Table of Contents

release notes 0.9

Visualea

Packages

New Packages

Openalea.Image

Openalea.Numpy

OpenAlea.PkgBuilder

Openalea.Pylab
Updated Packages

OpenAlea.Grapheditor

vplants.stat_tool

vplants.sequence_analysis

release notes 0.9

Here is a small summary of changes to find in the brand-new release of OpenAlea and its companions : VPlants and Alinea.

Visualea

Even though Visualea hasn't undergone a huge face lift, some useful changes must be pointed out.

Per-CompositeNode evaluator

In previous versions of OpenAlea, the evaluation method was set application-wide, inducing many side-effects when moving from a dataflow using evaluation A to another using evaluation B. Now it is set per-dataflow using the right-click context menu.

Memo-like annotations

Annotations now have by default the aspect of a memo. Annotations created before 0.9 still look as they used to but newly created annotations use the new look by default.

Additionally, they can be resized (using the handle in the bottom-right hand corner). Both the colour of the text and of the memo itself can be changed (using the vanishing toolbar that appears as you hover over the memo). Annotations can be promoted/demoted to the new or old style using the contextual menu (a bit tricky to obtain: aim the bottom edge of the note).

Coloured ports

To help figure out which port needs to be plugged with which other port, these are now coloured according to their Interface.

Search results with origin

The search widget now displays the package to which the item belongs. This makes finding the right item a bit easier.

Help widget

Now, if you click on an item from the Package Manager, or on a node in the dataflow, the Help widget will try to display interesting information about the selected element. This is a first implementation of the concept an will surely evolve in the future since we found it useful to have!

Packages

New Packages

Openalea.Image

This first release of the Openalea.Image package provides basic tools to manipulate images, including tools to display, place landmarks and process them. The images are represented as Numpy arrays and can be further processed using the Openalea.Numpy package.

See the online documentation for user and reference guide as well as a gallery of examples.

Openalea.Numpy

This first release of the Openalea.Numpy package wraps Numpy and exposes it to Openalea's dataflow programming interface making it easy to efficiently manipulate matrices through Openalea.

See the online documentation for user and reference guide as well as a gallery of examples.

OpenAlea.PkgBuilder

Create automatically an OpenAlea package for Python, C, or C++ with setup.py, scons files, documentation, ?

```
alea_create_package --name MyPackage --project openalea --languages cpp
```

See the documentation for further details.

Openalea.Pylab

This release provides a VisuAlea interface to Pylab/Matplotlib functionalities including most of the 2D plotting functions, some examples of 3D plotting functions, and many more related to axes/figure manipulation, patches ,drawing, ?.

See the online documentation for user and reference guide as well as a gallery of examples.

Updated Packages

OpenAlea.Grapheditor

This release features a major refactoring of the GraphEditor package to clean-up some basic concepts that weren't clearly defined in the first version. Most noticeably:

Vertices and connectors are now different entities. Edges connect to connectors. Vertices can have one or many connectors (there's a default invisible connector on each vertex that can be disabled)
This new version adds preliminary support for indirect communication between graph model elements and graph view elements through the graph structure. Previous version enforced that every entity of the view had it's openalea.core.observer.Observed-derived model counterpart and communication between both was direct (*i.e.* a graphical vertex needed an explicit model vertex, a graphical port needed a model port, even though the graph they belong to is already an observed graph).
Many bug fixes.

vplants.stat_tool

This release completes the wrapping of the stat_tool functionalities written in C++ into a complete python package that includes

Pylab plots for various data viewpoints
A python syntax identical to the original AML syntax
An on-line documentation (still in progress)
A complete test suite

See online documentation for further details

vplants.sequence_analysis

This release completes the wrapping of the sequence analysis functionalities written in C++ into a complete python package that includes

pylab plots for various data viewpoints
A python syntax identical to the original AML syntax
An online documentation (still in progress)
A complete test suite

See online documentation for further details about change log and usage.