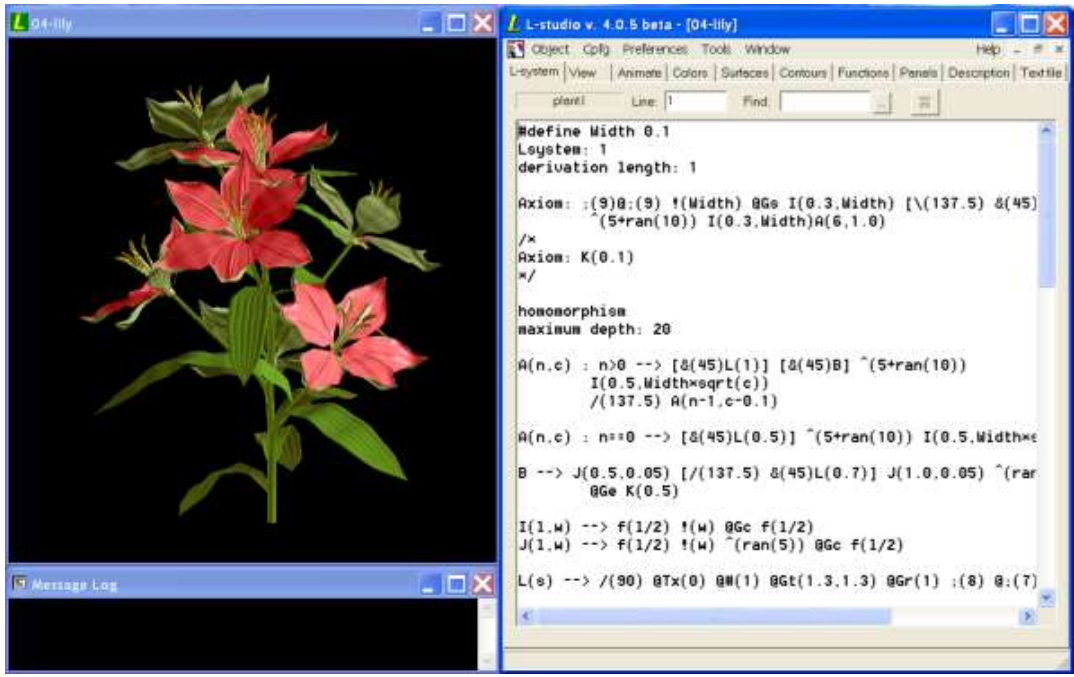
Reproducibility and OpenAlea

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?

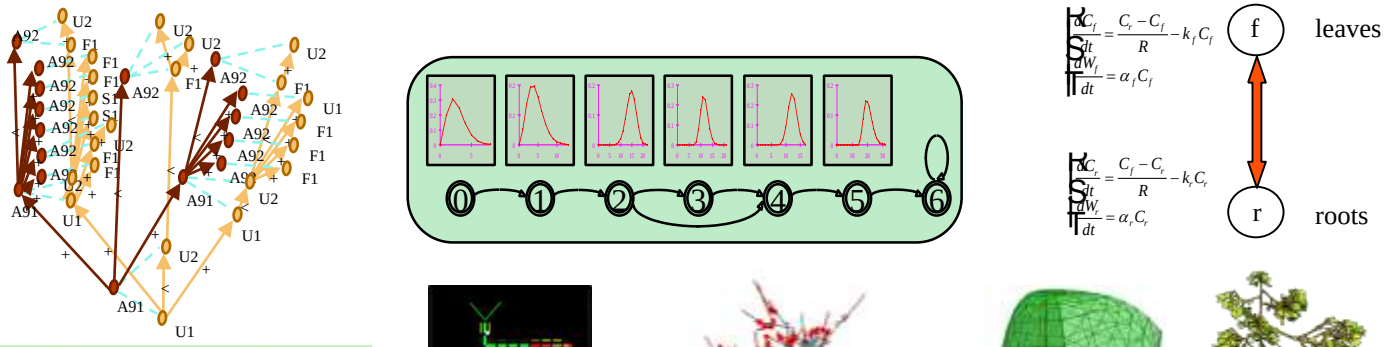


Plant modeling community

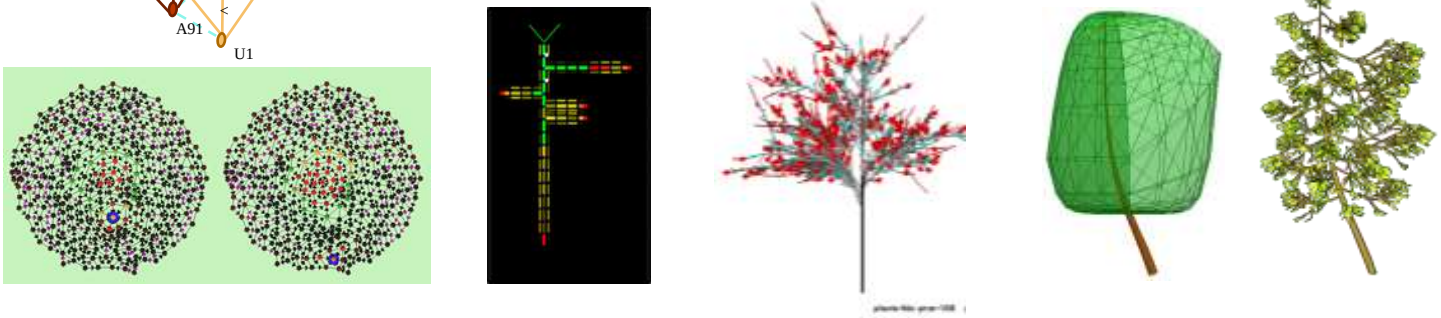
Biological objects

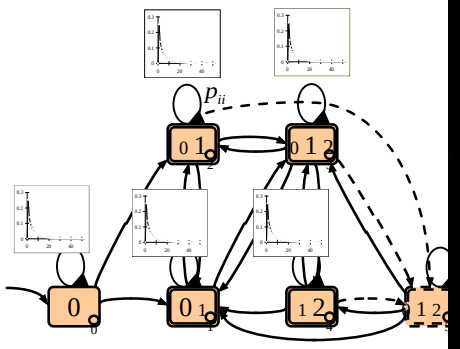
Mesure

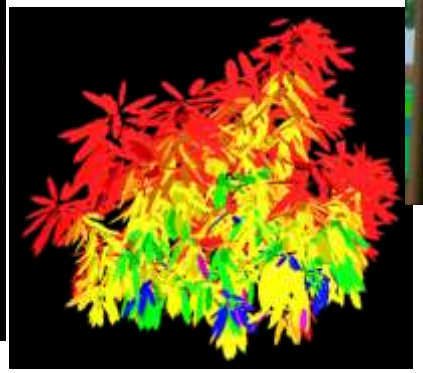
Modeling

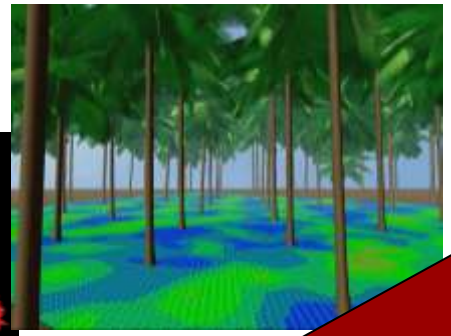


Simulation









time & complexity of biological problems

Interoperable
Software
Components
for modellers

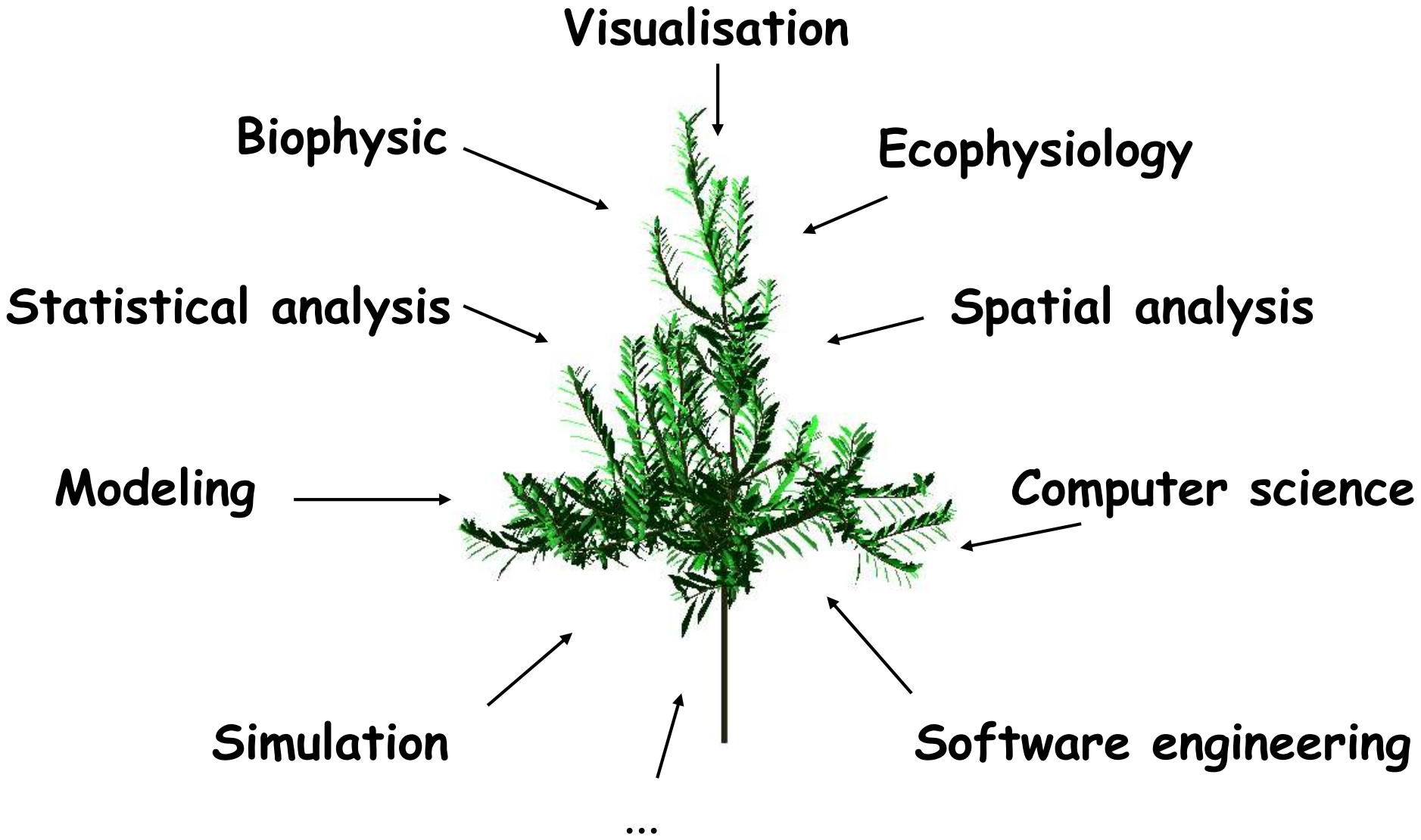
Interoperable
Software Components
for programmers

Software Components
for programmers

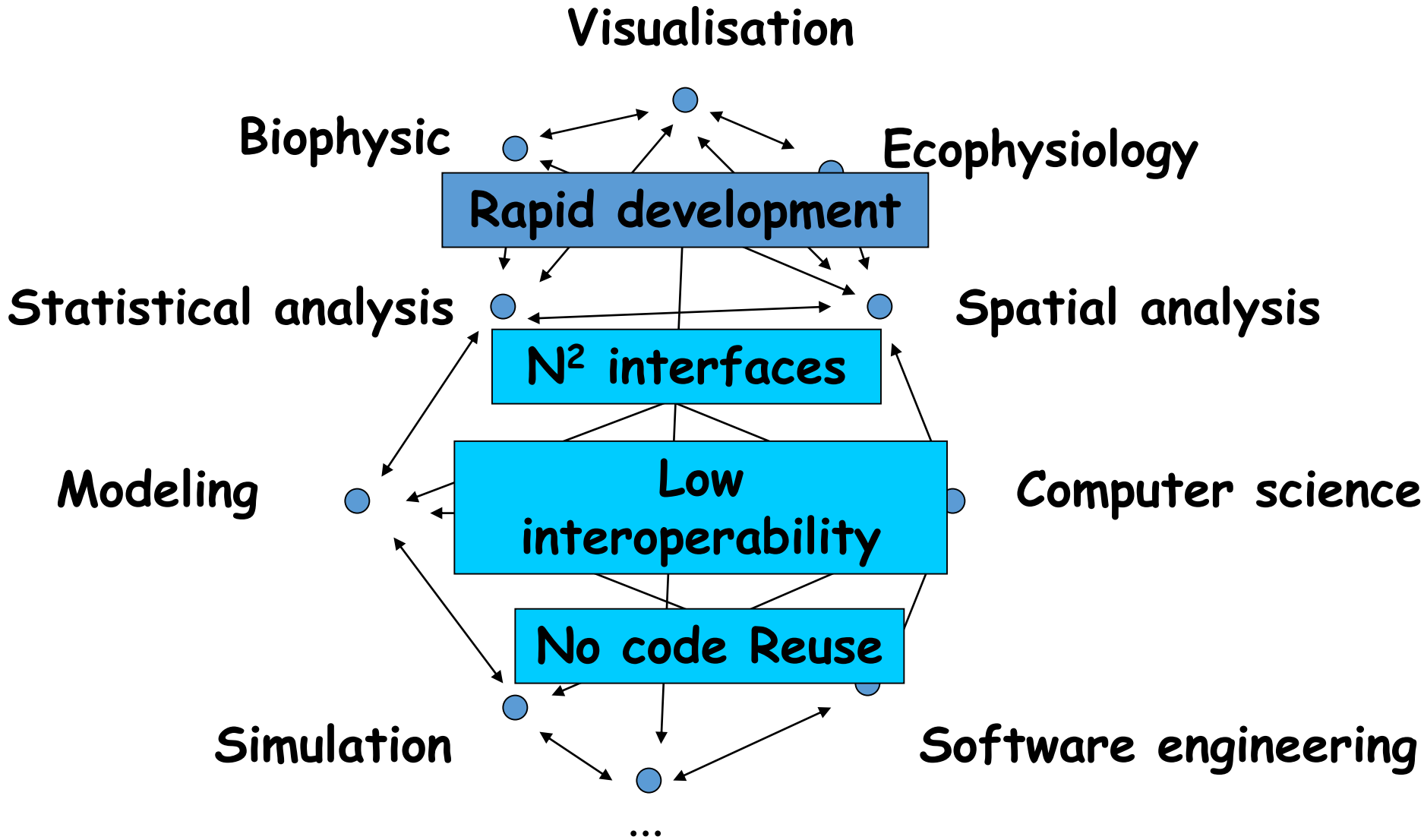
Monolithic Software

http://www.scripps.edu/~sanner/

Challenge – Software reuse



« Traditional » solution



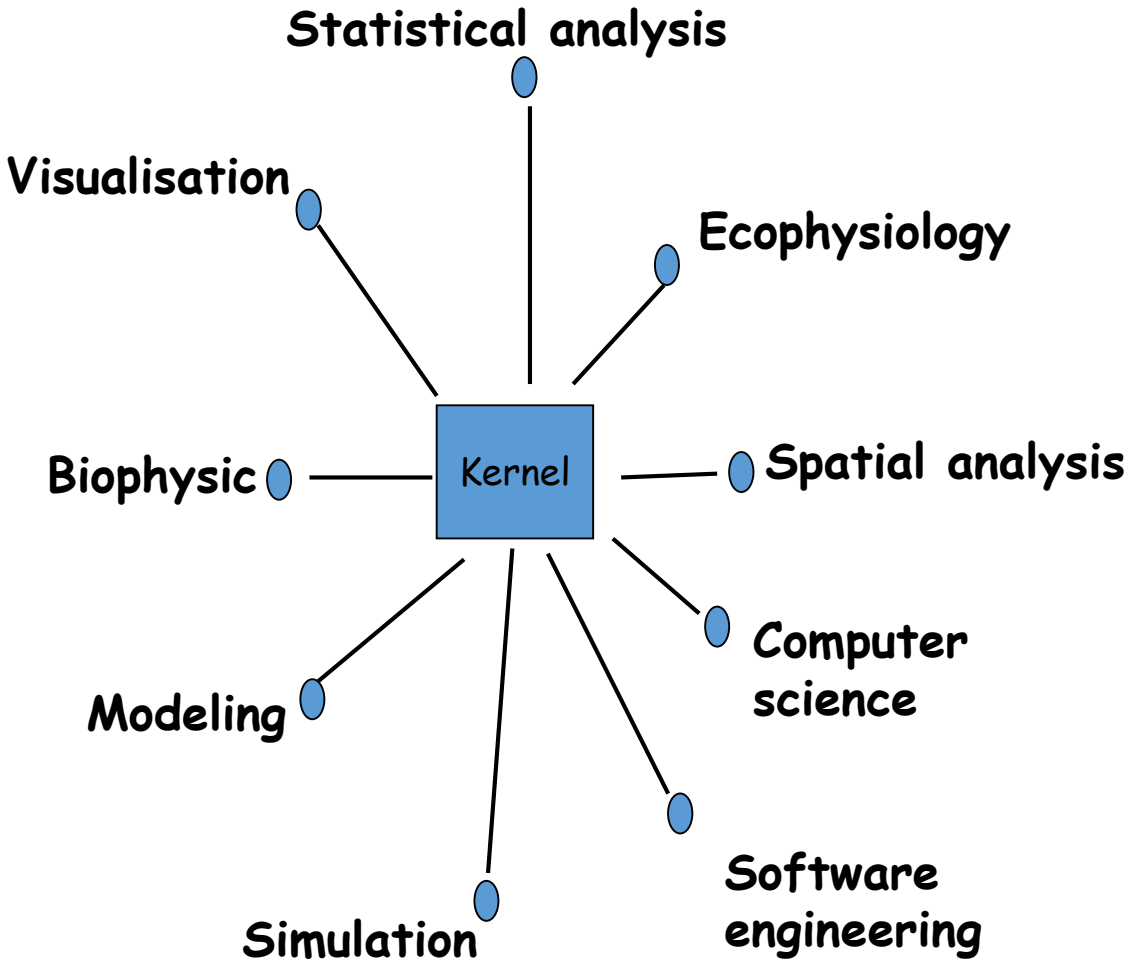
« Application » centric

- + Code Reuse by modularity

  Object oriented technics

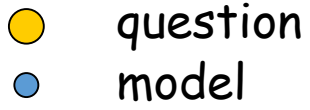
- « self-centric »: data, types,API

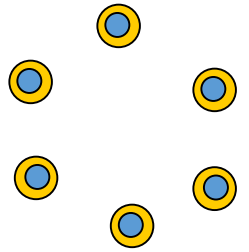
  Low interoperability between applications

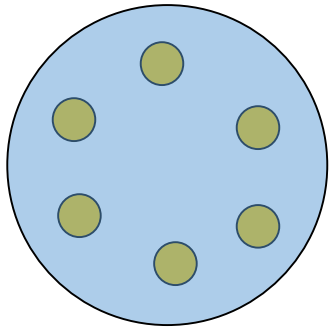


Modelling strategy

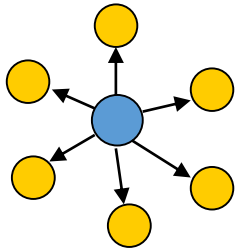
1. Construct the best model (efficient & simple) for each new situation







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?

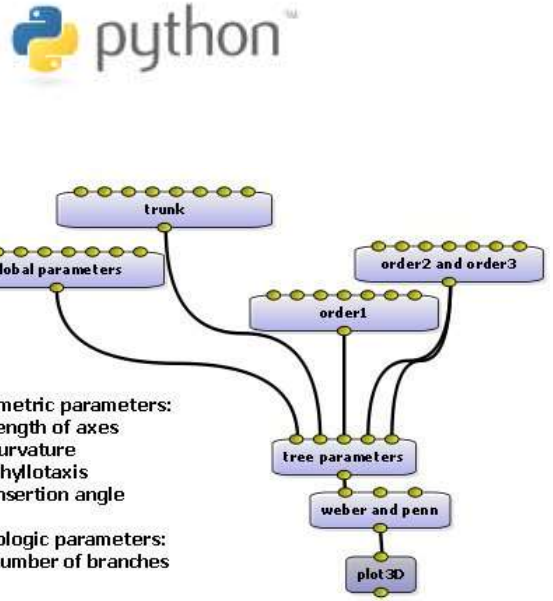
Christophe Pradal

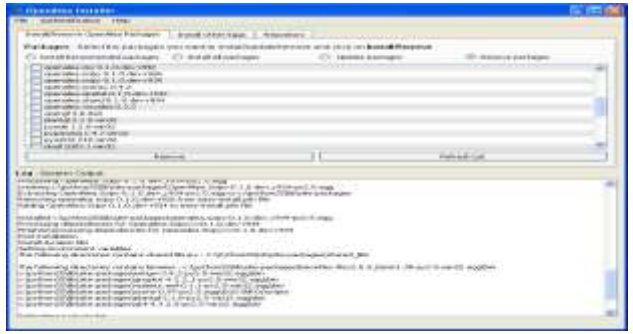
1) The freedom to run the program, for any purpose.
2) The freedom to study how the program works, and adapt it to your needs.
3) The freedom to redistribute copies so you can help your neighbor.
4) The freedom to improve the program, and release your improvements to the public, so that the whole community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

Source: http://fsfe.org/about/basics/freesoftware.en.html

Design Principles

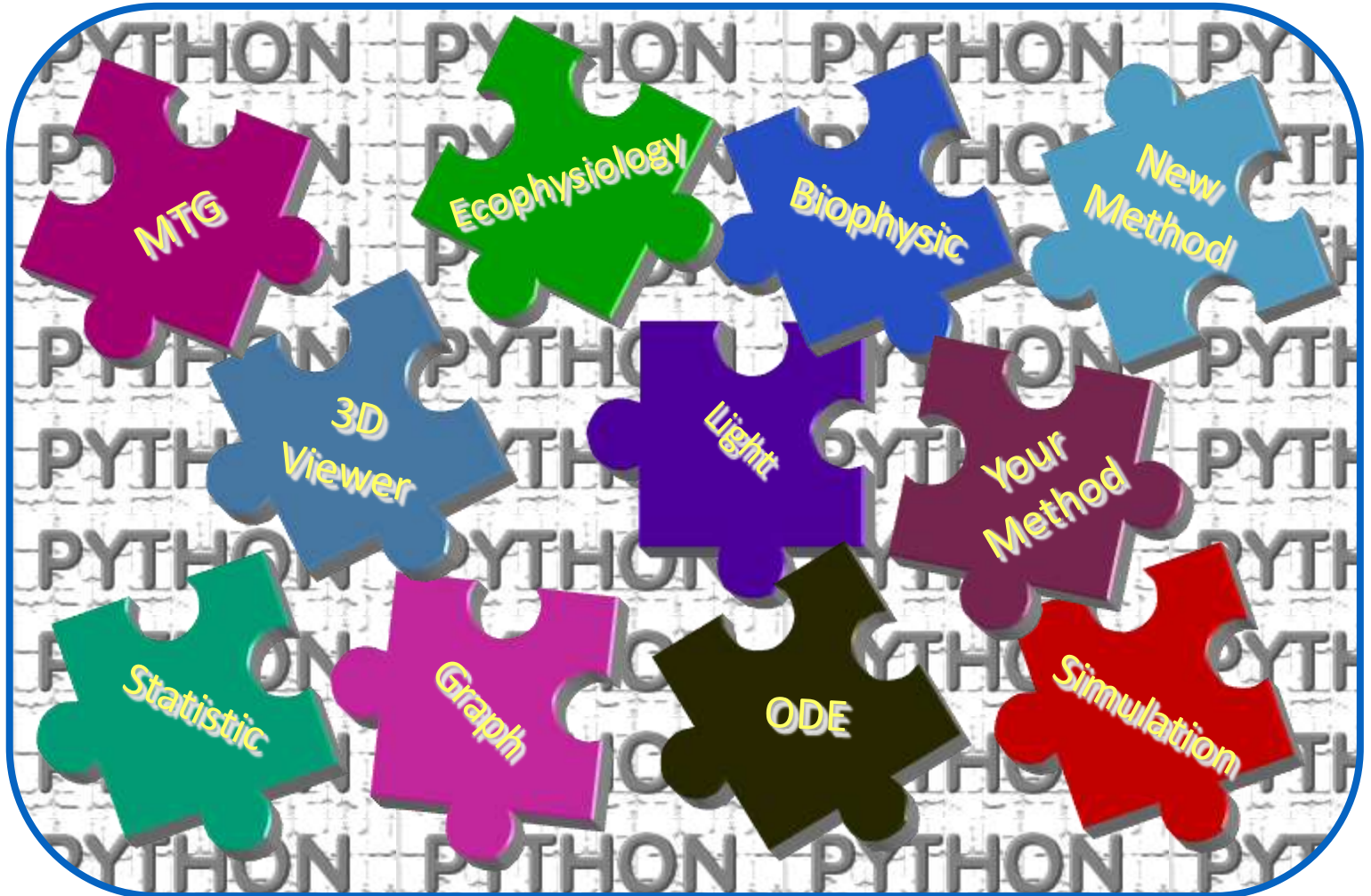
- Language centric
  - Common modeling language
  - Glue language
- Component architecture
  - Dynamic composition
  - High-level dataflow approach
- Visual programming
  - Graphical model representation
  - Automatic GUI generation
- Shared deployment tools
  - Build, packaging, distribution, installation,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





Language centric

High level language as a scripting environment



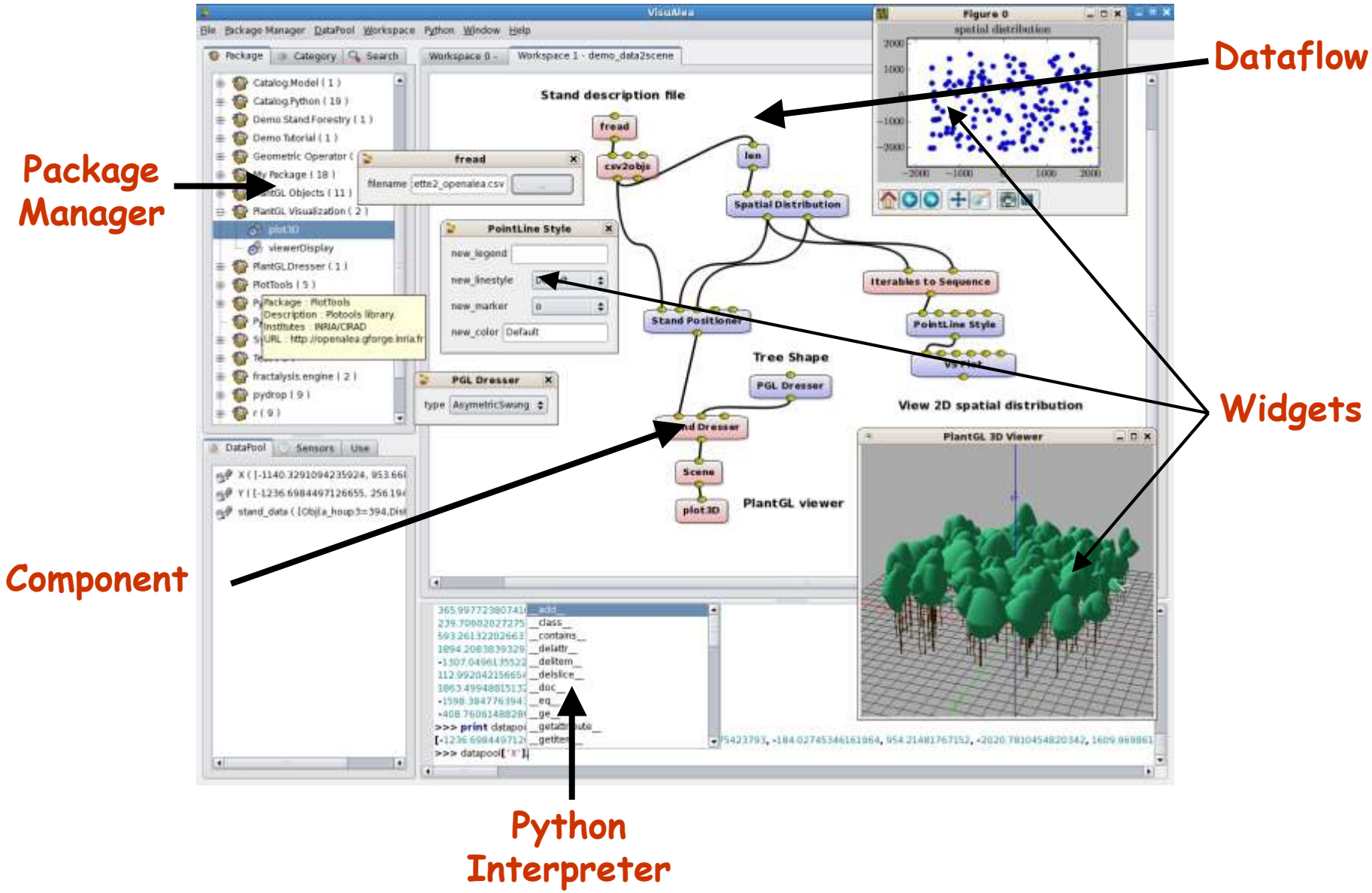
Interactive

Dynamic

Platform independent

http://www.scripps.edu/~sanner/

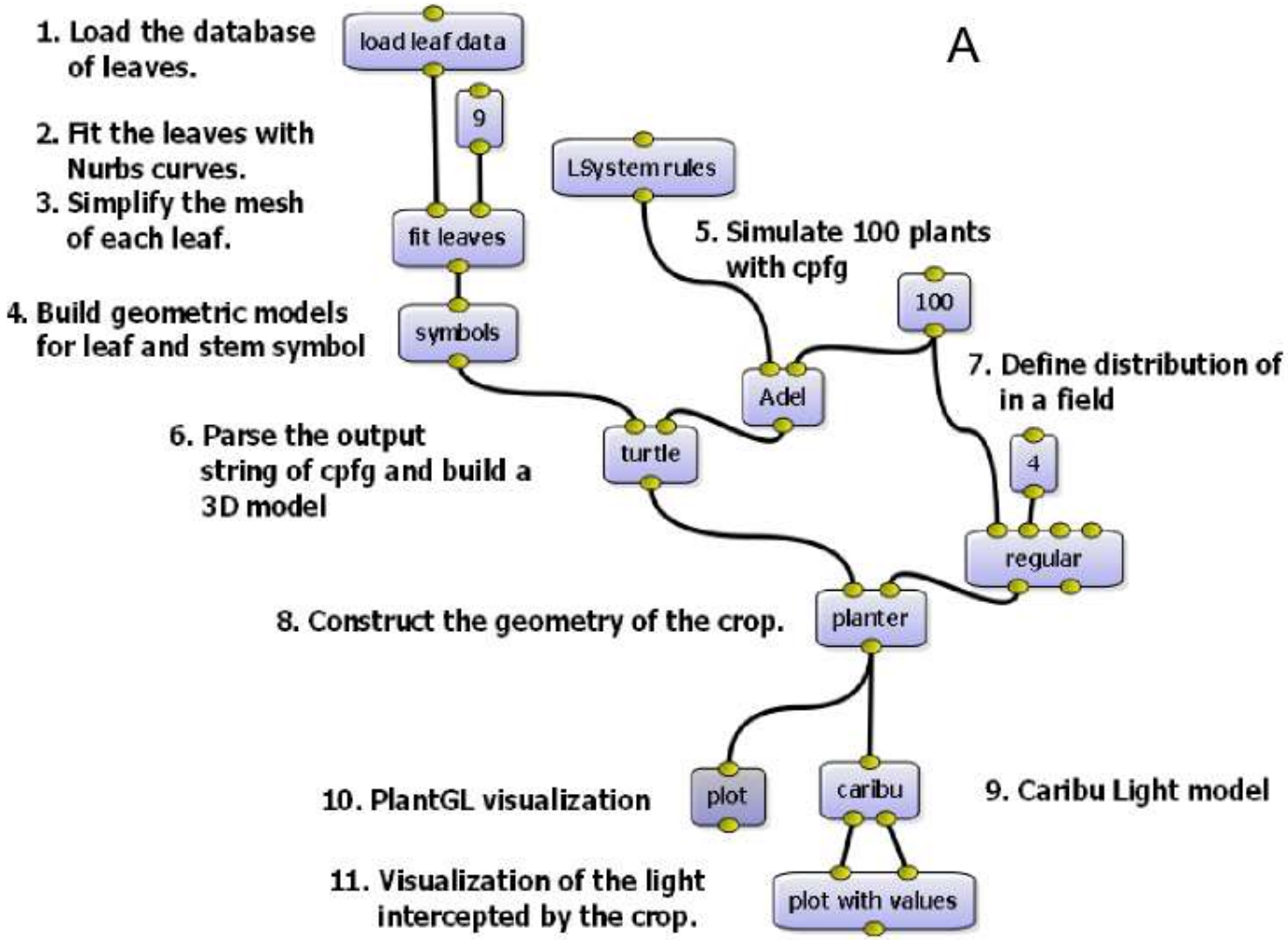
Component framework

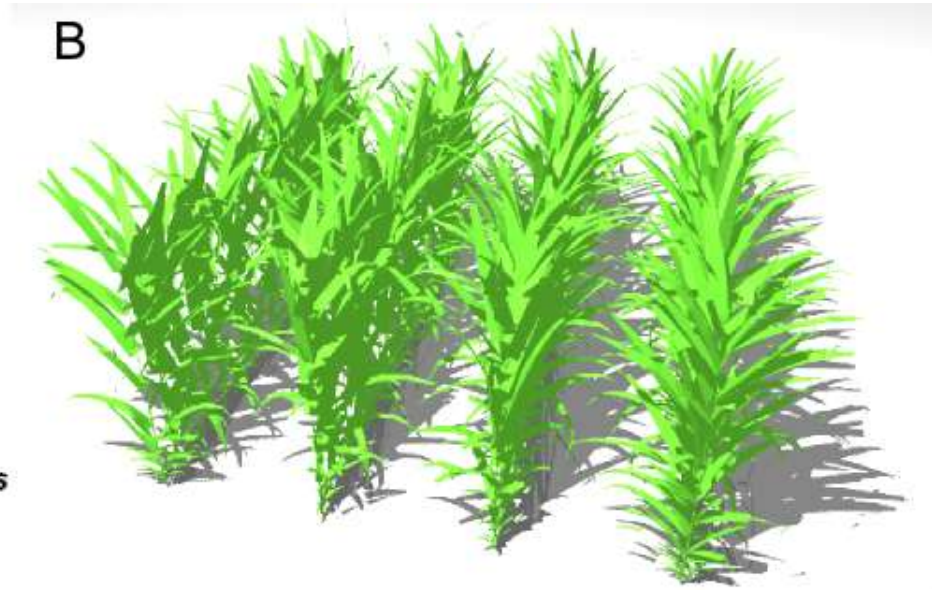


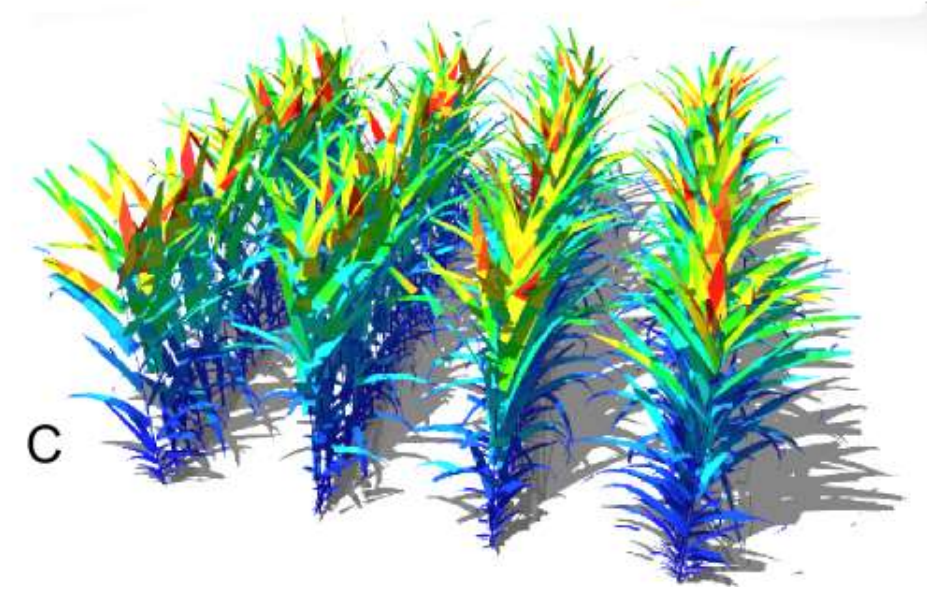
OpenAlea workflows

- 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



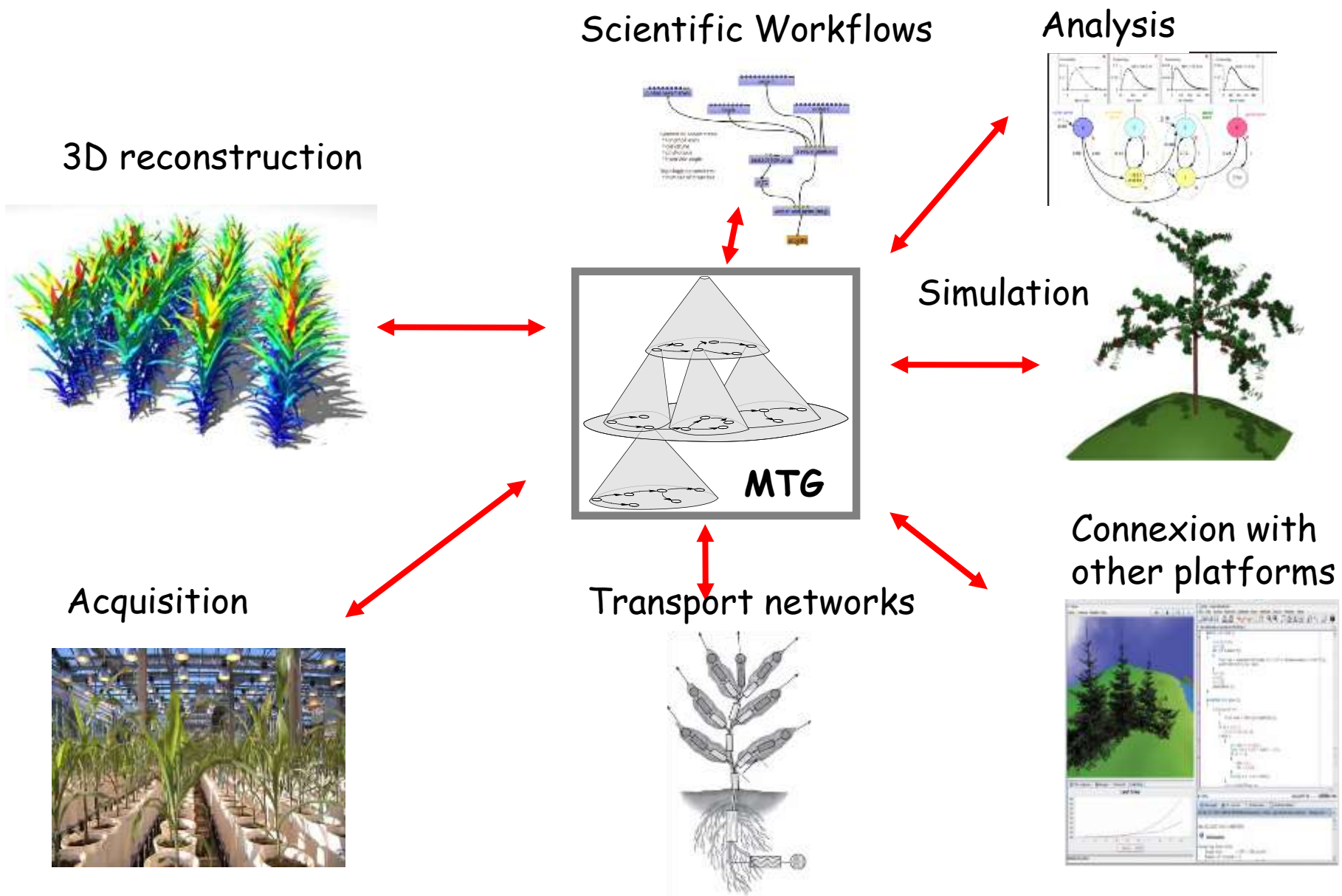


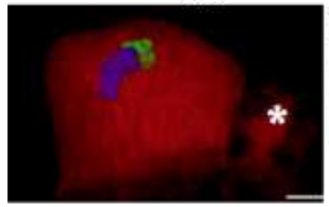


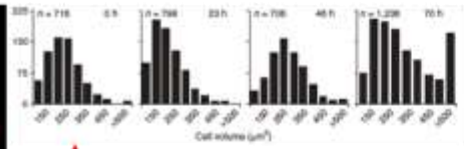
Reuse factors

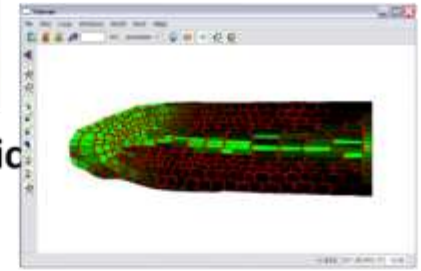
- Importance of shared data structures (numpy array, L-string, MTG, tissue, 3D images, gene sequence, …)

Scientific Computational Experiment

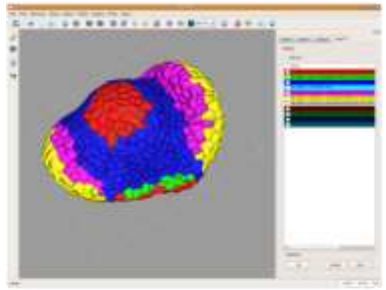








Statistical / geometric methods



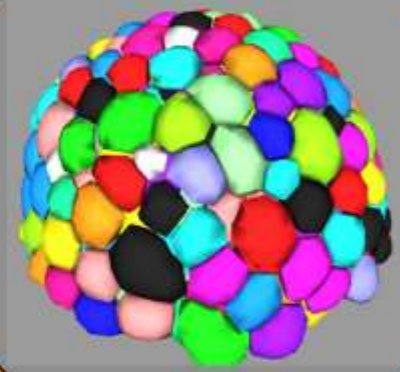
3D-Atlases

Data analysis

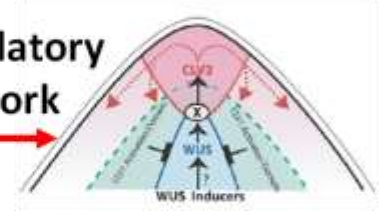
Transport processes

Databases

MCG



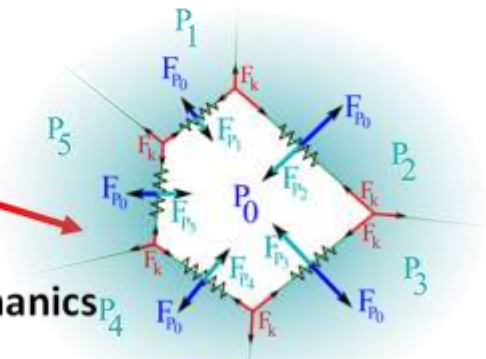
Regulatory network



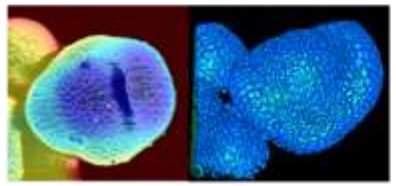
Modeling and programming

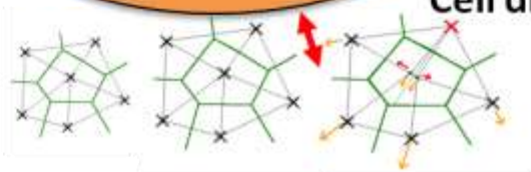
Laser microscopy
Automatic
4D reconstruction

Data acquisition



Mechanics





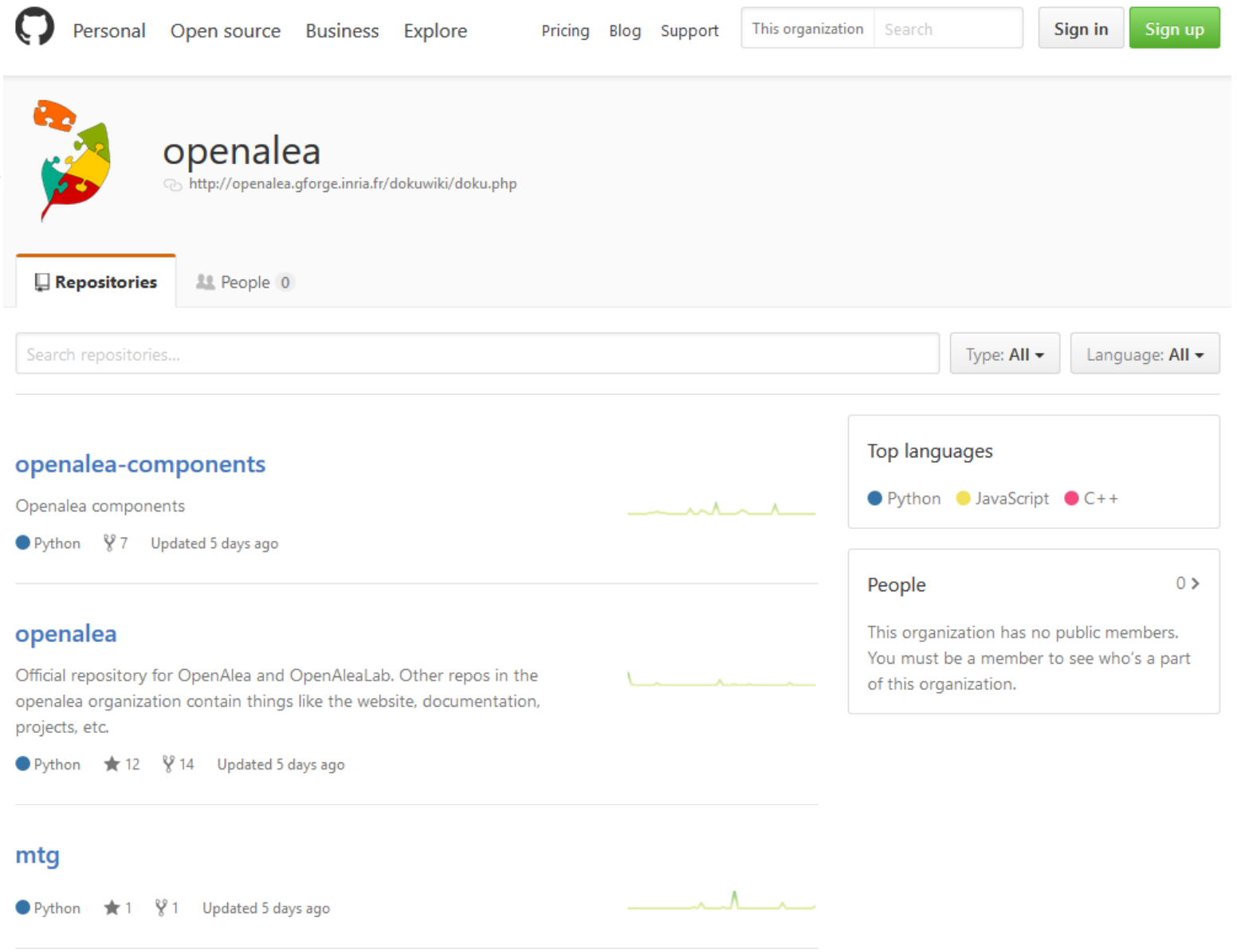
Cell division

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?

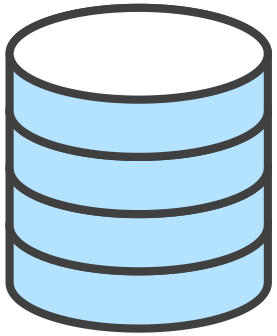
Technology

- Workflow github
- Organization OpenAlea

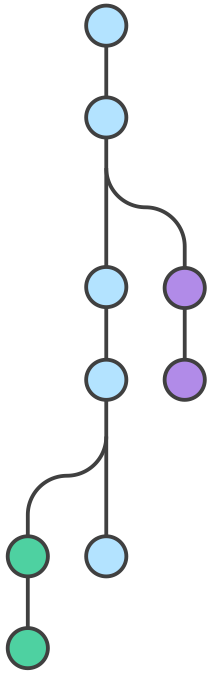


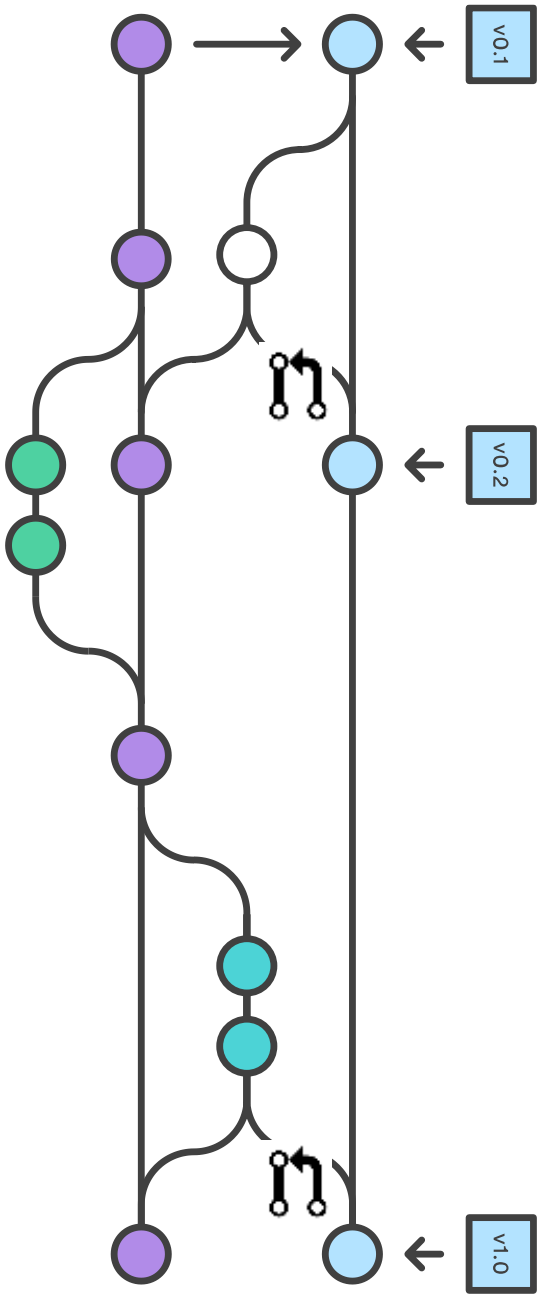
Technology

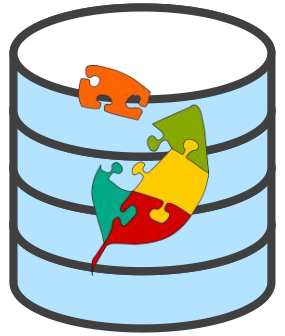
- Workflow github
- Organization OpenAlea
- Local fork or clone of a given project
- Local workflow git
- Merge requests



Personal fork or clone

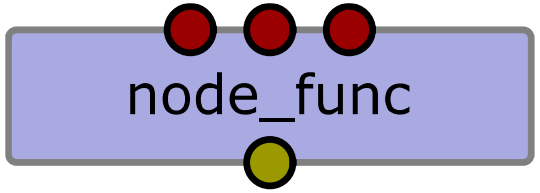




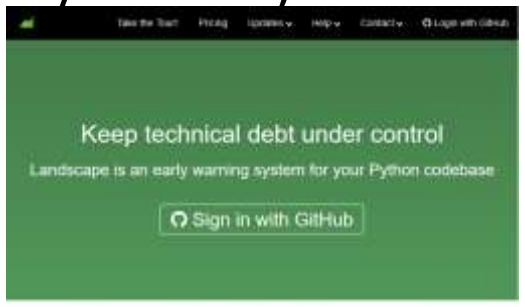


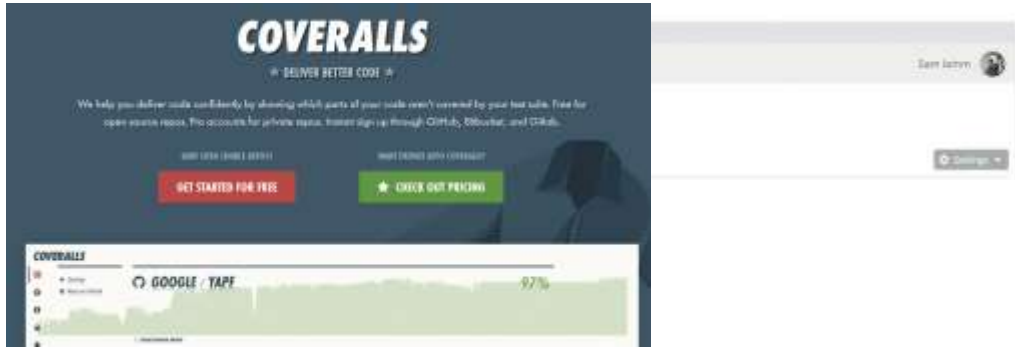
Openalea Project repository

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

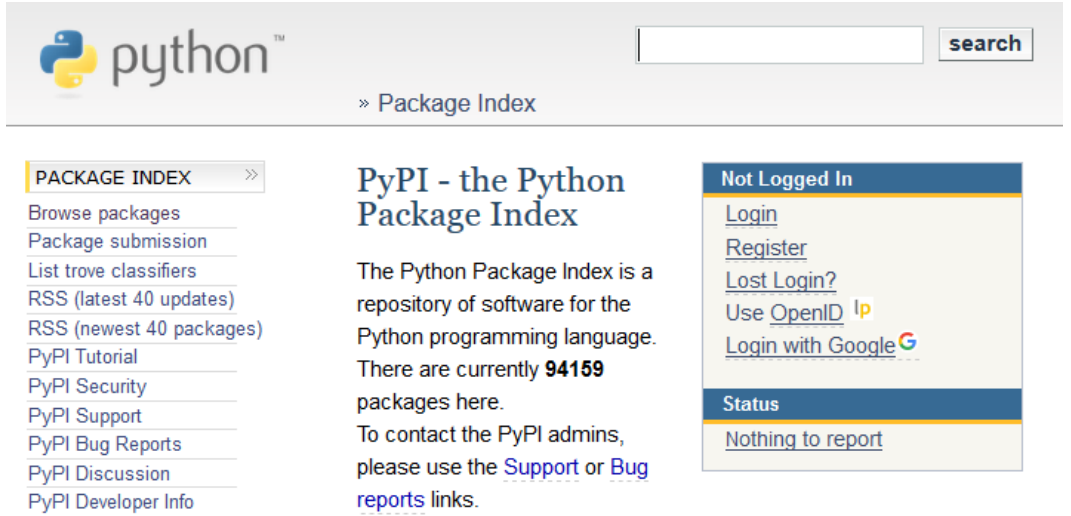




Technology

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

Google



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]

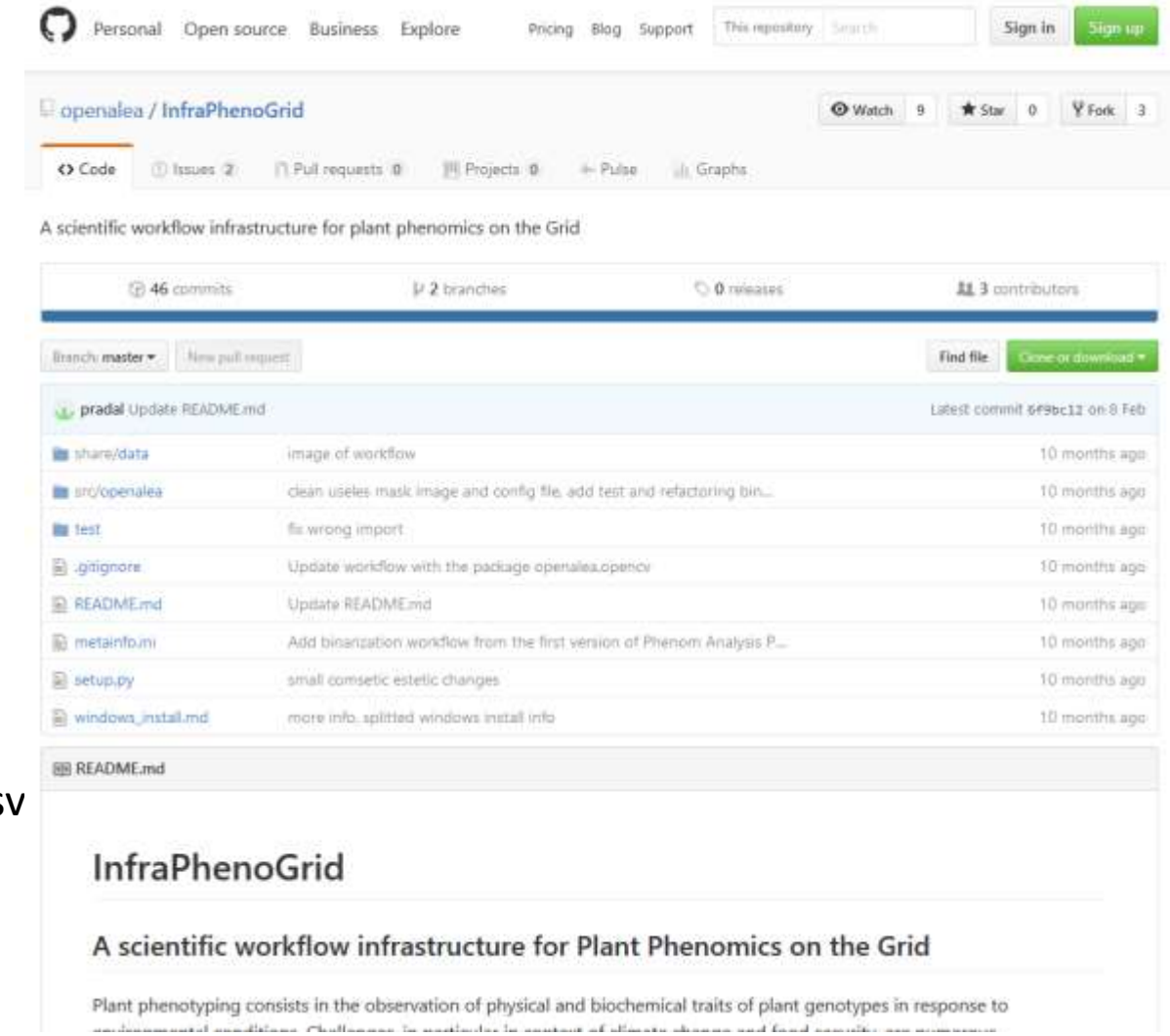
[a]*CIRAD, UMR AGAP, Montpellier, France*
[b]*Inria, VirtualPlants, Montpellier, France*
[c]*INRA, UMR459, LEPSE, F-34060 Montpellier, France*
[d]*INRA, UMR729, MISTEA, F-34060 Montpellier, France*
[e]*Inria, Zenith, Montpellier, France*
[f]*ICCH, NHLI, Imperial College London, UK*
[g]*Laboratoire de 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

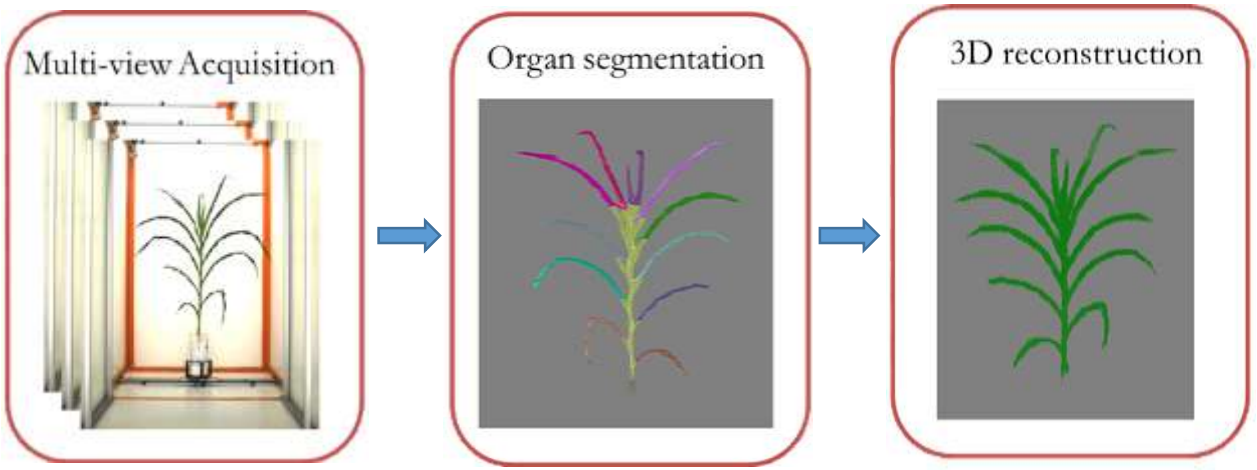
Demo redo

- 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



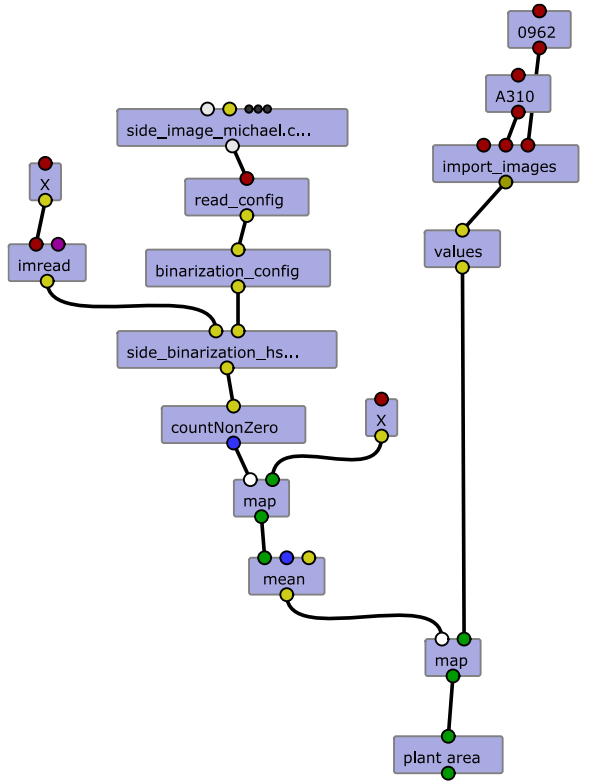
Demo reuse: Physio GWAS

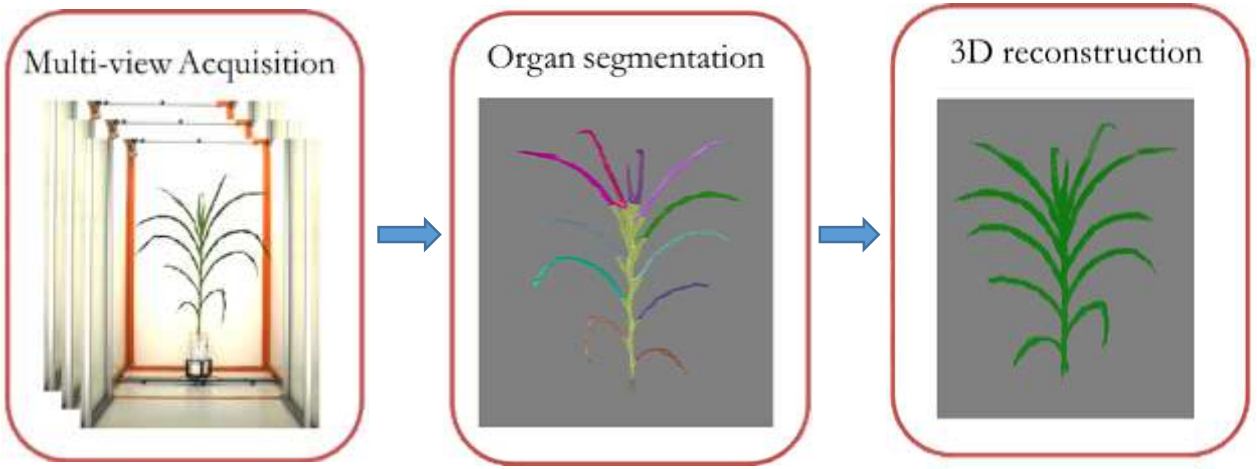
- Crop plant breeding
- Finding the right trait
- Phenome, Pheno-Arch
- NGS
- Physiology modeling



Physio GWAS

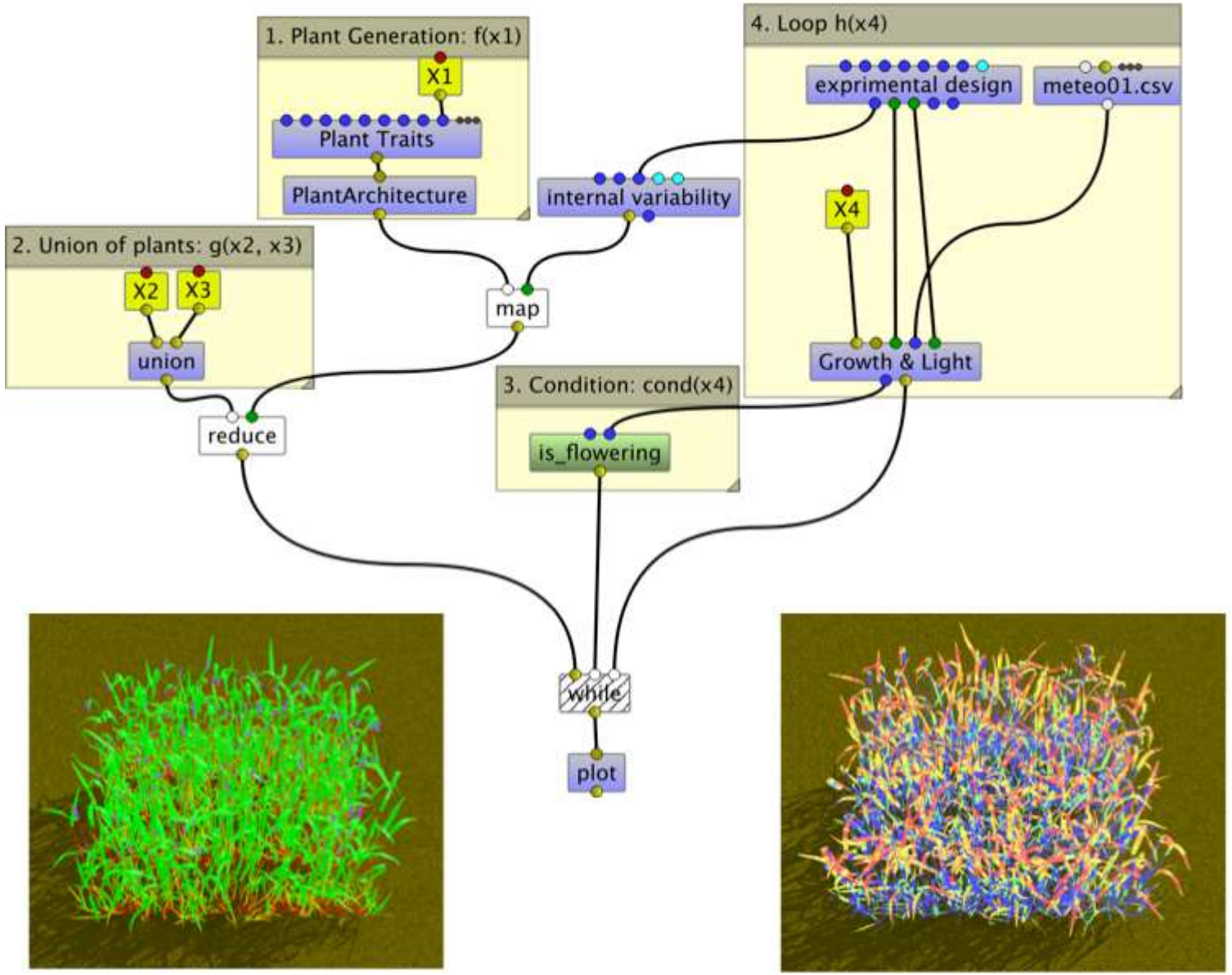
- Crop plant breeding
- Finding the right trait
- Phenome, Pheno-Arch
- NGS
- Physiology modeling





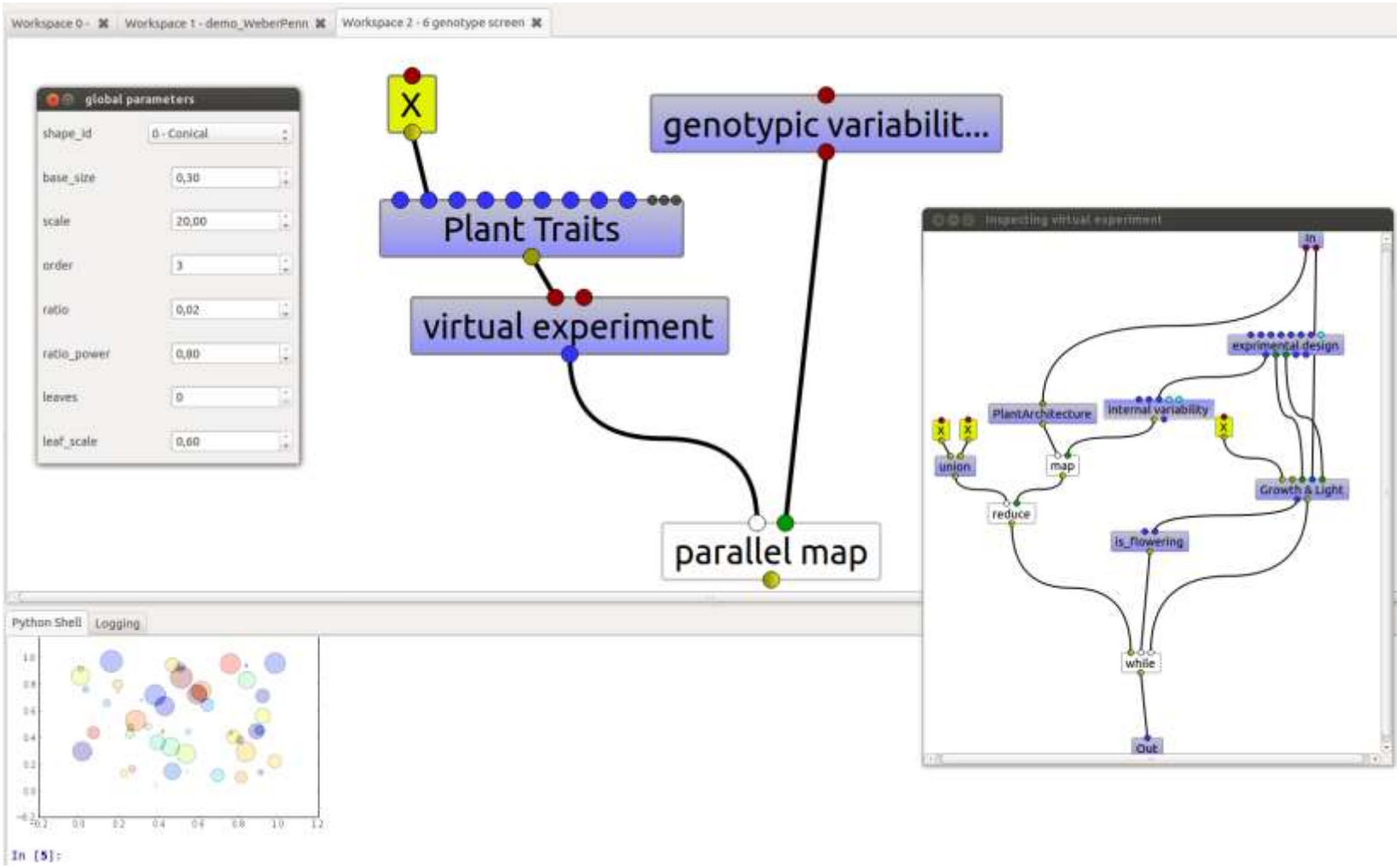
Physio GWAS

- 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 Fournier[3], Bruno Andrieu[2], Vianney Houlès[1], Corinne Robert[2] and Christophe Pradal[4,5]

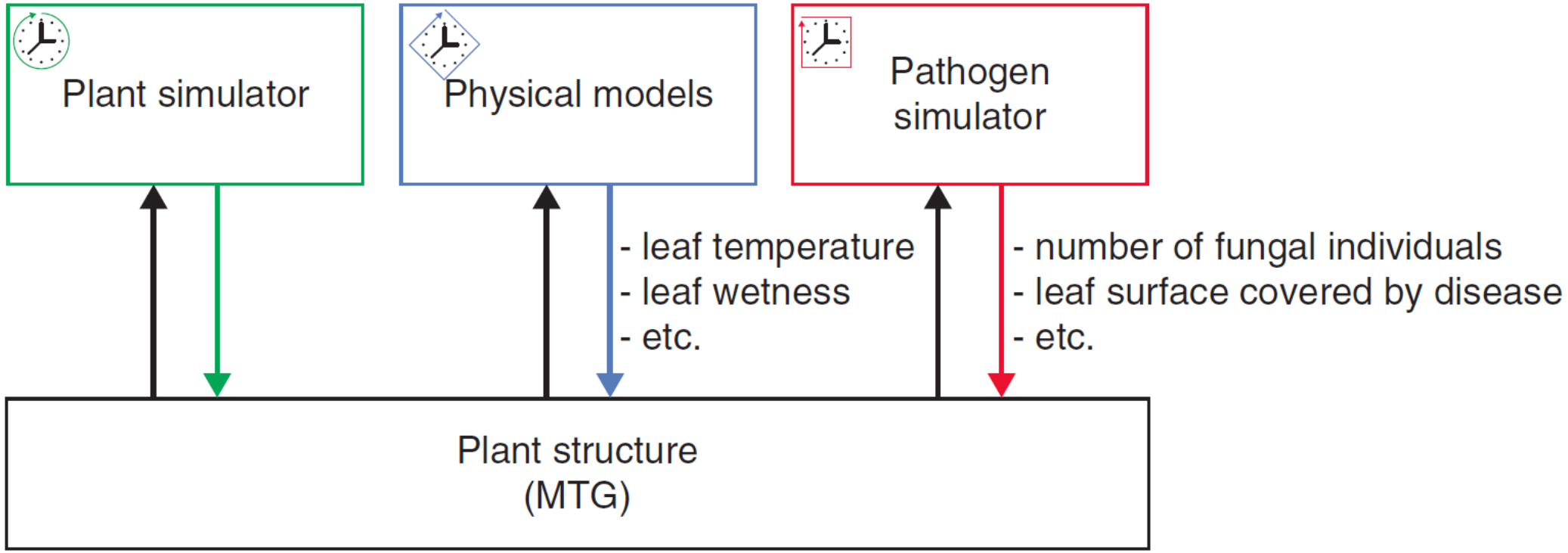
[1]*ITK, avenue de l'Europe, F-34830 Clapiers, France,* [2]*INRA, UMR 1091 EGC, F-78850 Thiverval-Grignon, France,* [3]*INRA, UMR 759 LEPSE, F-34060 Montpellier, France,* [4]*CIRAD, UMR AGAP and INRIA, Virtual Plants, F-34398 Montpellier, France and* [5]*Institut de Biologie Computationnelle, F-34095 Montpellier, France*
** For correspondence. E-mail guillaume.garin@itkweb.com*

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

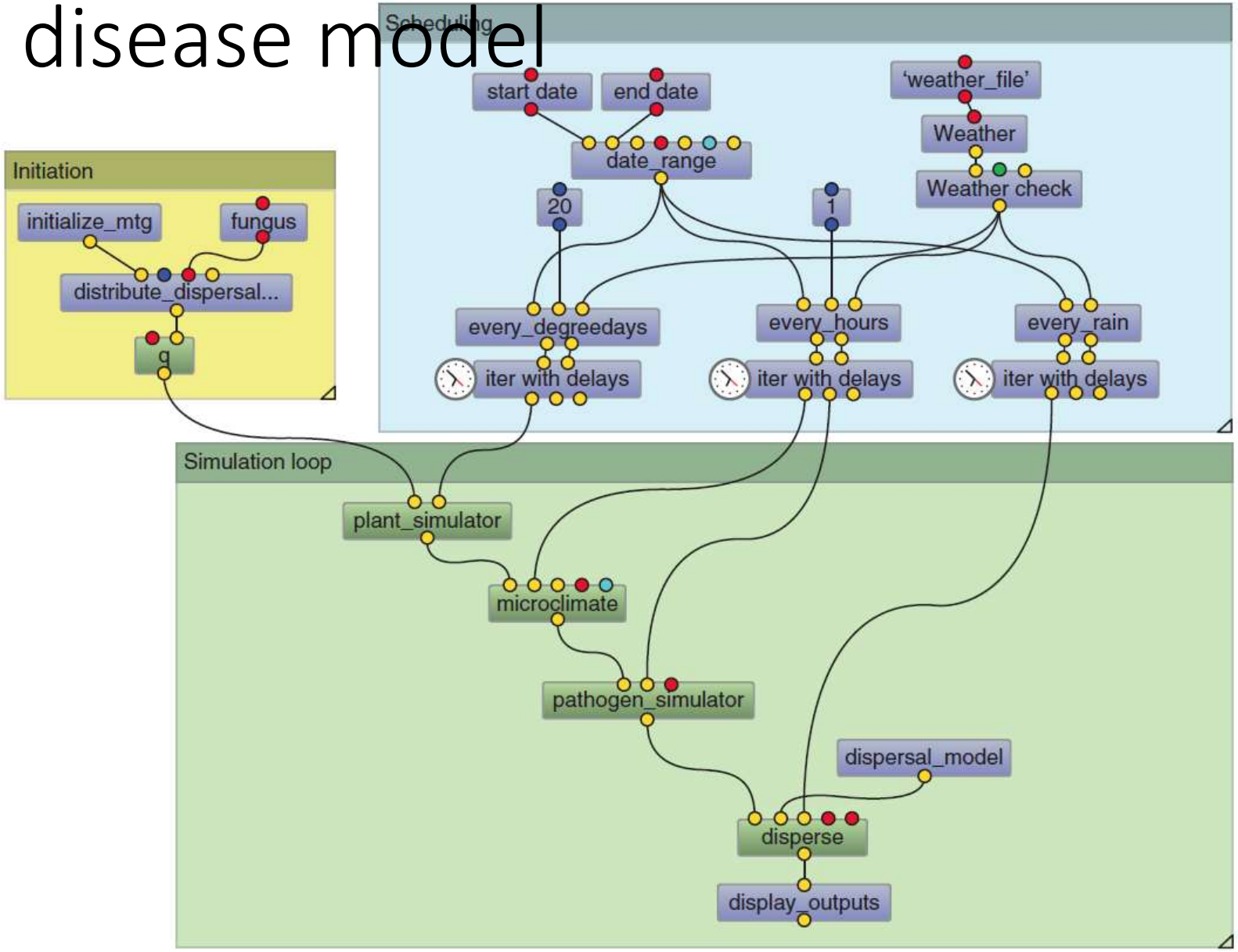
• **Background and Aims** Sustainable agriculture requires the identification of new, environmentally responsible strategies of crop protection. Modelling of pathosystems can allow a better understanding of the major interactions

Demo reuse: disease model

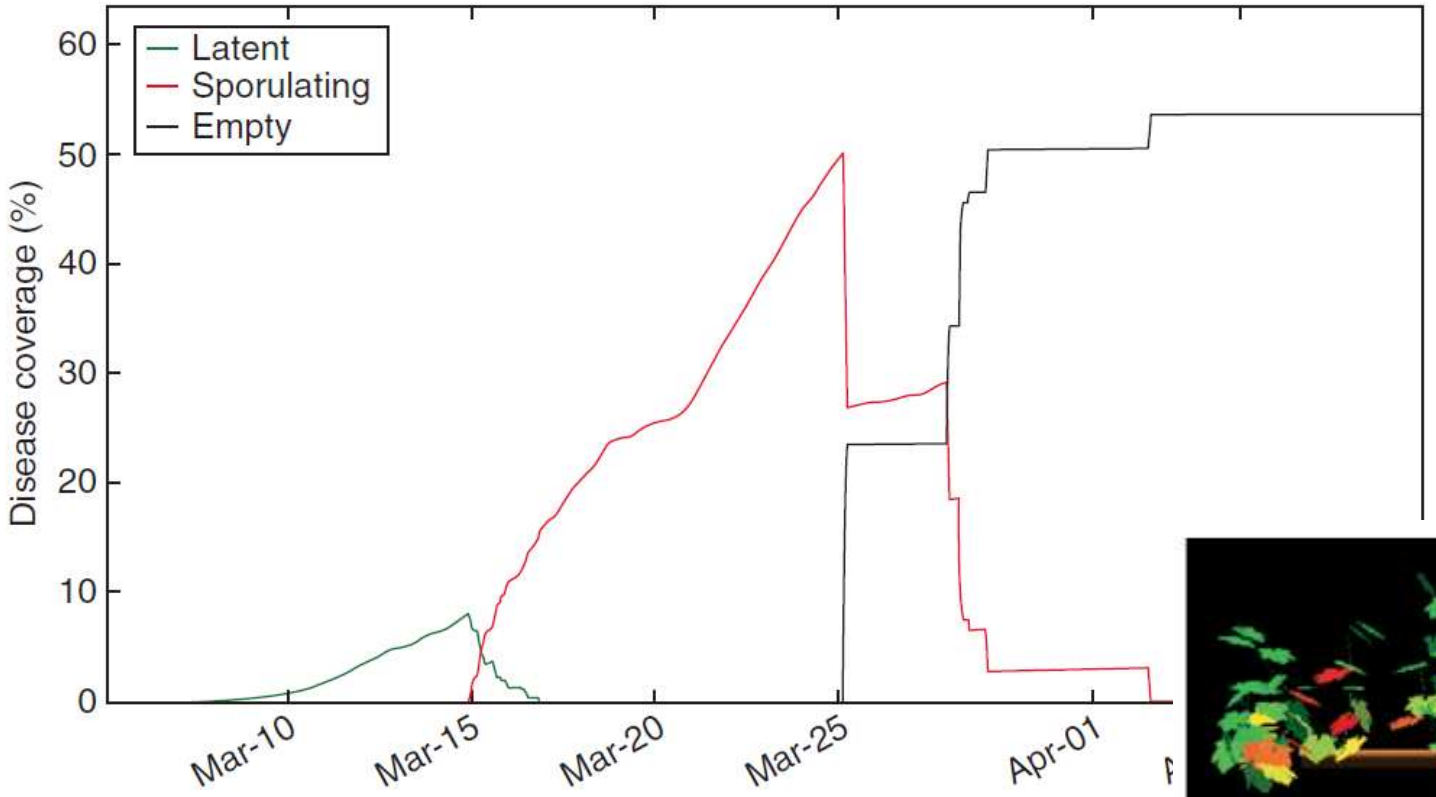
Garin et al. — *Modelling framework to simulate foliar fungal epidemics*

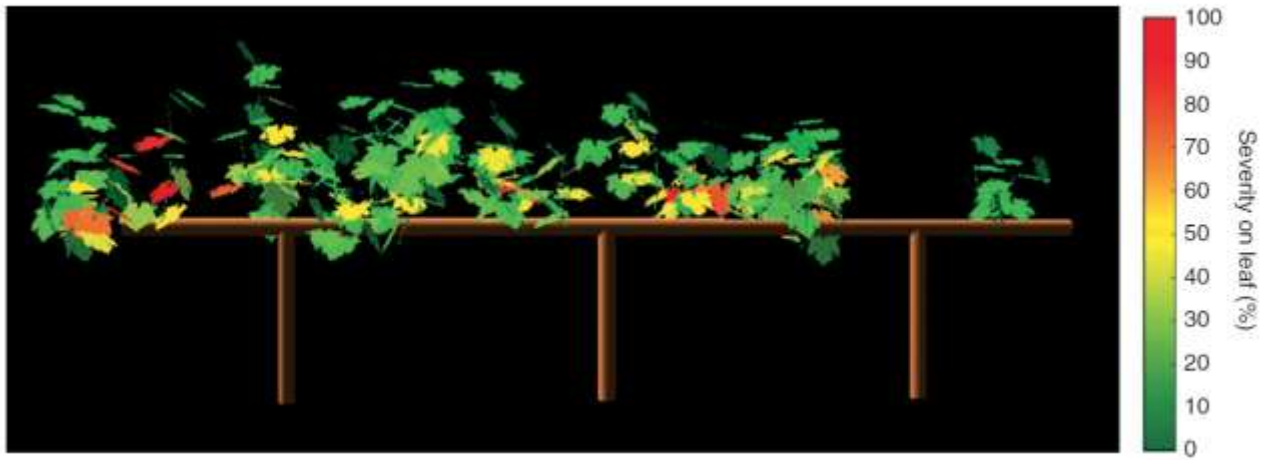


Demo reuse: disease model



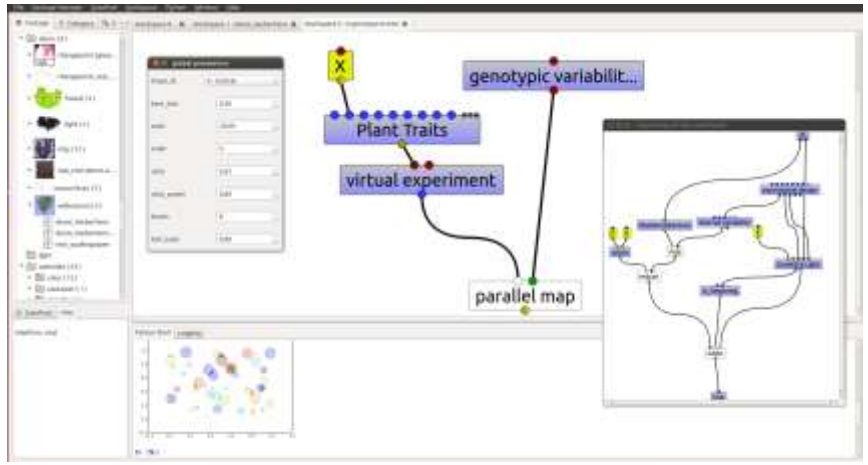
Demo reuse: disease model





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