

Reproducibility and OpenAlea

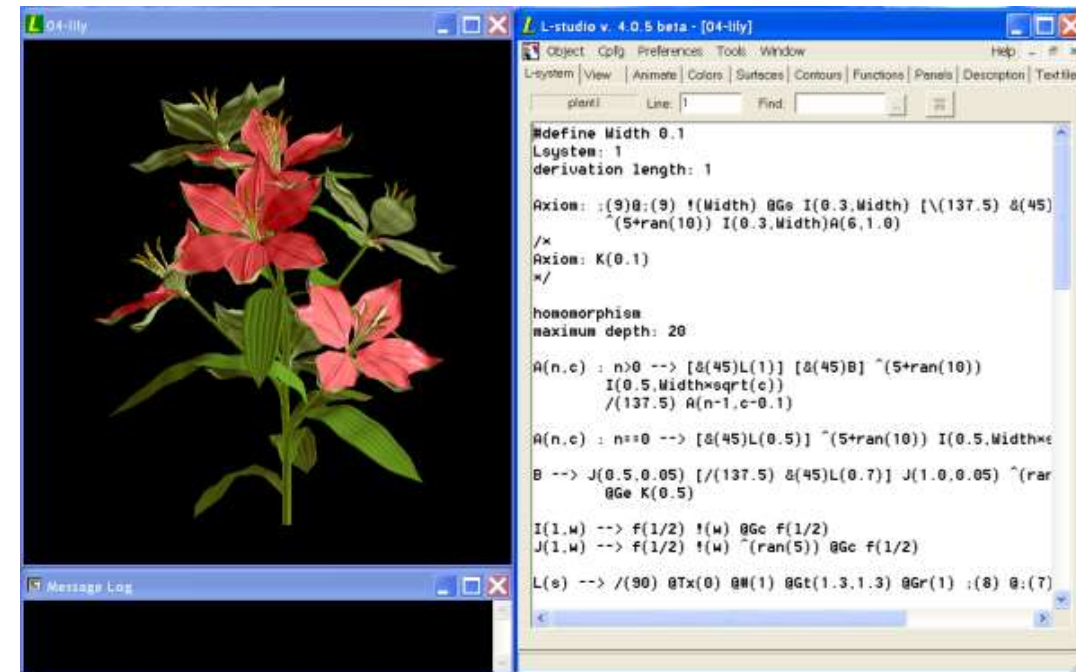
Jérôme Chopard, Christophe Pradal

Grenoble, 2016 December 6th

OpenAlea community



- Many ways to convince peoples to share some tools:
 - Your tool is the best on the market 😊
 - Users are captive:
 - hardware (e.g. iPhone)
 - Software (e.g. L-systems)
- General rule: everybody feels like there is a gain in sharing but nobody is willing to embrace someone else techno.
- Who's going to use it?



Plant modeling community

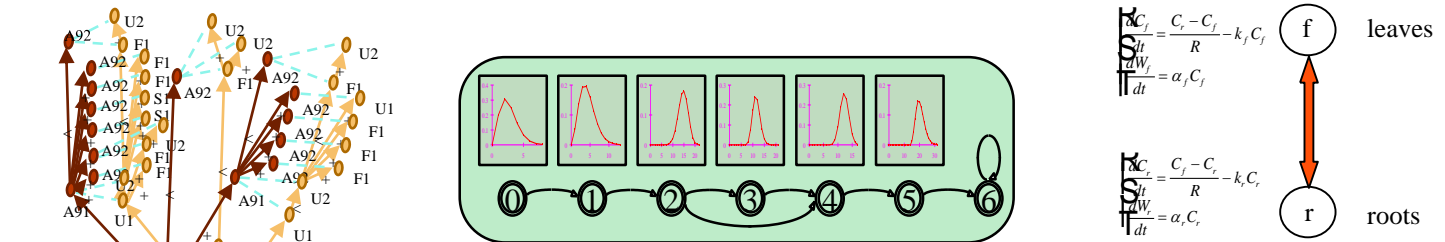
Biological
objects



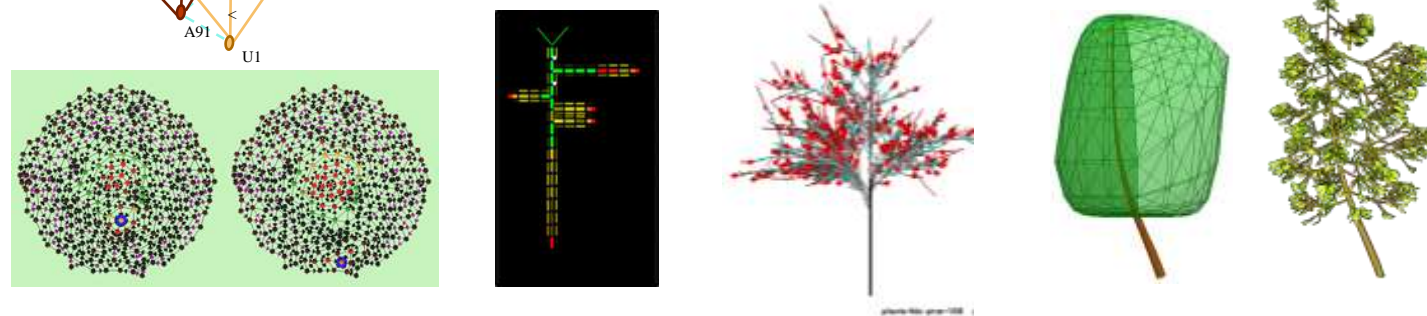
Mesure

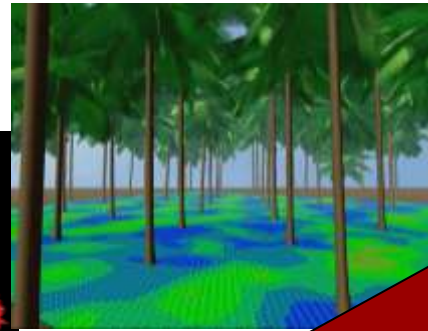
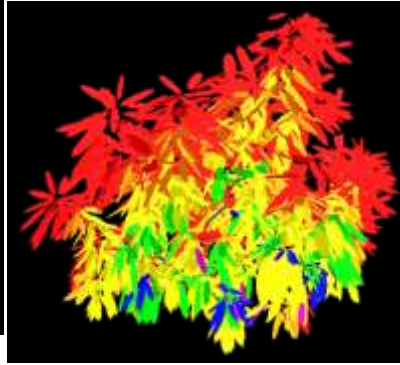
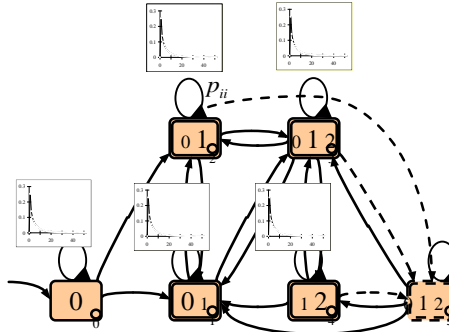


Modeling



Simulation





time & complexity of biological problems

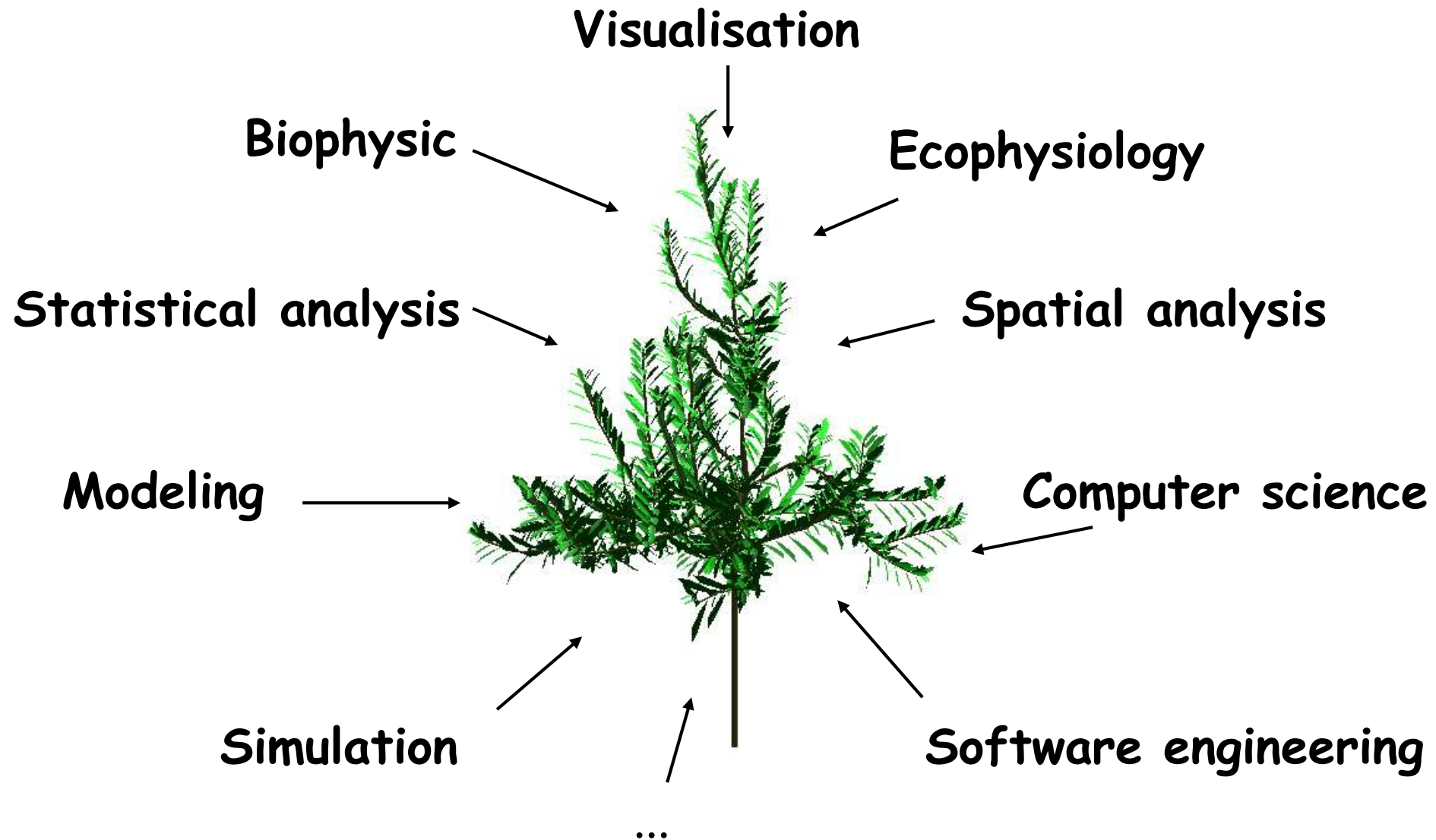
Monolithic Software

Interoperable
Software
Components
for modellers

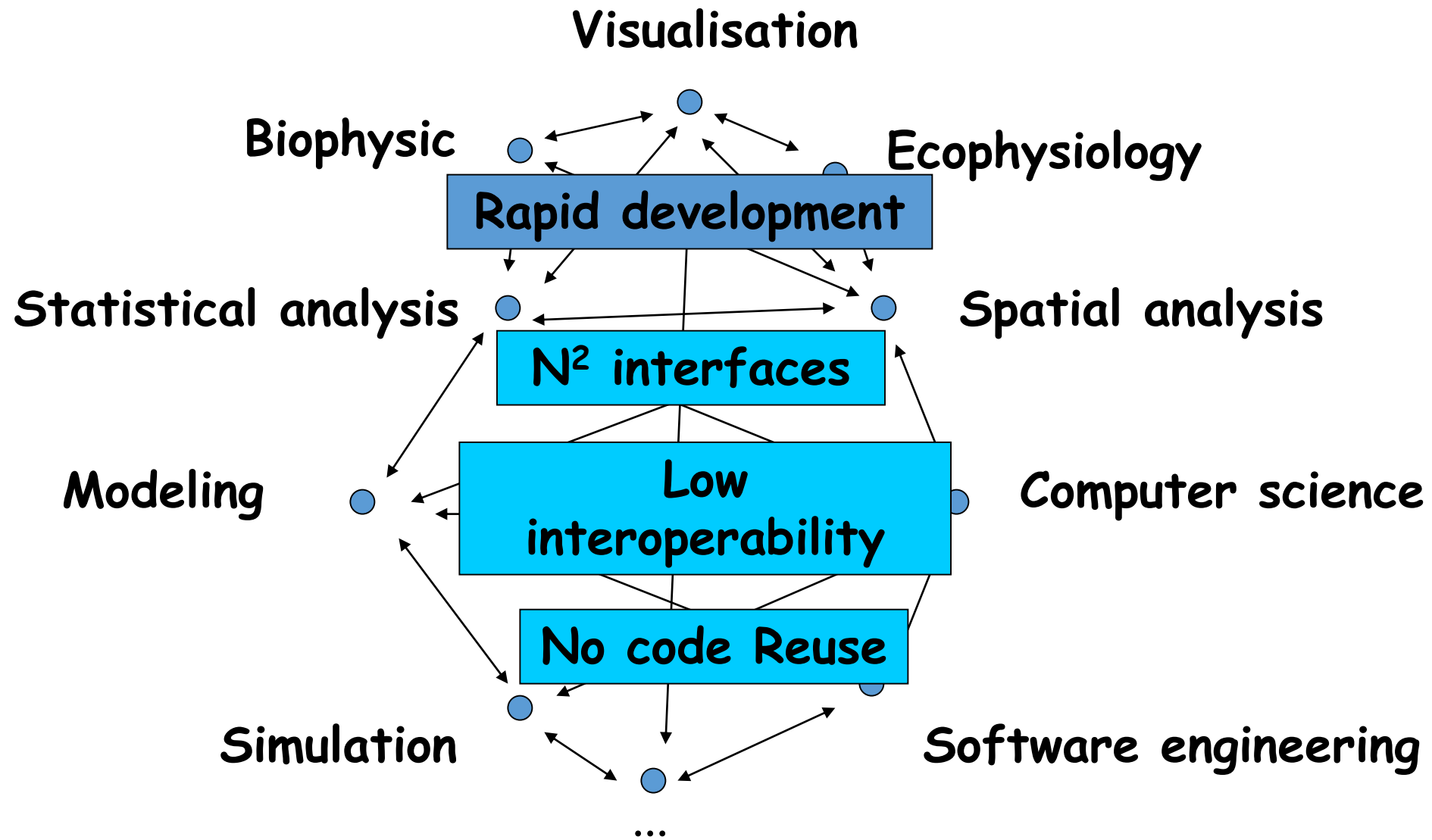
Interoperable
Software Components
for programmers

Software Components
for programmers

Challenge – Software reuse

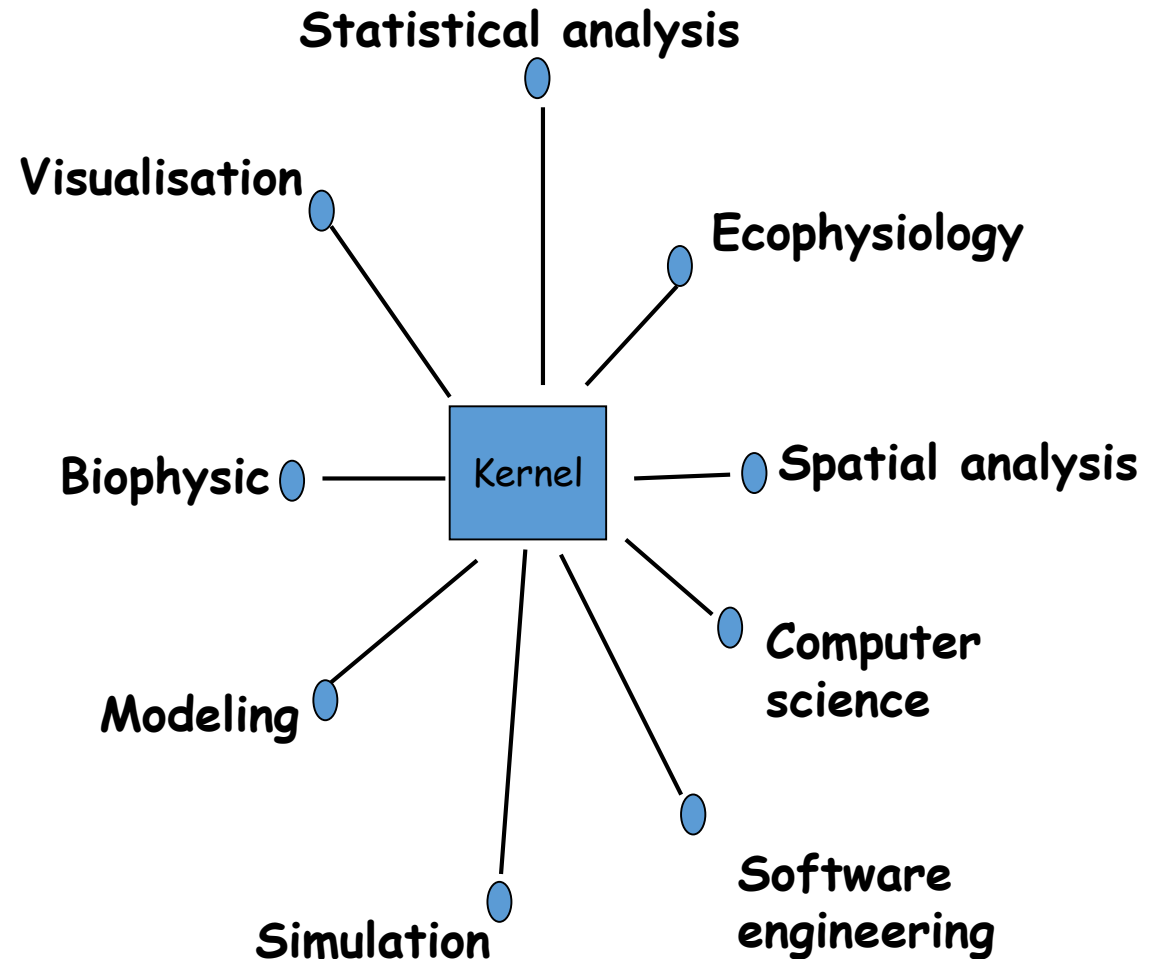


« Traditional » solution



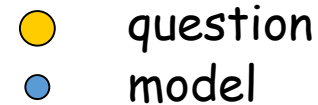
« Application » centric

- + Code Reuse by modularity
Object oriented technics
- « self-centric »: data,
types,API
Low interoperability between
applications



Modelling strategy

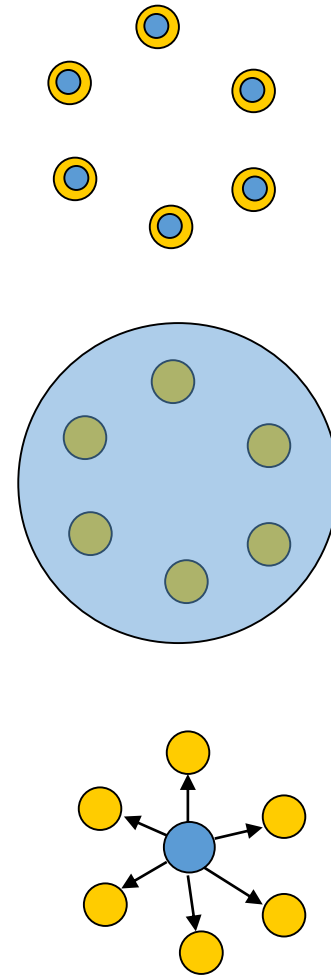
1. Construct the best model (efficient & simple) for each new situation



2. A general unified model

3. Defining common phenomenon, concepts and methods:

- Common to different situations
- Extensible



OpenAlea Goals



- OpenAlea (started in 2000)
 - A platform for plant modeling at different scales.
 - An initiative to share knowledge within FSPM community
- Sharing knowledge
 - Reuse software & tools
 - Share development between various teams
 - Share databases & training effort
- Common software platform
 - Integration of existing models & tools
 - Rapid development of new models
 - Enhance accessibility (GUI)
 - Quality rules

What is Free Software?

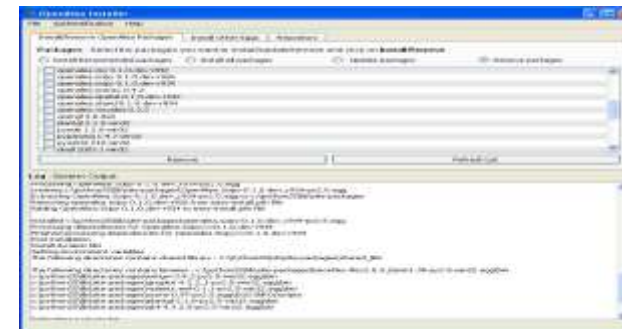
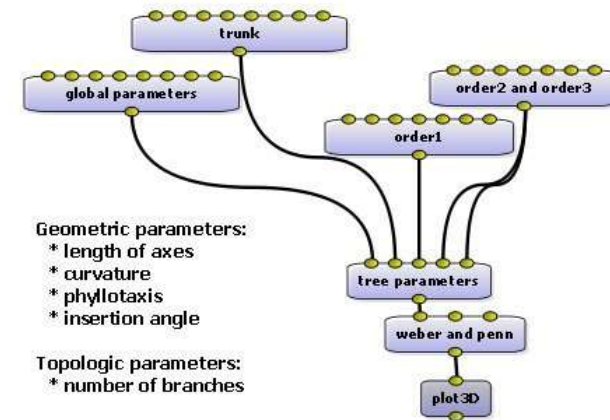


Christophe Pradal

- 1) The freedom to run the program, for any purpose.
- 2) The freedom to study how the program works, and adapt it to your needs.
- 3) The freedom to redistribute copies so you can help your neighbor.
- 4) The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

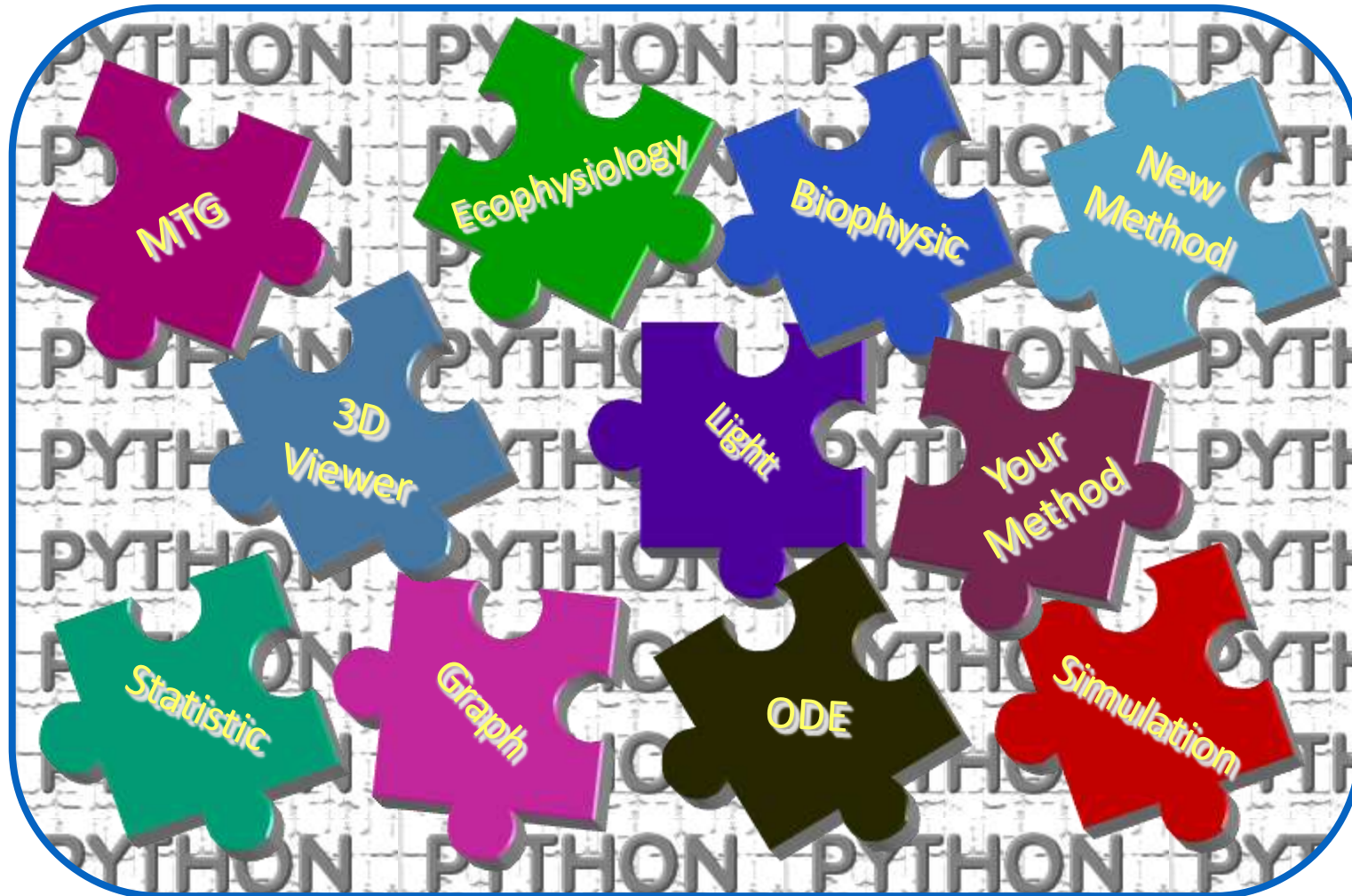
Design Principles

- Language centric
 - Common modeling language
 - Glue language
- Component architecture
 - Dynamic composition
 - High-level dataflow approach
- Visual programming
 - Graphical model representation
 - Automatic GUI generation
- Shared deployment tools
 - Build, packaging, distribution, installation, upgrade
- Multi-platform (Linux, Windows)



Language centric

High level language as a scripting environment

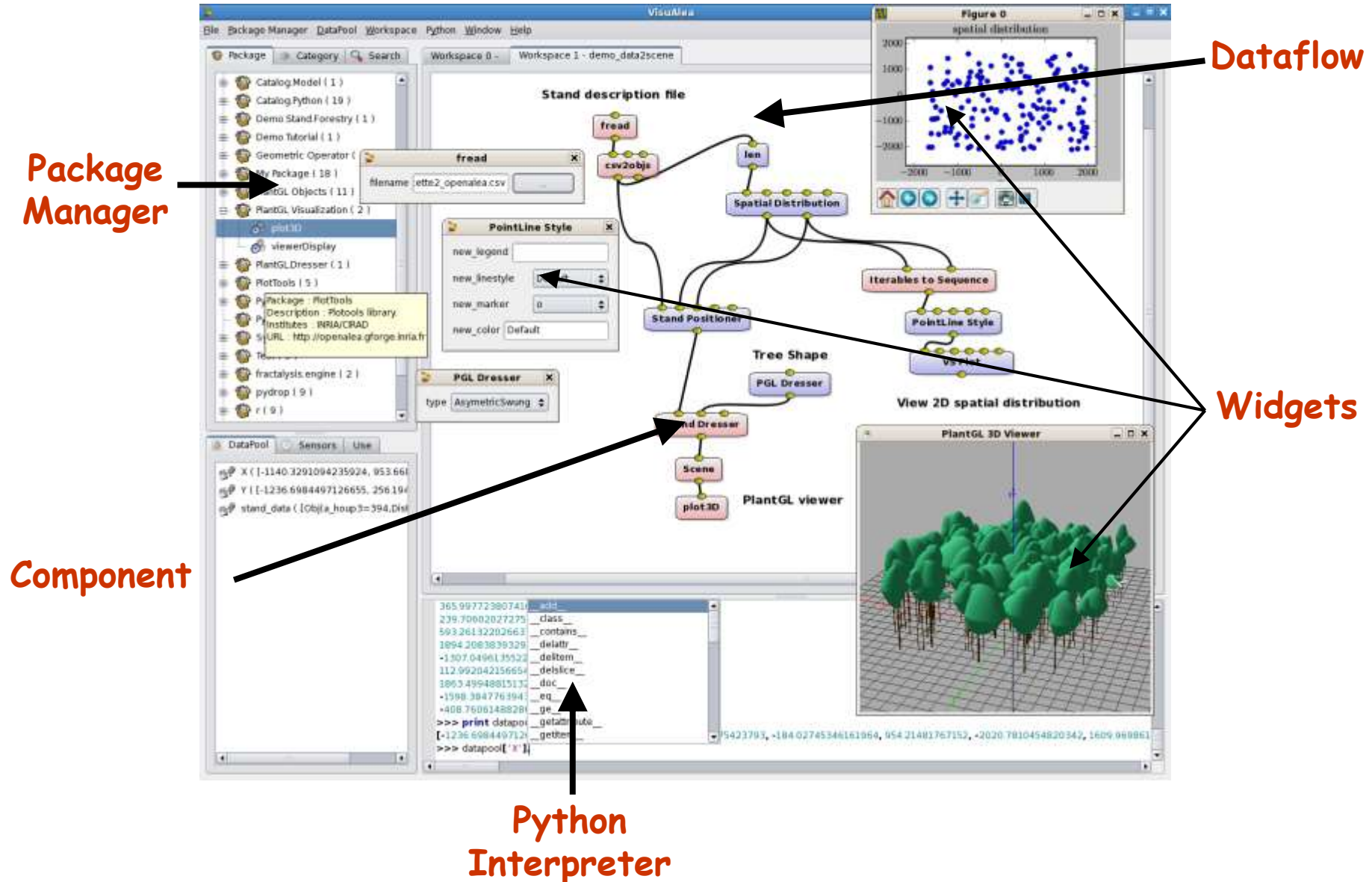


Interactive

Dynamic

**Platform
independent**

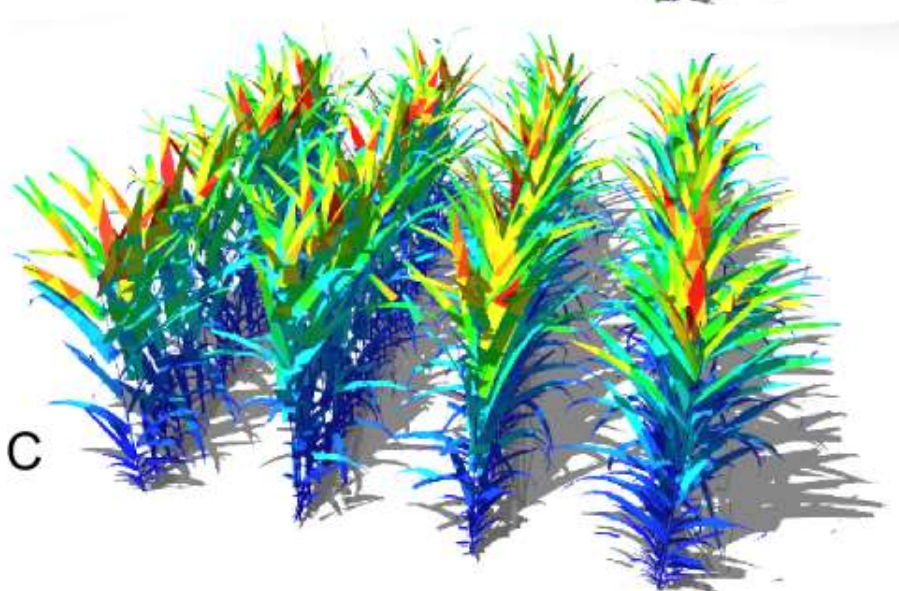
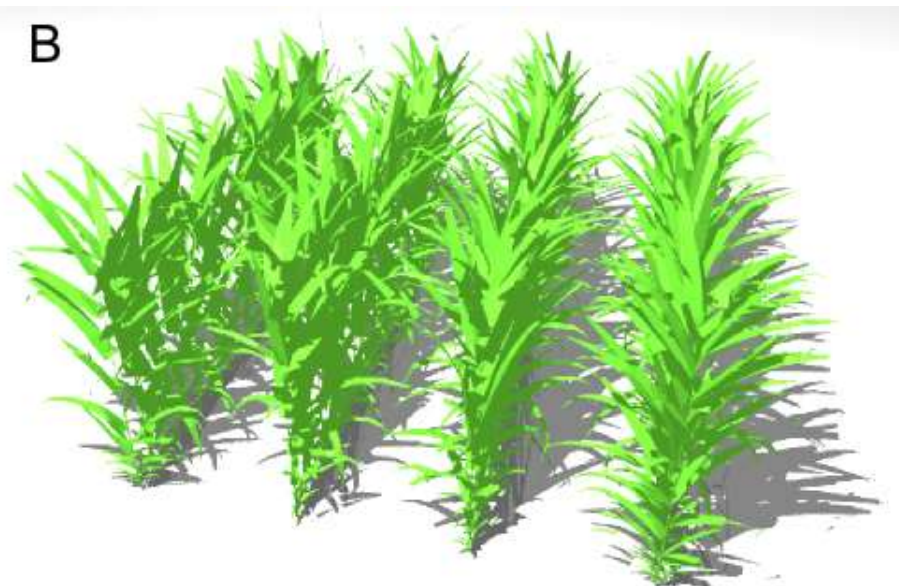
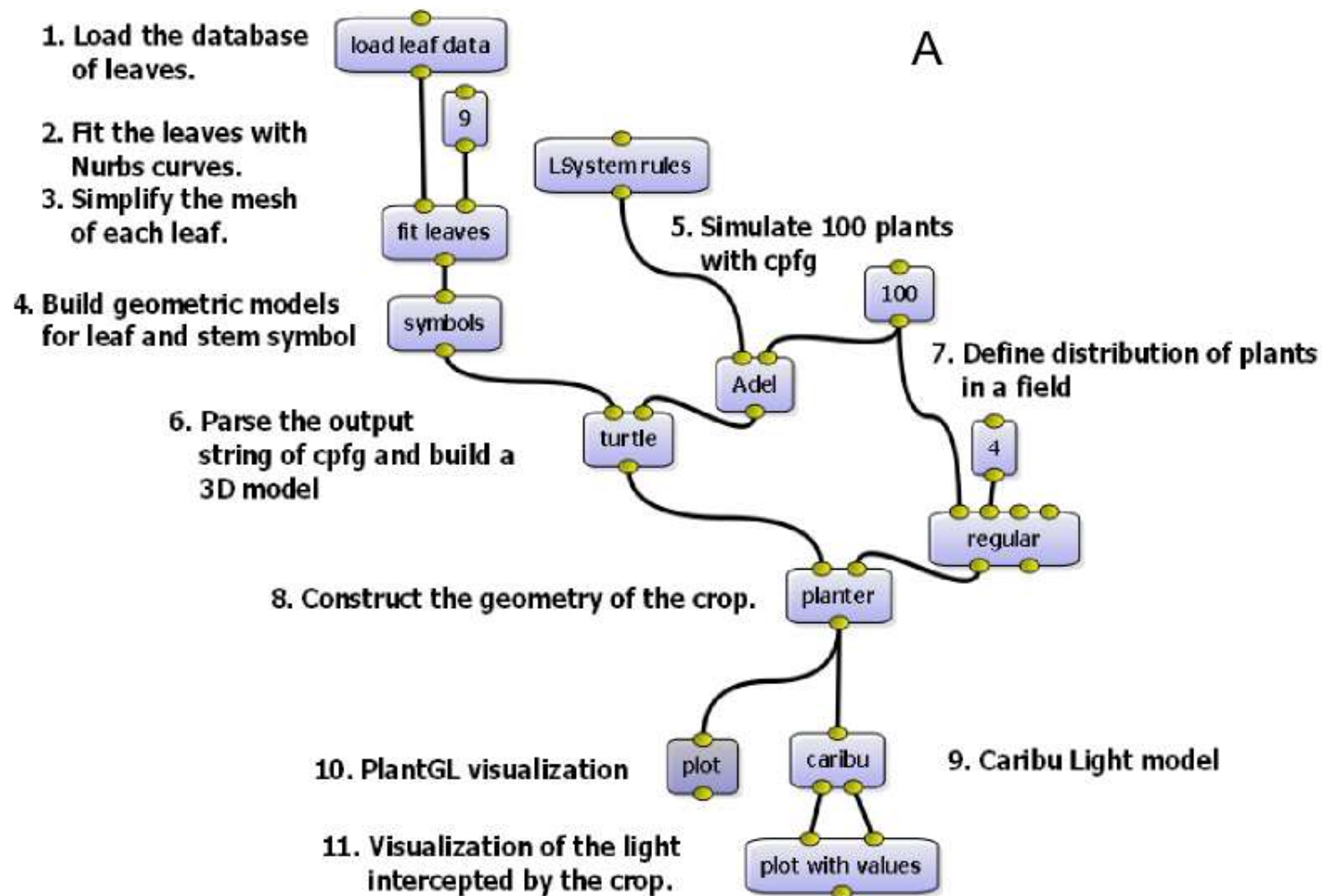
Component framework



OpenAlea workflows

- Presentation (demo) of OpenAlea tools: VisuAlea
 - Search for a node
 - Construct a workflow
 - Use of composite nodes
 - Create a node
 - Add a comment in workflow

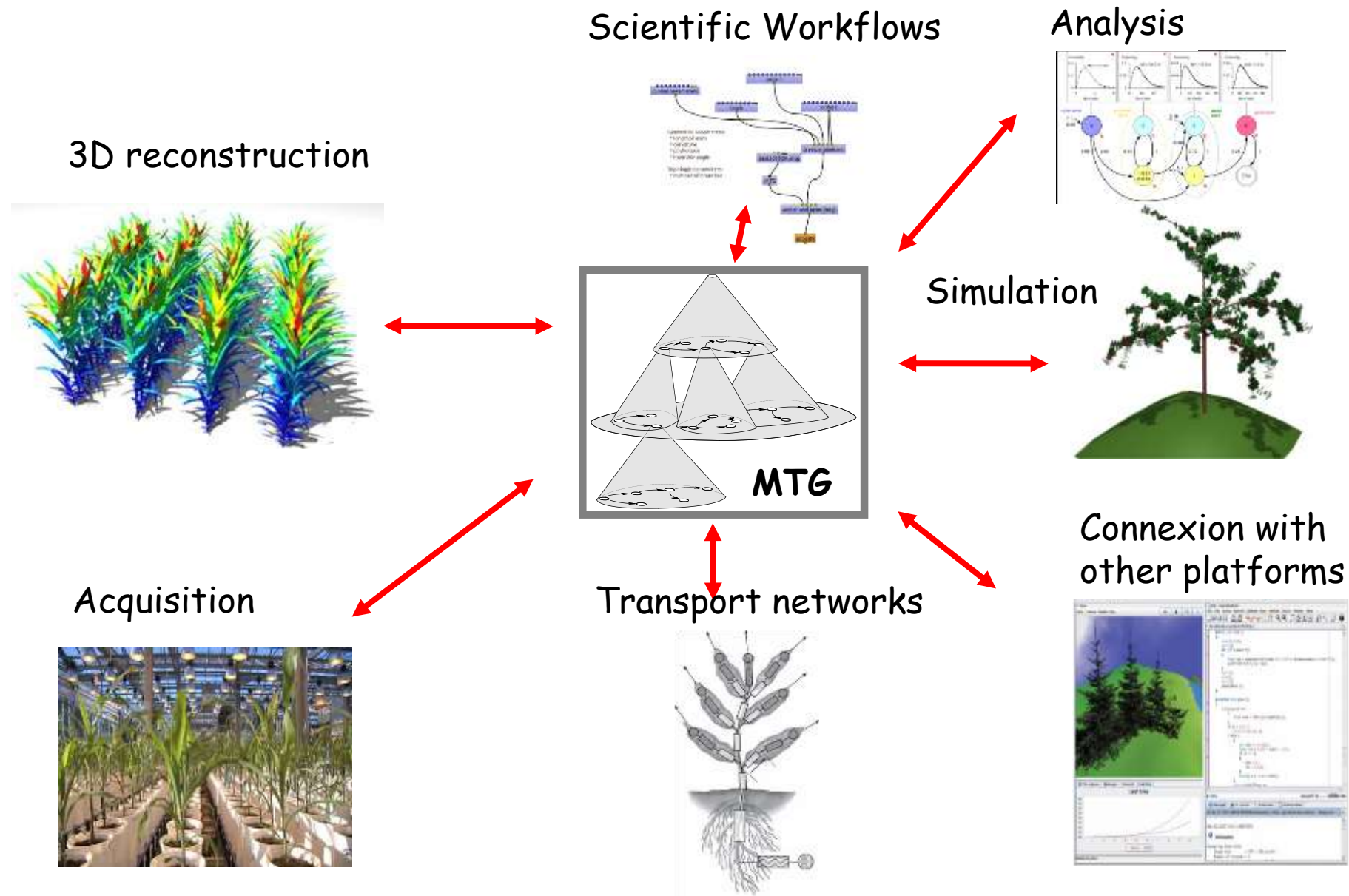
Adel Maize + Caribu

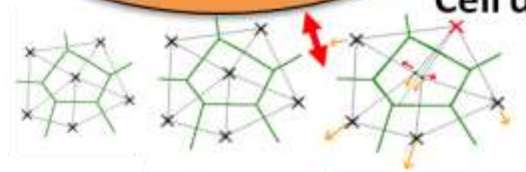
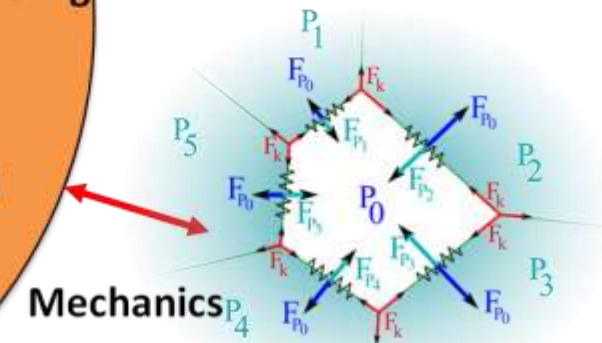
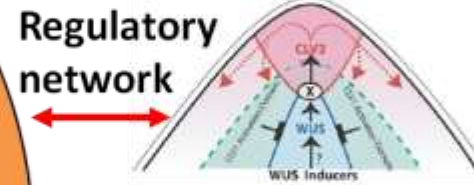
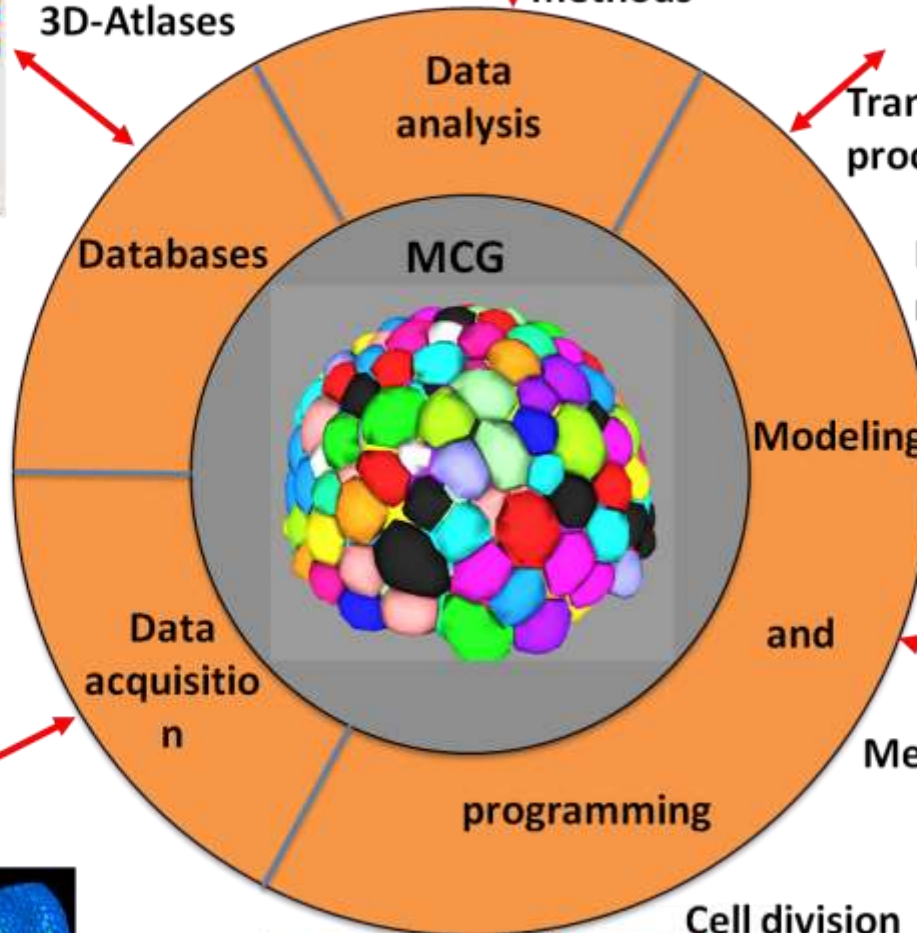
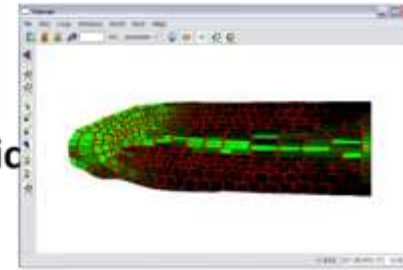
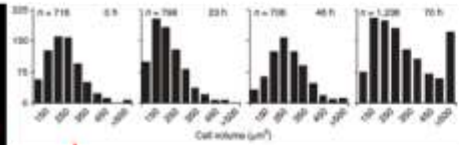
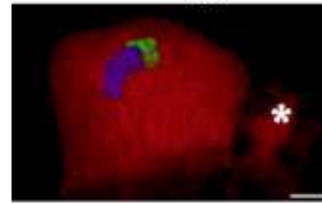
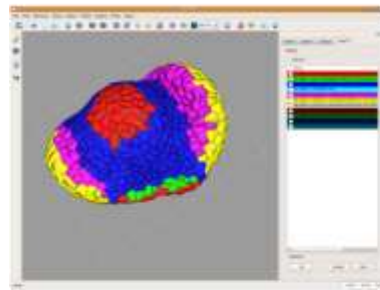


Reuse factors

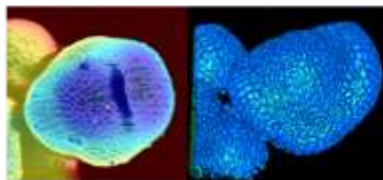
- Importance of shared data structures (numpy array, L-string, MTG, tissue, 3D images, gene sequence, ...)

Scientific Computational Experiment





Laser microscopy
Automatic
4D reconstruction

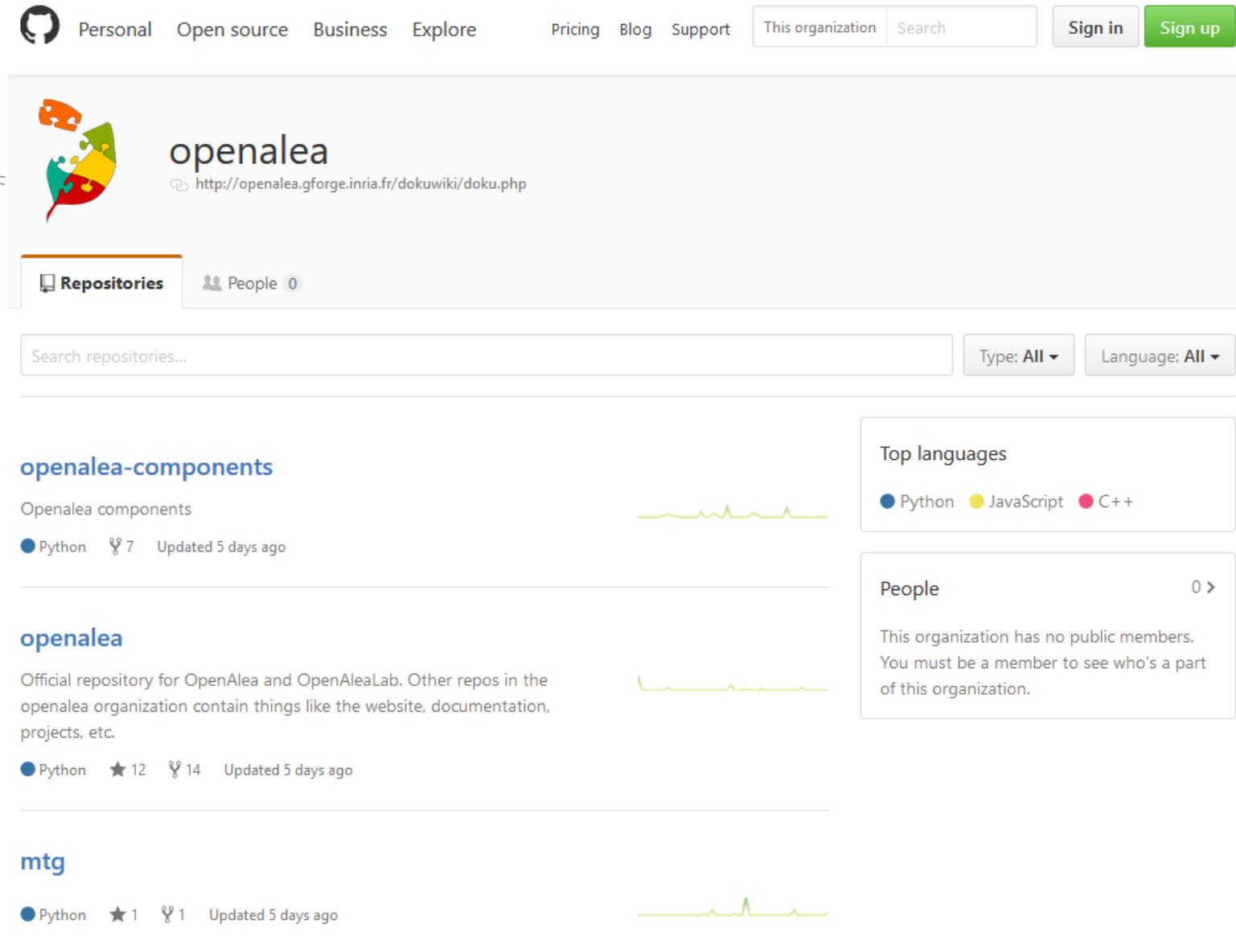


Reproducibility – How to?


- OpenAlea is primarily defined for reuse. But before reusing components you must guaranty redo and reproduce.
- How do you guaranty that using your component will be reproducible?

Technology

- Workflow github
- Organization OpenAlea

A screenshot of the OpenAlea GitHub organization page. The page header includes navigation links: Personal, Open source, Business, Explore, Pricing, Blog, Support, This organization, Search, Sign in, and Sign up. The organization's profile shows the OpenAlea logo (a colorful puzzle piece) and the name 'openalea' with a link to its documentation. Below the profile, there are tabs for 'Repositories' and 'People' (0). A search bar for repositories is present, along with filters for 'Type: All' and 'Language: All'. The main content area lists three repositories: 'openalea-components', 'openalea', and 'mtg'. Each repository entry includes a description, language (Python), star count, fork count, and update status. To the right, there are sidebars for 'Top languages' (Python, JavaScript, C++) and 'People' (0), with a note stating that the organization has no public members.

Personal Open source Business Explore Pricing Blog Support This organization Search Sign in Sign up

 **openalea**
http://openalea.gforge.inria.fr/dokuwiki/doku.php

Repositories People 0

Search repositories... Type: All Language: All

openalea-components
Openalea components
Python 7 Updated 5 days ago

openalea
Official repository for OpenAlea and OpenAleaLab. Other repos in the openalea organization contain things like the website, documentation, projects, etc.
Python ★ 12 🍴 14 Updated 5 days ago

mtg
Python ★ 1 🍴 1 Updated 5 days ago

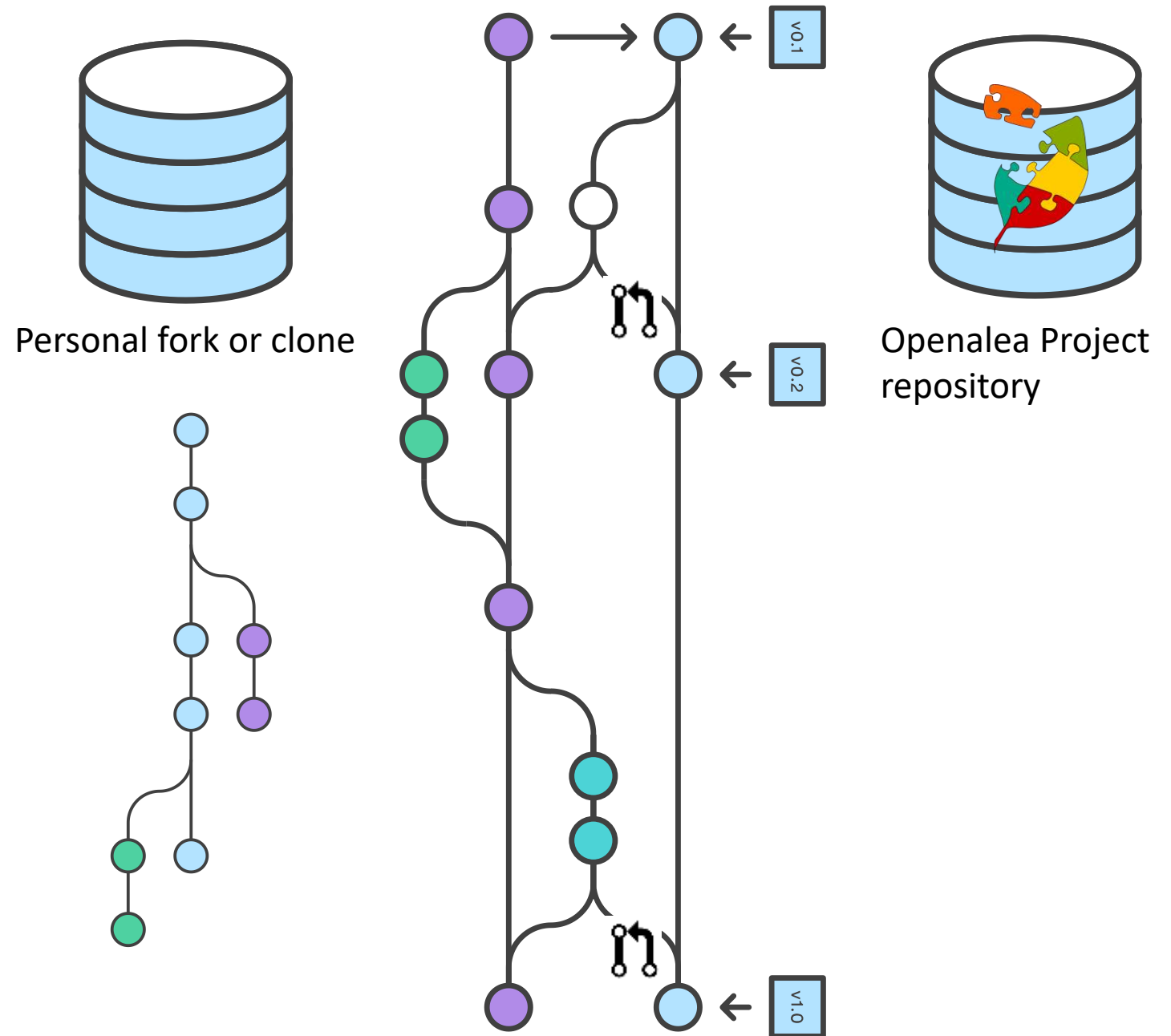
Top languages
Python JavaScript C++

People 0 >
This organization has no public members. You must be a member to see who's a part of this organization.

Technology

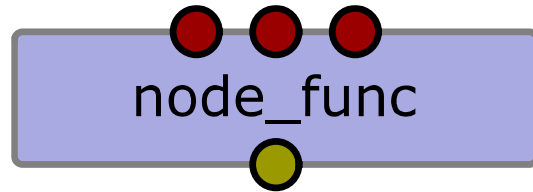


- Workflow github
- Organization OpenAlea
- Local fork or clone of a given project
- Local workflow git
- Merge requests

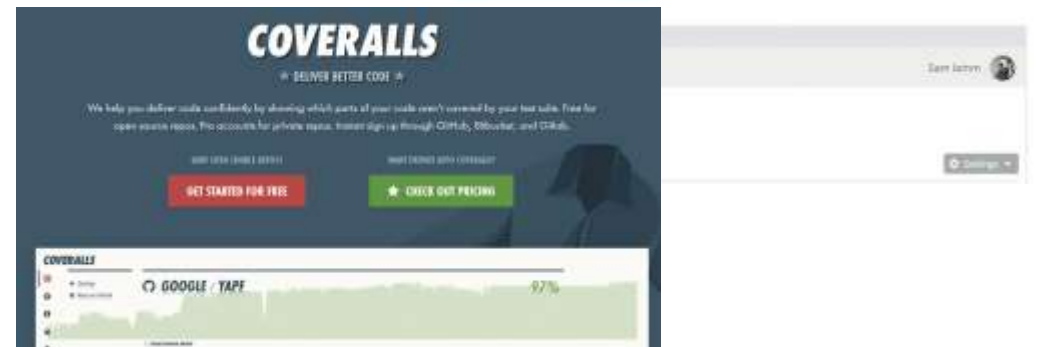


Technology - Code quality

- API
- Documentation
- Test
 - Unit tests (aka reproducibility of functions)
 - Functional tests (aka reproducibility of components)
 - Tutorials, examples (aka reproducibility of full experiments/models)
- Aesthetics

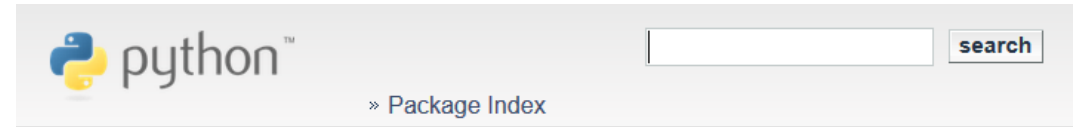


Add code quality metrics and trends to your existing deployment practices within minutes of signing up.



Technology

- Distribution, Installation
 - PyPI
 - Conda
 - github
- How to discover the component that will help you achieve your goal?



- PACKAGE INDEX »
- [Browse packages](#)
- [Package submission](#)
- [List trove classifiers](#)
- [RSS \(latest 40 updates\)](#)
- [RSS \(newest 40 packages\)](#)
- [PyPI Tutorial](#)
- [PyPI Security](#)
- [PyPI Support](#)
- [PyPI Bug Reports](#)
- [PyPI Discussion](#)
- [PyPI Developer Info](#)

PyPI - the Python Package Index


The Python Package Index is a repository of software for the Python programming language. There are currently **94159** packages here. To contact the PyPI admins, please use the [Support](#) or [Bug reports](#) links.


Not Logged In

[Login](#)

[Register](#)

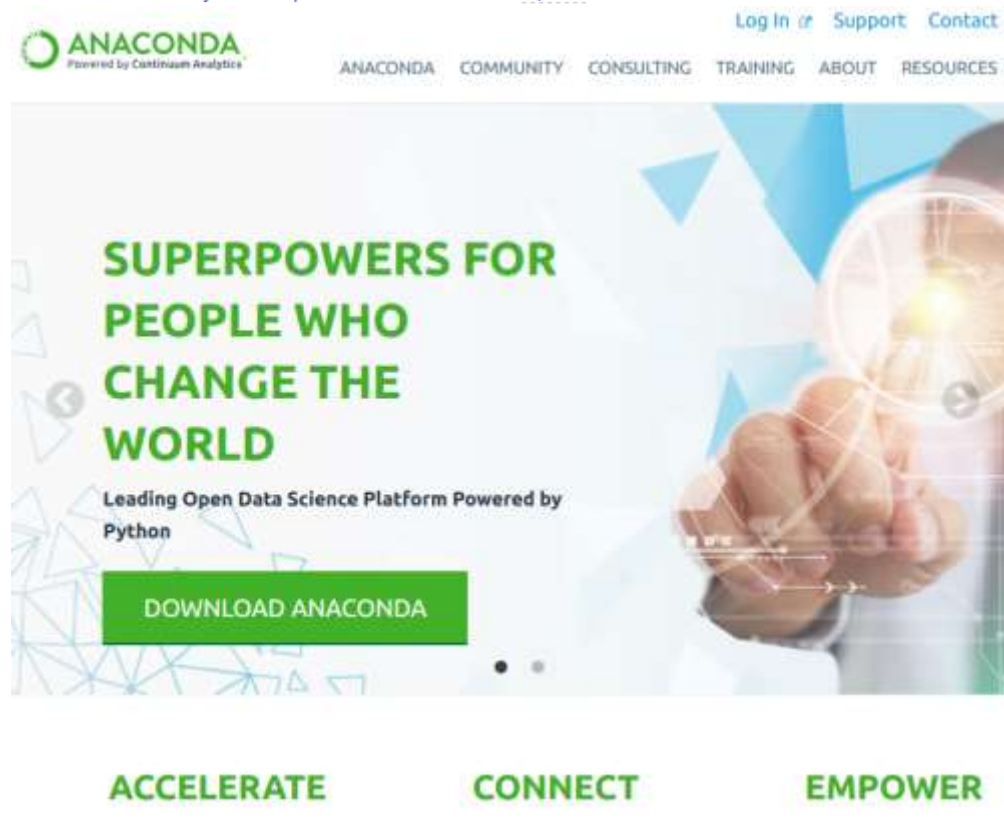
[Lost Login?](#)

[Use OpenID](#) 

[Login with Google](#) 

Status

[Nothing to report](#)



Demos

Demo redo

InfraPhenoGrid: A scientific workflow infrastructure for Plant Phenomics on the Grid

Christophe Pradal^{a,b}, Simon Artzet^c, Jerome Chopard^d, Dimitri Dupuis^e,
Christian Fournier^{c,b}, Michael Mielewczik^{c,f}, Vincent Negre^c, Pascal Neveu^d,
Didier Parigot^e, Patrick Valduriez^e, Sarah Cohen-Boulakia^{b,e,g}

^a*CIRAD, UMR AGAP, Montpellier, France*

^b*Inria, VirtualPlants, Montpellier, France*

^c*INRA, UMR459, LEPSE, F-34060 Montpellier, France*

^d*INRA, UMR729, MISTEA, F-34060 Montpellier, France*

^e*Inria, Zenith, Montpellier, France*

^f*ICCH, NHLI, Imperial College London, UK*

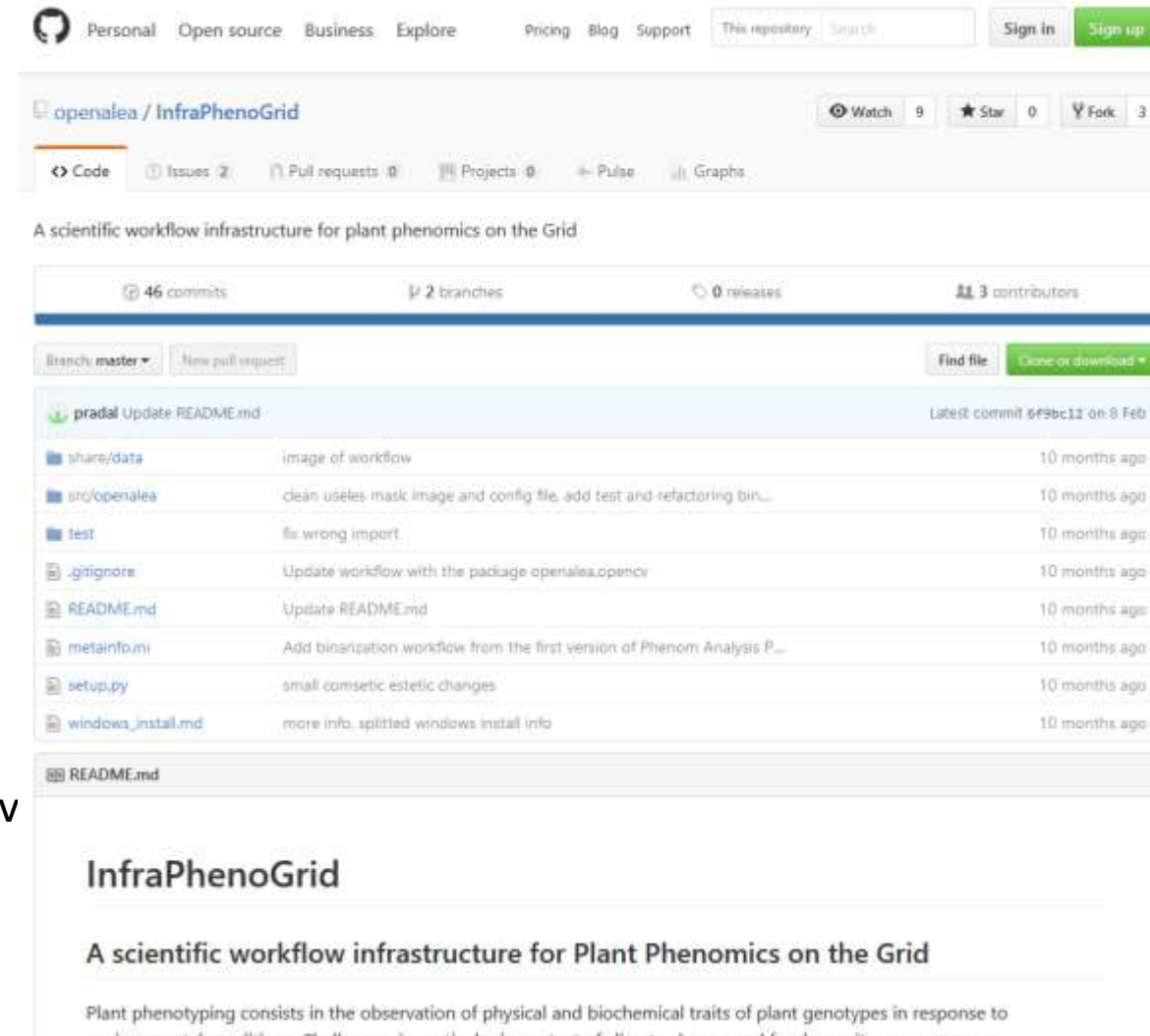
^g*Laboratoire de Recherche en Informatique, Université Paris-Sud, CNRS UMR 8623,
Université Paris-Saclay, Orsay, France*

Abstract

Plant phenotyping consists in the observation of physical and biochemical

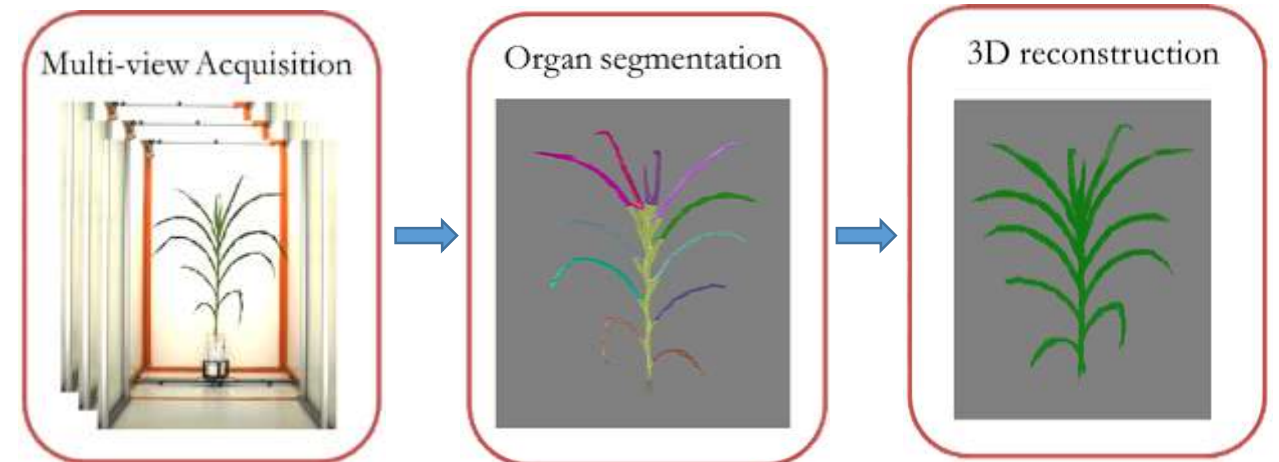
Demo redo

- Install package infraphenogrid:
 - Git clone and python setup.py install
 - Pip install
 - Conda install
- Launch OpenAlea workflow environment
- Replay
infraphenogrid/demo/binarization/demo_binarization_hsv



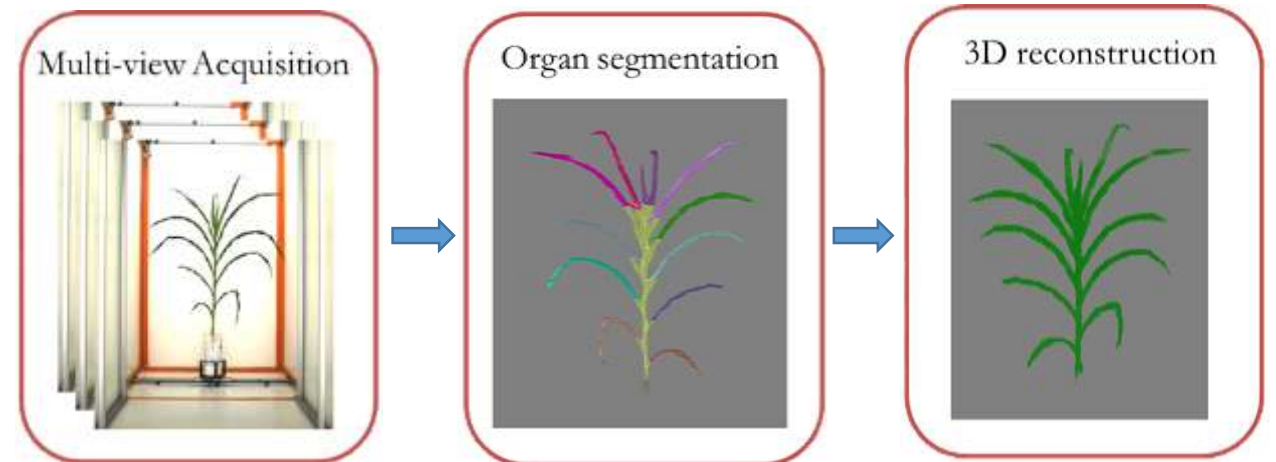
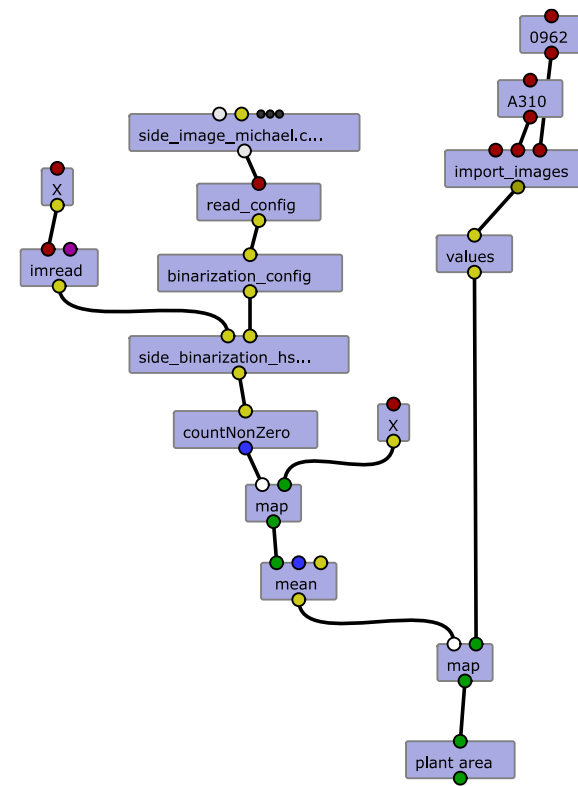
Demo reuse: Physio GWAS

- Crop plant breeding
- Finding the right trait
- Phenome, Pheno-Arch
- NGS
- Physiology modeling



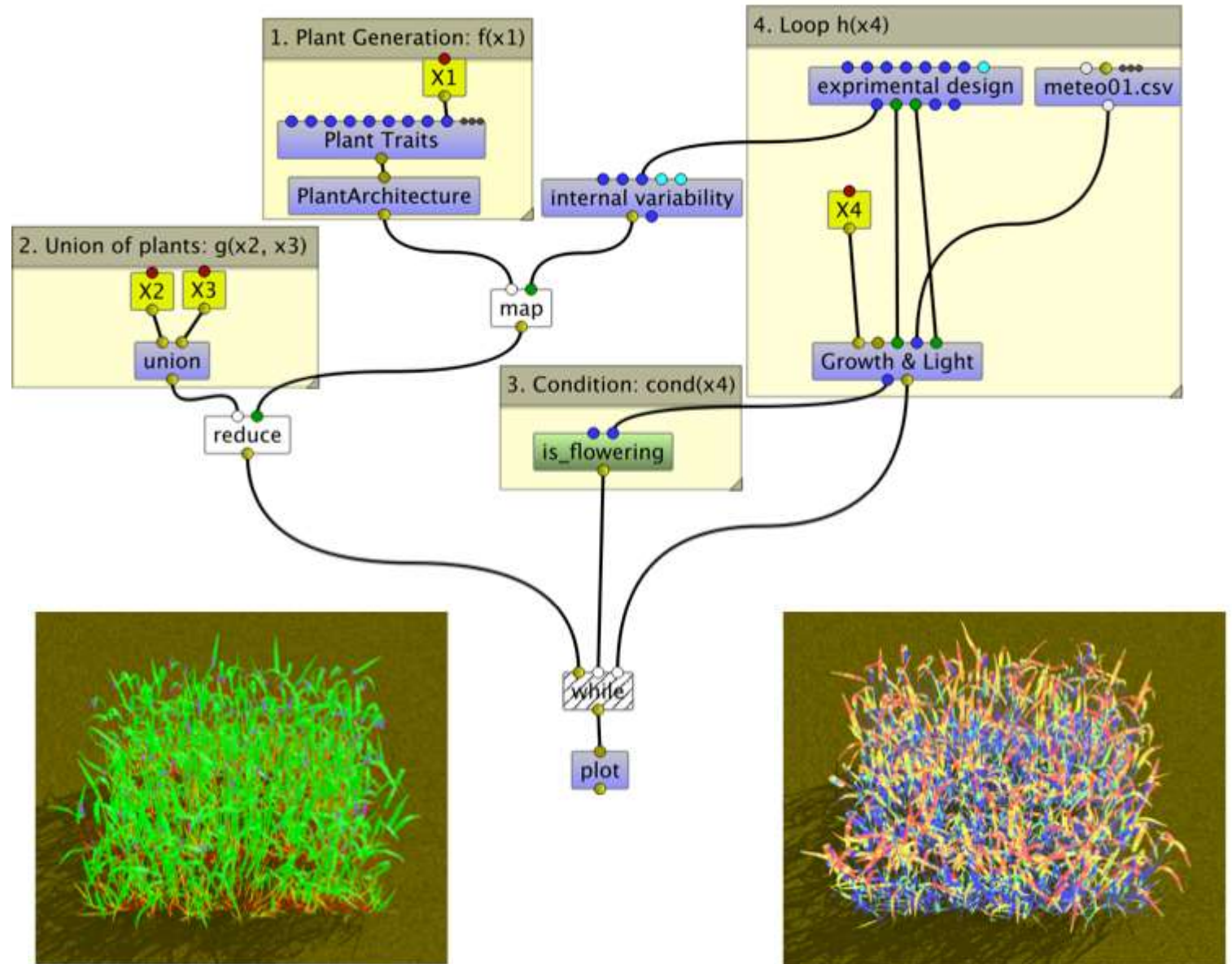
Physio GWAS

- Crop plant breeding
- Finding the right trait
- Phenome, Pheno-Arch
- NGS
- Physiology modeling



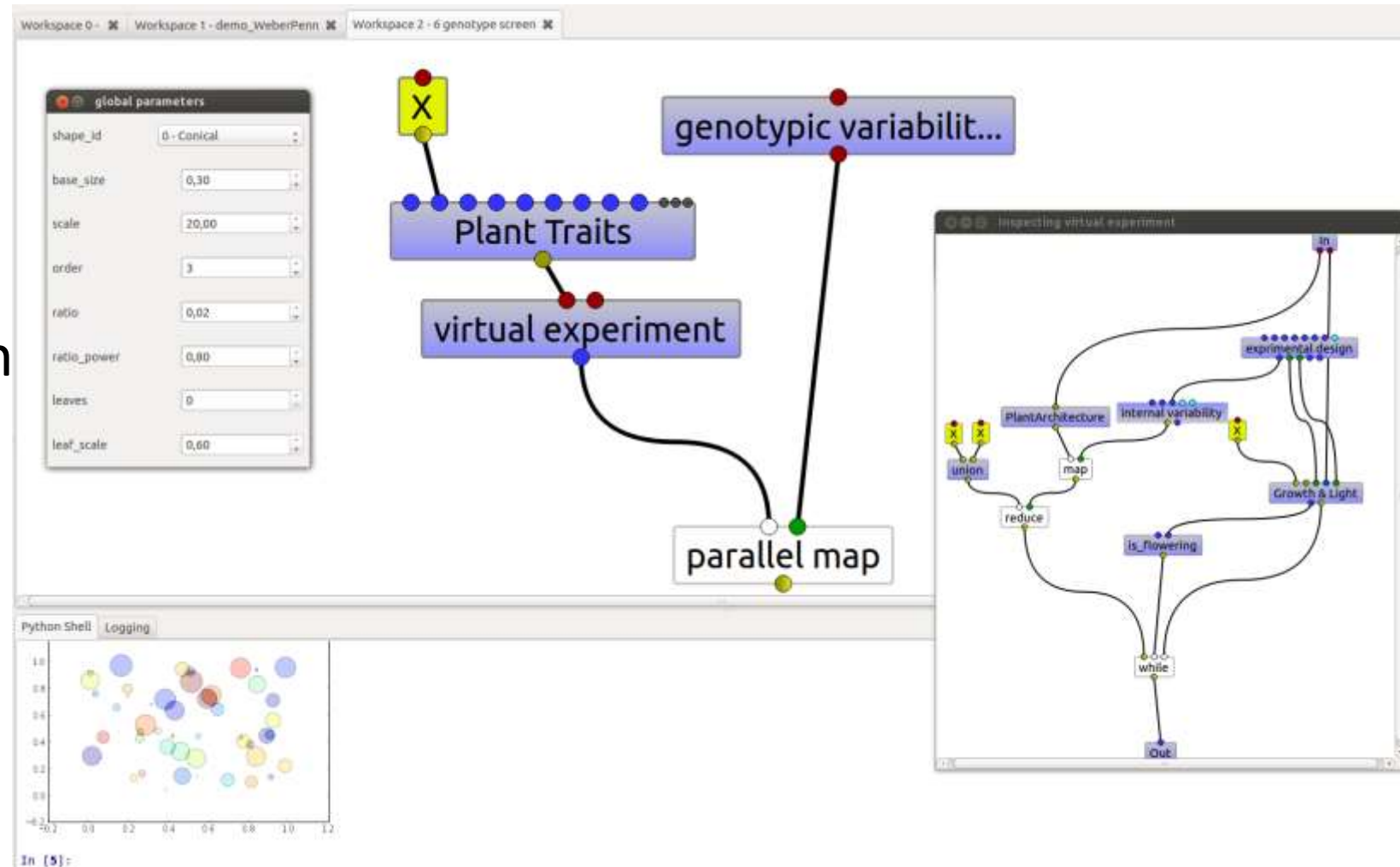
Physio GWAS

- Crop plant breeding
- Finding the right trait
- Phenome, Pheno-Arch
- NGS
- Physiology modeling



Physio GWAS

- Crop plant breeding
- Finding the right trait
- Phenome, Pheno-Arch
- NGS
- Physiology modeling



Demo reuse: disease model

Annals of Botany Page 1 of 18

doi:10.1093/aob/mcu101, available online at www.aob.oxfordjournals.org

ANNALS OF
BOTANY
Founded 1887

PART OF A SPECIAL ISSUE ON FUNCTIONAL–STRUCTURAL PLANT MODELLING

A modelling framework to simulate foliar fungal epidemics using functional–structural plant models

**Guillaume Garin^{1,2,*}, Christian Fournier³, Bruno Andrieu², Vianney Houlès¹, Corinne Robert²
and Christophe Pradal^{4,5}**

¹*ITK, avenue de l'Europe, F-34830 Clapiers, France,* ²*INRA, UMR 1091 EGC, F-78850 Thiverval-Grignon, France,* ³*INRA, UMR 759 LEPSE, F-34060 Montpellier, France,* ⁴*CIRAD, UMR AGAP and INRIA, Virtual Plants, F-34398 Montpellier, France and*

⁵*Institut de Biologie Computationnelle, F-34095 Montpellier, France*

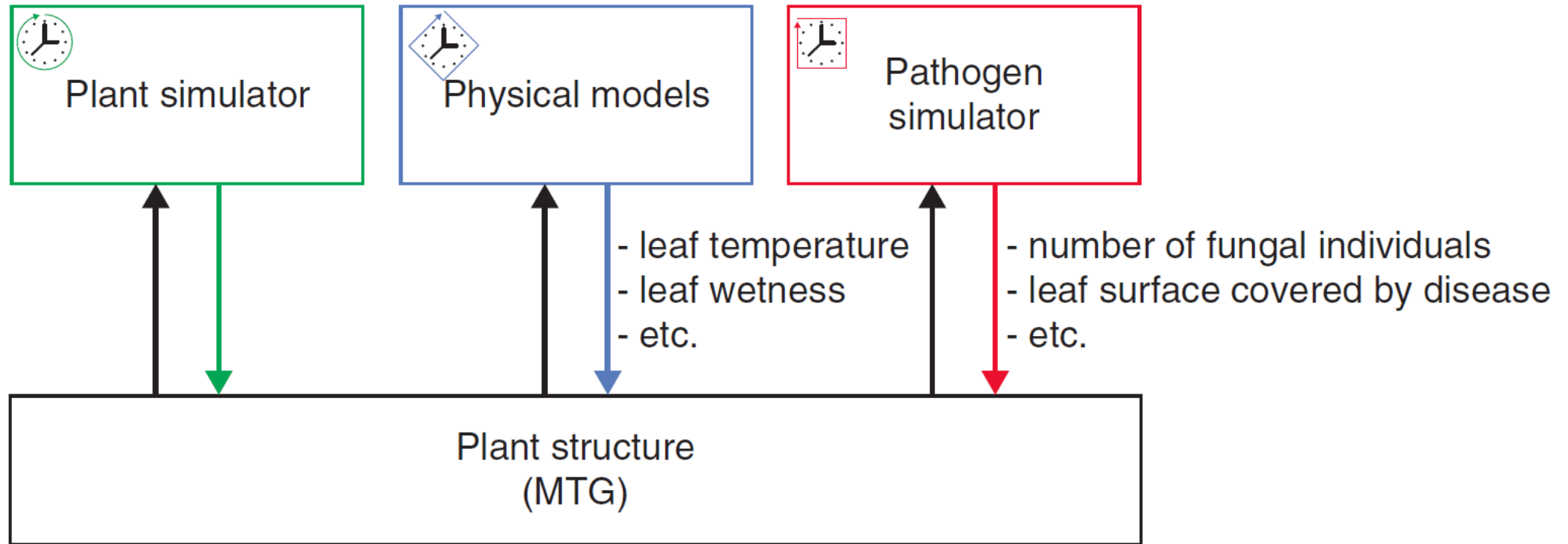
**For correspondence. E-mail guillaume.garin@itkweb.com*

Received: 29 November 2013 Returned for revision: 17 March 2014 Accepted: 28 April 2014

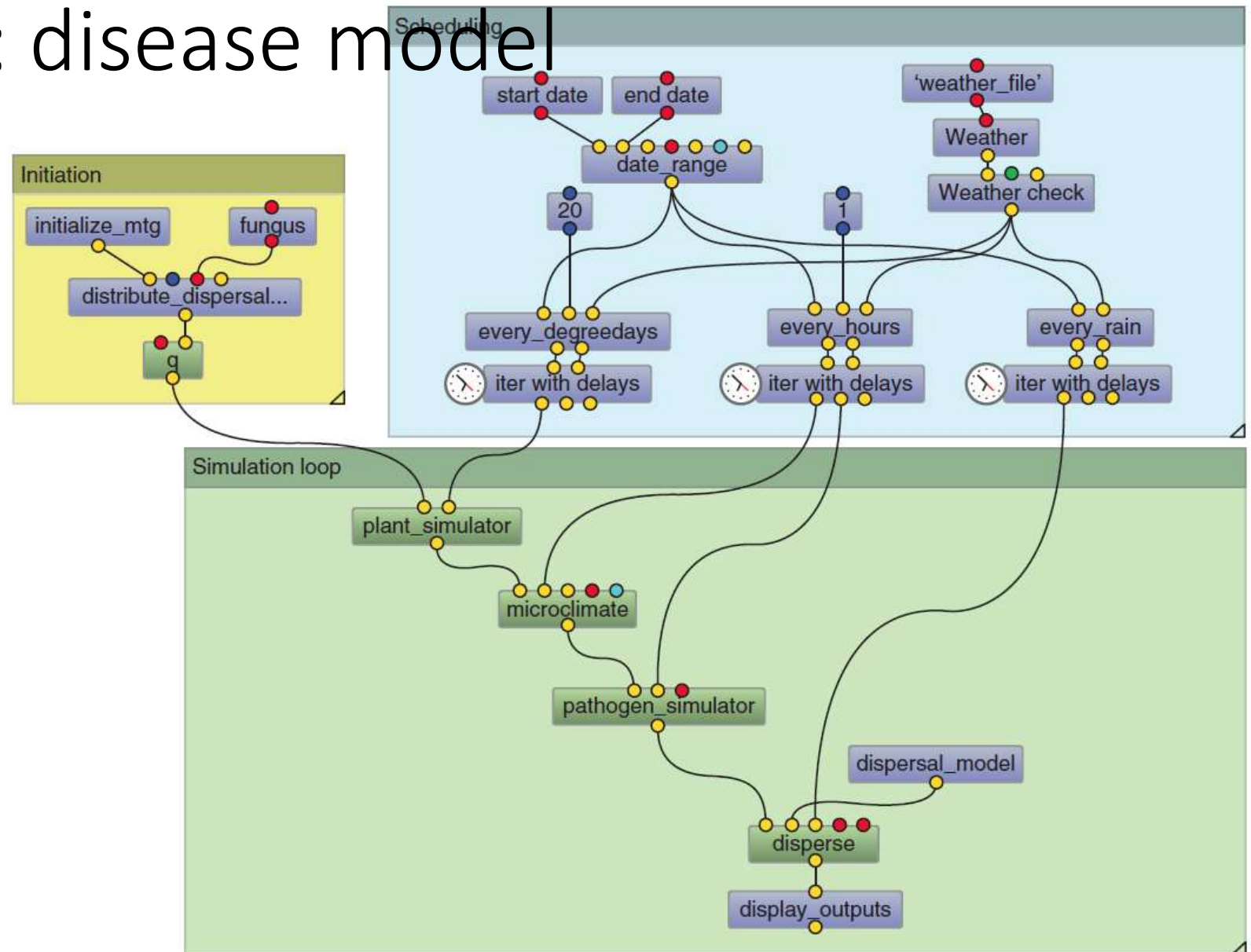
• **Background and Aims** Sustainable agriculture requires the identification of new, environmentally responsible strategies of crop protection. Modelling of pathosystems can allow a better understanding of the major interactions

Demo reuse: disease model

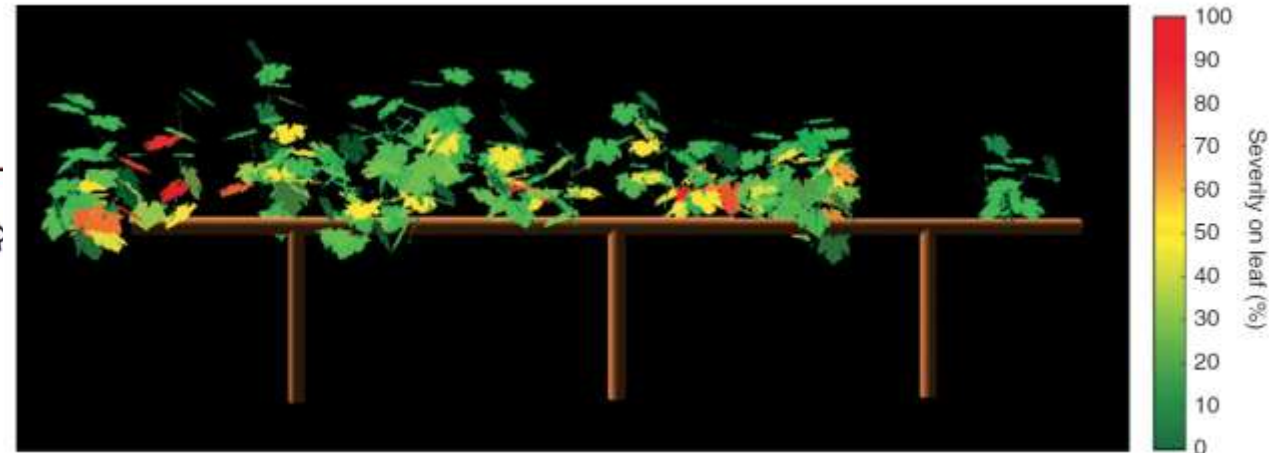
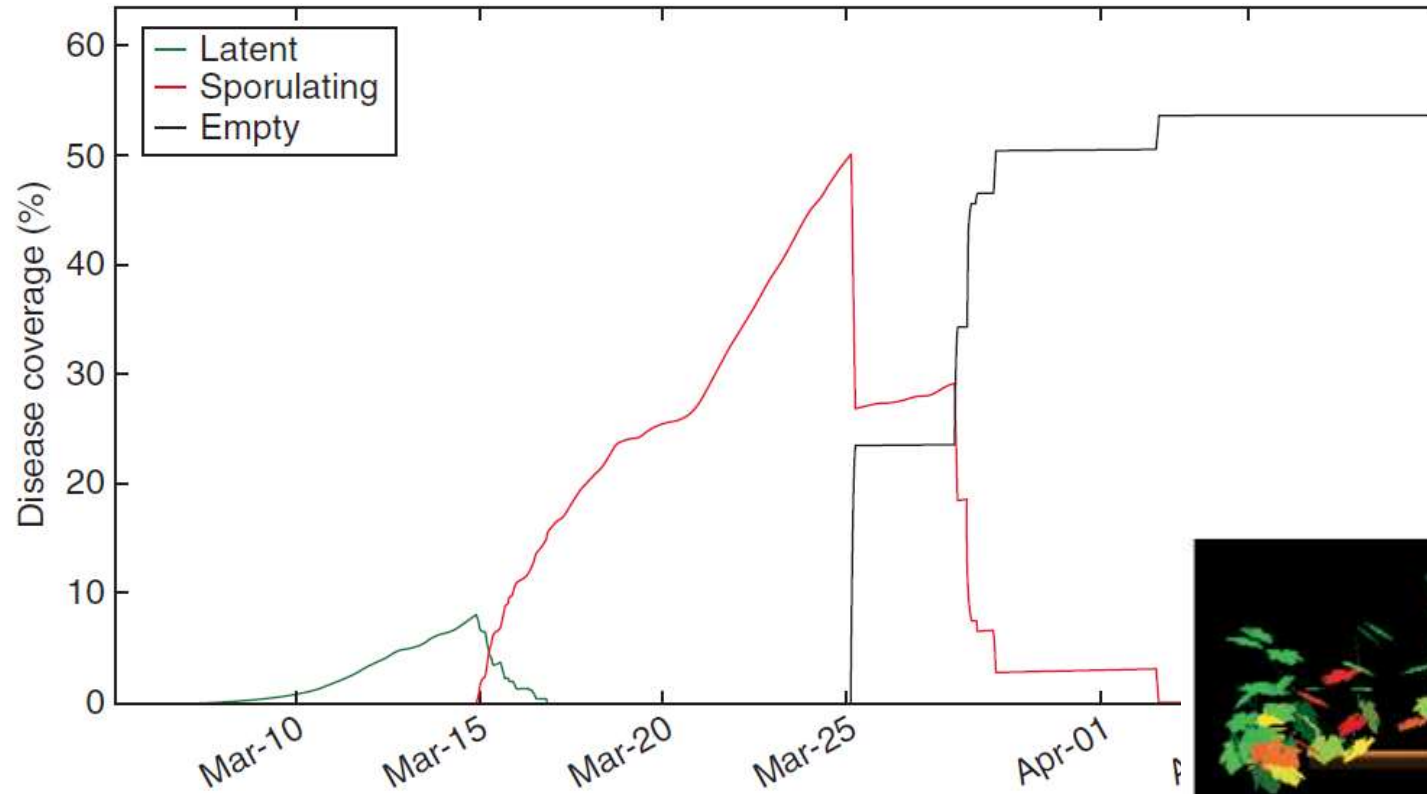
Garin et al. — Modelling framework to simulate foliar fungal epidemics



Demo reuse: disease model



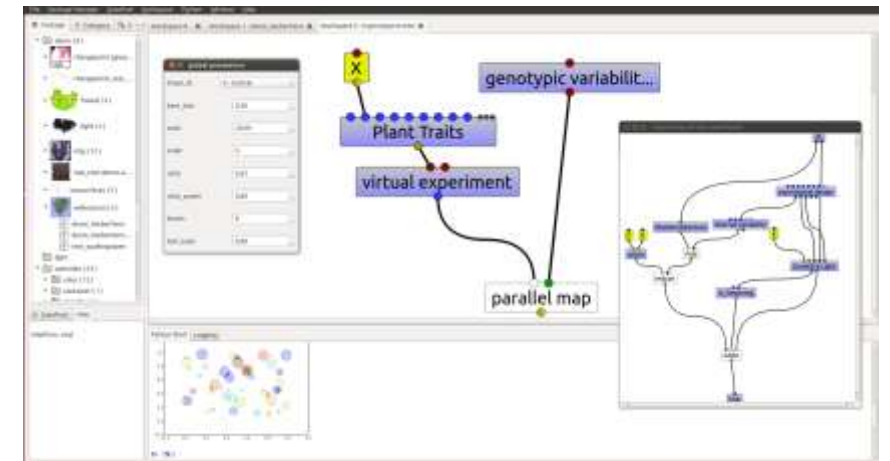
Demo reuse: disease model





Conclusion on OpenAlea workflow system

- Easy install on personal computers by end users
- Everybody can create nodes and workflows easily
- Nested workflows aka Composite nodes
- Mix analysis workflows with simulation workflows (e.g. loops)



End