ConfigurationMenuManager, FileMenuManager, HelpMenuManager and ViewMenuManager*) are just *ActionListeners* for each of the corresponding menu sections.

2.3.3 es.usal.bicoverlapper.kernel.configuration

This package contains all the classes that deal with the configuration of *Diagrams*. By now, configuration includes:

- Color, position and size configuration for each *Diagram*
- Loading/saving configurations for the *Diagrams*

DiagramConfiguration

Stores the configuration for a Diagram (position, dimension and *Vector* of *Colors*)

ConfigurationLoader

Reads and writes configuration files. Configuration files are simple xml files with configuration data for each diagram.

ConfigurationHandler

Just a *Vector* of *DiagramConfigurations*. To be deprecated.

2.3.4 es.usal.bicoverlapper.data

This package contains all classes that store the data that feed the application. There is a class for each kind of data with which the application can deal (TRNData, MicroarrayData and BubbleData), along with some other minor classes. In addition, some classes to be deprecated or deeply restructured are in the package: MultidimensionalData and DataLayer. These classes deal with generic multidimensional data

By now, these data classes are independent, but in the future they will be unified so all the shared characteristics will be inherited from a generic abstract class (probably, MultidimensionalData). This package also contains *data.files*, the package that deals with file reading and writing, following the specified formats.

Bubble

This class contains all the information to represent a Bubble for the bubble map, as public fields (position, size, homogeneity, etc.)

BubbleData

This is primarily a builder of Prefuse Tables to represent bubbles, doing the projections of biclusters in 2D to position them.

DataLayer

To be deprecated class that stores *MultidimensionalData* and *Session* information.

DataUtils

Some useful generic functions to basic statistical functions as average, deviation and normalization.

Field

A Field or dimension in a multidimensional variable. It maintains an internal *Vector* with values of each variable or tuple for this dimension.

MicroarrayData

This is primarily a builder of Prefuse Tables to represent a microarray data matrix.

MultidimensionalData

Basic class to maintain multidimensional data, as a *Vector* of *Fields*.

TRNData

This is primarily a builder of Prefuse Graphs to represent a Transcription Network. It also do some motifs computations (Fast Forward Loops), although this is not reflected in final representation by now.

2.3.5 es.usal.data.files

The classes in *files* are:

- *Structures*: read/write in delimited files
- *Parsers*: convert and read existing file formats

- *Filters*: File Filters to check file extensions
- *Managers*: read/write in delimited files (to be deprecated and fused with Structures)

GenericFileStructure

Simple interface for reading and writing in files

FileStructure

Implements *GenericFileStructure* to read and write from delimited files.

TRNFilter and MicroarrayFilter

FileFilter extensions to deal with TRN (*TRNFilter*) and Microarray (*MicroarrayFilter*) file extensions.

SimpleFileManager

This class just converts a file to a *LinkedList* with the lines in the file or a *LinkedList* of strings to a file

CompleteFileManager

Extension of *SimpeFileManager* that allows to read delimited fields from a file converted in a *LinkedList*

FileParser

Singleton for file parsing utilities. By now, only conversion from BicOverlapper bicluster format to BiVoc [2] format.

TRNParser

Converts SynTReN xml format for transcription networks to GML. In the future, more formats will be supported.

2.3.6 es.usal.bicoverlapper.visualization

visualization contains all the swing JPanels that maintain the different visualizations and communicates with the kernel. These JPanels are called *diagrams*, all of them inherit from an abstract class Diagram and there is one for each visualization. Some of the diagrams use Prefuse architecture. *visualization.diagrams* contains these diagrams, while *visualization.diagrams.graphs* has additional classes used by the Overlapper graph implementation.

2.3.7 es.usal.bicoverlapper.visualization.diagrams

This package is formed by the class *Diagram* and their children. *Diagram* is basically a *JPanel* with linkage functionality, that is, information of the active *Session* and communication functions. *BubblesDiagram, HeatmapDiagram, ParallelCoordinatesDiagram, TRNDiagram* and *OverlapperDiagram* are extensions of *Diagram* with particularities for each view, such as size and color configuration.

2.3.8 es.usal.bicoverlapper.visualization.overlapper

This is probably the most important package for BicOverlapper, since it contains the classes that implement the Overlapper visualization. Based on a Processing prototype, it still follows some of Processing conventions for drawing and animation.

Cluster

A *Cluster* is a group of *Nodes*. Because nodes can be genes or conditions, a Cluster truly can represent any kind of bicluster, including, as special cases, clusters and individual genes or conditions. This class contains just drawing functions, no connection methods between the nodes in the Cluster.

ClusterSet

As its name says, this is just a set of *Cluster*. It's useful to group all the clusters returned by an specific clustering algorithm.

Edge

Basic connection between two nodes. Although *Overlapper* just deals with undirected graphs, *Edge* class are directed.

ForcedNode

Node extension that implements Forces. Note that forces at *ForcedNode* are not real vector forces, but the position at which the node should be placed if the force is applied. True vectors are computed at *Overlapper*.

Graph

This class holds the structure visualized by *Overlapper*. It maintains *Map* structures for the nodes, edges, and cluster sets.

Handle

Deprecated class to handle forces.

JProcessingPanel

JComponent that emulates most of the Processing functions: drawing, mouse and keyboard input, files and transformations.

MaximalCluster

Cluster that builds edges amongst all the nodes in it. Note that edges are not maintained in the *MaximalCluster*, but in the *Graph* to which it is linked.

Node

Basic element of a *Graph*, it maintains all the basic information for a node: position, size, label, type (gene or condition). It also has information about the clusters in which it is in and the neighbor nodes.

Overlapper

A *JProcessingPanel* that implements the Overlapper Visualization. It contains a *Graph* instance, managing the application of forces and the user's interaction.

RadialCluster

Cluster that implements a central dummy Node, to which all other nodes are connected. Note that edges are not maintained in the *RadialCluster*, but in the *Graph* to which it is linked.

SpringEdge

SpringEdge implements elastic edges. Elastic edges tend to a natural length and apply an (attractive or repulsive) lineal force to the nodes it connects.

2.3.9 es.usal.bicoverlapper.utils

This package is just a repository of utility classes for different purposes: color management, array manipulation, translation, url checking, etc.

2.3.10 es.usal.bicoverlapper.resources

This package is designed to include all non-class files used by the application, that is, image files for buttons and icons and temporary text files. To avoid dispersion, all new images or text files must be included here.

Bibliography

[1] J. Heer, S. K. Card, and J. A. Landay, “prefuse: a toolkit for interactive information visualization,” in *Proceedings of SIGCHI Human Factors in Computing Systems*, 421–430, ACM Press, (New York, NY, USA) (2005).

[2] G. A. Grothaus, A. Mufti, and T. Murali, “Automatic layout and visualization of biclusters,” *Algorithms for Molecular Biology* **1**(15) (2006).