

Root System Markup Language: Toward a Unified Root Architecture Description Language^{1[OPEN]}

Guillaume Lobet², Michael P. Pound², Julien Diener², Christophe Pradal², Xavier Draye*, Christophe Godin, Mathieu Javaux, Daniel Leitner, Félicien Meunier, Philippe Nacry, Tony P. Pridmore, and Andrea Schnepf

PhytoSYSTEMS, Université de Liège, 4000 Liege, Belgium (G.L.); Centre for Plant Integrative Biology, School of Biosciences, University of Nottingham, Sutton Bonington LE12 5RD, United Kingdom (M.P.P.); Virtual Plants, Inria, Cirad, Institut National de la Recherche Agronomique, 34095 Montpellier, France (J.D., C.P., C.G.); Institut de Biologie Computationnelle, F-34095 Montpellier, France (C.P.); Earth and Life Institute, Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium (X.D., M.J., F.M.); Institut für Bio- und Geowissenschaften: Agrosphäre, Forschungszentrum Jülich, D-52425 Jülich, Germany (M.J., A.S.); Computational Science Center, University of Vienna, 1090 Vienna, Austria (D.L.); Biochemistry and Plant Molecular Physiology, Unité Mixte de Recherche 5004 Centre National de la Recherche Scientifique/Institut National de la Recherche Agronomique/SupAgro-M/UM2, Institut de Biologie Intégrative des Plantes, 34060 Montpellier cedex 1, France (P.N.); and School of Computer Science, University of Nottingham, Nottingham NG8 1BB, United Kingdom (T.P.P.)

ORCID IDs: 0000-0002-5883-4572 (G.L.); 0000-0002-8214-6411 (J.D.); 0000-0002-6168-5467 (M.J.); 0000-0001-7766-4989 (P.N.).

The number of image analysis tools supporting the extraction of architectural features of root systems has increased in recent years. These tools offer a handy set of complementary facilities, yet it is widely accepted that none of these software tools is able to extract in an efficient way the growing array of static and dynamic features for different types of images and species. We describe the Root System Markup Language (RSML), which has been designed to overcome two major challenges: (1) to enable portability of root architecture data between different software tools in an easy and interoperable manner, allowing seamless collaborative work; and (2) to provide a standard format upon which to base central repositories that will soon arise following the expanding worldwide root phenotyping effort. RSML follows the XML standard to store two- or three-dimensional image metadata, plant and root properties and geometries, continuous functions along individual root paths, and a suite of annotations at the image, plant, or root scale at one or several time points. Plant ontologies are used to describe botanical entities that are relevant at the scale of root system architecture. An XML schema describes the features and constraints of RSML, and open-source packages have been developed in several languages (R, Excel, Java, Python, and C#) to enable researchers to integrate RSML files into popular research workflow.

¹ This work was supported by the Belgian Science Policy Interuniversity Attraction Poles Program (grant no. P7/29 to G.L. and X.D.), by the Fonds National de la Recherche Scientifique (to G.L. and F.M.), by the French National Research Agency (HydroRoot project, grant no. ANR-11-BSV6-0018 to C.P. and C.G.), by the European Union FP7-KBBE-2011-5 project (EURoot: Enhancing Resource Uptake from Roots under Stress in Cereal Crops to P.N., T.P.P., M.P.P., and X.D.), by the Agropolis Foundation of Montpellier, France (Rhizopolis grant to P.N., C.P., J.D., and C.G.), by the German Research Association (Transregio Collaborative Research Center 32, Patterns in Soil-Vegetation-Atmosphere Systems: Monitoring, Modeling, and Data Assimilation to A.S.), and by the Austrian Academy of Sciences (APART fellowship at the Computational Science Center at the University of Vienna to D.L.).

² These authors contributed equally to the article.

* Address correspondence to xavier.draye@uclouvain.be.

The author responsible for distribution of materials integral to the findings presented in this article in accordance with the policy described in the Instructions for Authors (www.plantphysiol.org) is: Xavier Draye (xavier.draye@uclouvain.be).

[OPEN] Articles can be viewed without a subscription.

www.plantphysiol.org/cgi/doi/10.1104/pp.114.253625

By securing access to water and nutrients, root systems are generally recognized as having a critical influence on plant productivity (Lynch, 1995). As an example, in maize (*Zea mays*), historical increases in yield in the U.S. Corn Belt have been linked to an increase in root system size (Hammer et al., 2009). Tailoring root architecture, therefore, is thought to be a critical step toward dealing with extreme environmental conditions such as drought (Comas et al., 2013; Lobet et al., 2014) or nutrient-poor soils (Lynch, 2007; Postma and Lynch, 2011).

While precise root system architecture characterization methods have been studied in woody plants for many years (Danjon et al., 1999, 2013; Danjon and Reubens, 2007), physiological studies on smaller plants (e.g. *Arabidopsis* [*Arabidopsis thaliana*]) have often neglected detailed root architecture, using mainly global estimators, such as total root length or the maximal depth of the root system convex hull (Galkovskiy et al., 2012). However, an increasing number of research questions now require a precise quantification of root

system architecture. As an example, nutrient or water deficiencies can have strong effects on root development (Al-Ghazi et al., 2003; Hammer et al., 2009; Péret et al., 2012; Gruber et al., 2013; Kellermeier et al., 2014), and only accurate root reconstruction allows the quantification of these effects. In addition, functional structural plant models are becoming increasingly popular to investigate the belowground ecophysiology of crops (Draye et al., 2010; Comas et al., 2013; Dunbabin et al., 2013; Lobet et al., 2014), and these models require a precise quantification of root system architecture, either to evaluate root developmental parameters or as a direct model input.

Root system architecture is generally described at three main levels (Godin and Sinoquet, 2005; Lynch, 2007; Postma and Lynch, 2011). The geometry of a root system describes the physical position, in space and time, of its component root axes. The topology describes the root system as a network and can be seen as the backbone, or skeleton, of the root system. Finally, the successive segments that constitute individual root axes can be further characterized by their properties, such as the local root diameter and color or the presence/absence of root hairs.

While these levels can be easily represented for simple root systems or single roots (Fig. 1A, I), the complexity of the representation can dramatically increase for complex root systems and branched root axes (Fig. 1A, II and III). In addition, while root system analysis is classically performed on two-dimensional (2D) images of root projections, recent years have seen the development of three-dimensional (3D) acquisition devices (Danjon et al., 1999, 2013; Danjon and Reubens, 2007; Iyer-Pascuzzi et al., 2010; Clark et al., 2011; Mooney et al., 2012), which solve the issues of object occlusion yet increase the complexity of the root system description (Fig. 1B). Finally, the recourse to dynamic traits and

the tracking of individual roots in root development studies require elaborated time series data representation (Fig. 1C).

The past few years have seen the development of a variety of solutions for the analysis of root system images (for an updated listing, see Lobet et al., 2013). Several of these solutions deal with root architecture per se and consider explicitly the morphological and topological properties of the root system (Table I). Such a variety of software solutions reflects the coexistence of complementary approaches to the analysis of root systems. As a direct consequence of this diversity, many independent root system architecture representation and storage methods have been implemented, leading to multiple data sets lacking common structure, which restricts the possibilities to compare root system architecture structures or measurements obtained using different tools or to validate new algorithms.

The Multi-Scale Tree Graph formalism (MTG; Godin and Caraglio, 1998) is a formalism used to represent the topology and the geometry of any type of plant architecture at different levels of organization. This formalism has become a de facto standard in the plant architecture community, encoding plant architecture and its development (Godin et al., 1999, 2005; Danjon and Reubens, 2007; Griffon and de Coligny, 2014) for a wide variety of plant architectures, such as root systems (Danjon et al., 1999), annual plants (Mündermann et al., 2005; Fournier et al., 2010), and fruit trees (Guédon et al., 2001; Negrón and Contador, 2013). In recent years, computational and mathematical models of growth, branching, and architecture have been developed around this formalism on the basis of qualitative botanical knowledge (for review, see Barthélémy and Caraglio, 2007). The MTG is the central data structure of the OpenAlea platform for functional-structural plant models (Fournier et al., 2010; Boudon

Figure 1. Root system architecture description. A, Single root image. I, Single root; II, single root axis; III, root system (e.g. for monocots). B, 3D image stack. C, Time series. D, Example of parameters used to describe root architecture. First order roots are shown in red, and second order roots are shown in green.

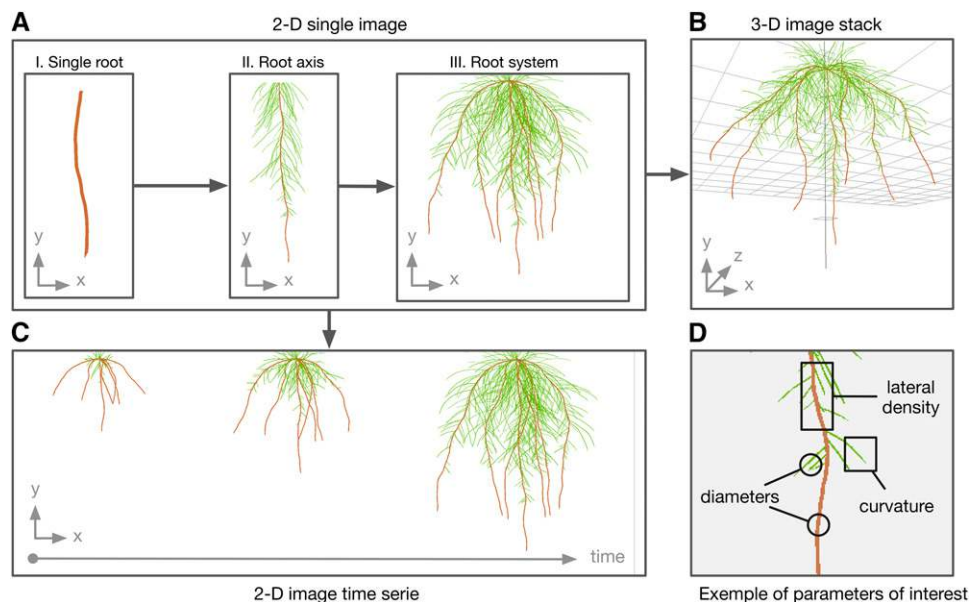


Table 1. Description of existing root system architecture image analysis tools

Dashes indicate that the data are not stored in an external format.

| Software | Automation | Image Type | Storage | Topology | Diameter | Time Series | RSML Support | Reference |
|--------------------|---------------|------------|---------|----------|----------|-------------|--------------|-------------------------|
| ARIA | Automated | 2D | XLS | Yes | No | No | No | Pace et al. (2014) |
| EZ-Rhizo | Automated | 2D | SQL | Yes | No | No | No | Armengaud et al. (2009) |
| DART | Manual | 2D | TXT | Yes | No | No | No | Le Bot et al. (2010) |
| OpenAlea.RhizoScan | Automated | 2D | MTG | Yes | No | Yes | Yes | Diener et al. (2013) |
| RootNav | Semiautomated | 2D | – | Yes | No | No | Yes | Pound et al. (2013) |
| RootReader2D | Automated | 2D | XML | Yes | No | No | No | Clark et al. (2013) |
| RootReader3D | Automated | 3D | XML | Yes | No | No | No | Clark et al. (2011) |
| RootSystemAnalyser | Automated | 2D | MAT | Yes | Yes | Yes | Yes | Leitner et al. (2014) |
| RooTrak | Automated | 3D | – | Yes | No | No | Yes | Mairhofer et al. (2012) |
| RootTrace | Automated | 2D | – | Yes | No | Yes | In progress | French et al. (2009) |
| SmartRoot | Semiautomated | 2D | XML | Yes | Yes | Yes | Yes | Lobet et al. (2011) |

et al., 2012; Garin et al., 2014) that eases communication between different models developed by different research groups. To achieve that level of genericity, MTGs do not assume any specific type of plant architecture ontology and can be adapted to each new protocol in a flexible manner. However, this flexibility may induce additional complexity when exchanging data between different groups of scientists using different protocols and software within a specific domain. For each new protocol, for example, a modeler must define the number of scales, their meaning, the name of the attributes, and how to encode them. While OpenAlea provides software solutions to manage this complexity, external software must implement and manage this complexity in their own programming environment and language.

Now, a new step must be taken to further improve the ability of researchers to acquire plant architecture data using different software and to exchange and share these data. To achieve this, the research community needs to agree on a common biological language to build up shared databases and quantitative tools and to compare hypotheses and approaches. This is unfeasible at the level of genericity that was used in the design of MTGs. However, genericity can be achieved at the level of particular plants, plant parts (such as roots), or applications by developing specific ontologies on the top of MTGs.

This work introduces the Root System Markup Language (RSML), a unified language that enables root system architecture information storage based on the MTG formalism and XML standards. RSML aims (1) to accommodate the richness of root system architecture types and complexities (2D, 3D, and time series) and (2) to be open, cross platform, and easy to implement in new tools and software.

At the time of writing, RSML support has been implemented into the following imaging or modeling suites: ArchiSimple (Pagès et al., 2014), OpenAlea (Pradal et al., 2008), RhizoBox (Leitner et al., 2010), RhizoScan (Diener et al., 2013), RootNav (Pound et al., 2013), RooTrak (Mairhofer et al., 2012, 2013), RootSystemAnalyser (Leitner et al., 2014), R-SWMS (Javaux et al., 2008), and SmartRoot (Lobet et al., 2011).

RESULTS AND DISCUSSION

Description of the RSML Format

The RSML format defines an XML file in which the topological, geometrical, and numerical information describing a root system is stored.

In practice, the RSML format is split into metadata and the scene elements (Fig. 2). The metadata store experimental and technical information relevant to the scene, while the scene itself contains the root system representation. Supplementary data can be stored using properties (at the scene, plant, or root level), functions along the root axes, and annotations.

A brief outline of the main components of the RSML format is presented in the following sections. The RSML Web site (<http://rootssystemml.github.io/>) provides technical specifications of the format as well as RSML examples and related image files that illustrate the possibilities offered by the RSML format. It also includes links to all software reading and writing RSML files. Supplemental Data S1 contains a simple example of an RSML file.

Metadata

The metadata specify the experimental context in which an RSML file was constructed. They also provide a concise description of the file content, allowing documents to be quickly scanned, filtered, or classified without reading the entirety of each file.

Included in the metadata element are a unique identifier for the scene, file information, and the real-world unit and resolution for the root conversion of pixel data. The <software> and <last-modified> elements allow tracking of changes when multiple software tools have handled the same document. The <property> and <function> definitions describe the associated properties or functions that will appear in the document. If the RSML file is one of many in a time series data set, a <time-sequence> element identifies this set and the position of the file in the sequence.

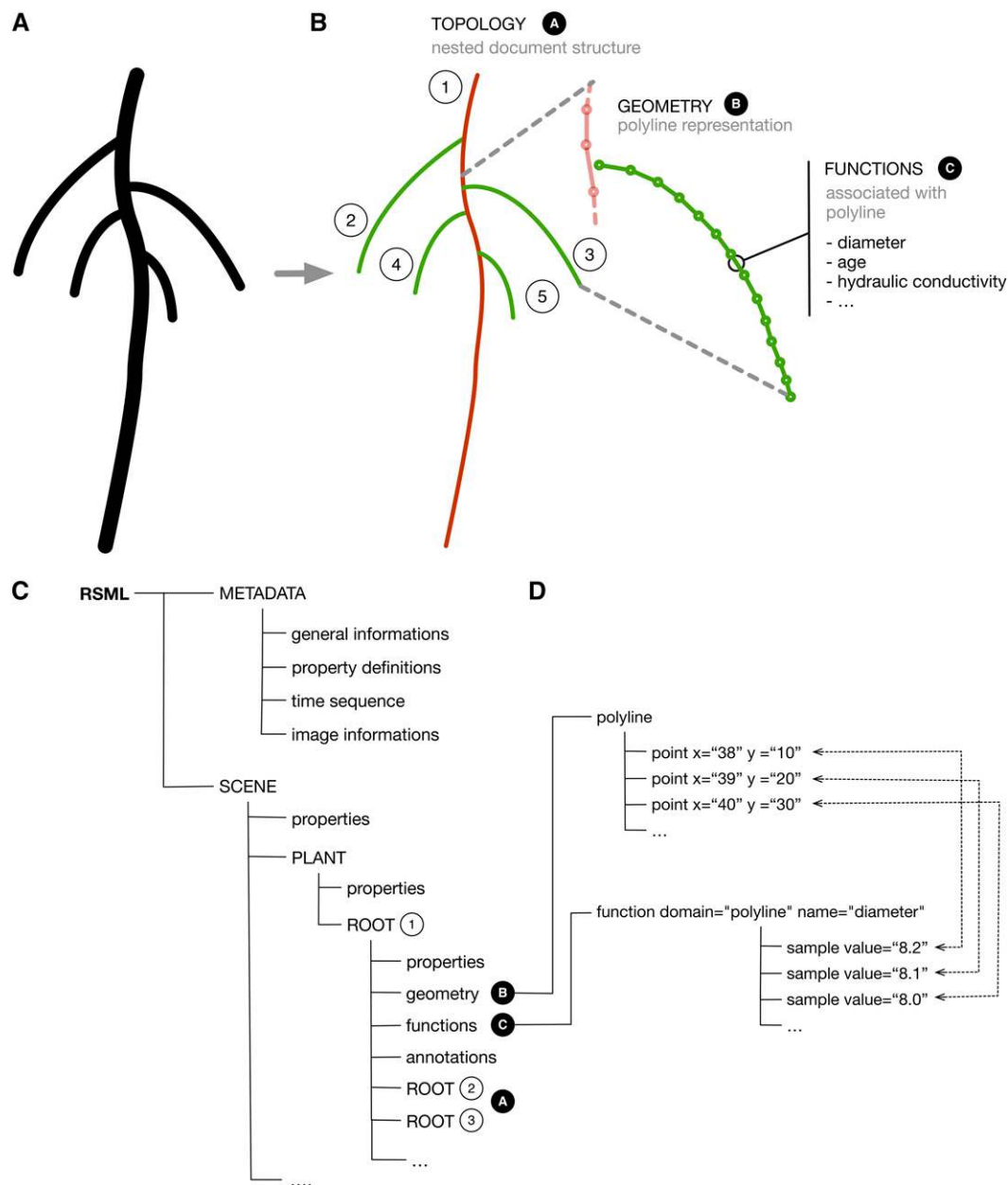


Figure 2. Visual representation of the RSML structure. A, Original image. B, Graphical representation of the structure. Topology (primary root in red and lateral roots in green), geometry, and properties are represented at different levels. C, Schematic representation of an RSML file structure. D, Representation of the coupling between the root geometry and its associated functions (here, the diameter). Dotted lines represent data from the same point in a polyline.

Scene and Topology

The scene represents a single image, possibly within a series of 2D or 3D images, that will contain at least one root system. Within the scene, there is at least one plant containing at least one root. Finally, roots may also contain additional child roots (i.e. lateral roots). RSML documents mimic this structure (Fig. 2): the scene element will contain one or more plant elements, which in turn will contain one or more root elements. Further levels of root elements are used to show lateral

roots of higher orders (Fig. 2C). Therefore, the topological aspects of the root system, such as the connections between a primary root and its child lateral roots, are represented through the nested structure of the scene element in the RSML document itself. All geometry and other measurements are stored as data attached to the relevant elements throughout the document (Fig. 2C).

In order to ensure a uniform naming of the different root types and terms across different files, root type

descriptors used in RSML refer to the Plant Ontology Database (Plant Ontology Consortium, 2002; Avraham et al., 2008), a plant ontology widely accepted within the plant community. The current root type terminology in the Plant Ontology Database is presented in Table II. This list is not exhaustive and might be extended with other terms from the Plant Ontology Database.

Root Geometry

While the topology of the scene is implicit in the hierarchy of the document, geometric information is defined explicitly. Root elements contain a geometry element in which the root geometry is detailed as a polyline, a succession of linear segments. Scenes and plants contain no geometrical information; their geometries are the combined geometries of all child root elements.

RSML has been designed to allow the sharing of root architecture between different software packages, where these systems may contain different representations of root geometry. We use the polyline as the primary geometric structure (each root element must include a polyline geometry). Each software package that makes use of RSML is responsible for the conversion between a polyline and any alternative structures that the software may use, such as continuous splines. Inside the geometrical description of a root, a polyline element contains an ordered list of points that provide the end points of the successive segments that make up the root. Each point element contains *x* and *y* attributes for 2D scenes and an optional *z* attribute for 3D architectures. All geometric units in RSML are given in pixel coordinates, referring directly to the image associated with this root system. RSML metadata can be used to provide the scaling necessary to convert into real-world units.

It may be the case that the conversion to polyline form comes at the expense of accuracy in a given software package. Should a certain application store geometry in spline form, for example, the conversion to a polyline will only approximate the curve. In this case, additional geometries are permitted alongside, but not instead of, the polyline. These can take any form as long as they are contained within a single child

of the geometry element. It should be noted that additional geometry types are included for the convenience of individual software developers. Other software that reads the RSML format need only examine the polyline form of the geometry and may disregard any additional information. In this way, the portability of geometric information between software is ensured, but more specific structures are available if the RSML files are being used as a storage format for a particular application.

Root Functions

It will often be desirable to attach additional information along the polyline. This would be the case of root diameter, root age, root hair length, or the presence of nodules. In RSML, continuous functions are used to describe quantitative information as a function of the longitudinal position along the root axis. The function domain is explicitly defined and specifies the mapping between observed function values and their corresponding positions along the root (Supplemental Fig. S1). Depending on the software implementation, the sample points of a function can be uniformly spread over a root length or attached to a given position on the polyline, either using an index or a length.

Through the use of functions, quantifiable variables can be added within an RSML document. Information that is not directly associated with a root geometry, or categorical information that cannot be provided as a function, is instead stored in separate entities within the RSML specification that are described below.

Properties

Many aspects of root systems cannot be linked directly to root geometry and are instead related to botanical entities. For example, while diameter is intrinsically linked to a position along the root, qualitative (long or short root, dead root) or global (length, insertion angle, or position) information is better attached to the whole root as a property. A `<property>` element contained within a scene, plant, or root element specifies a measurement of that object. Properties can take one of numerous data types, allowing binary, integers, real types, or text values. As properties may be specific to the generating software, the meaning of a property can be supplied within the document metadata. Properties also might be used for a more efficient parsing of the RSML document, enabling analysis tools to directly retrieve these precalculated properties without having to compute them from the topological and geometrical information.

Annotations

It may be useful to attach general information such as user observations to a given scene. For example, a region of a given scene could be marked as out of

Table II. Plant ontology terms currently used in the RSML format

This list is not exhaustive, as any term contained in the Plant Ontology Database (www.plantontology.org) could be used.

| Plant Ontology Name | Plant Ontology Accession |
|---------------------|--------------------------|
| Root | PO:0009005 |
| Basal root | PO:0025002 |
| Embryo root | PO:0000045 |
| Lateral root | PO:0020121 |
| Primary root | PO:0020127 |
| Shoot-borne root | PO:0000042 |
| Tuberous root | PO:0025523 |

focus, letting software know that image analysis performed in this area may be less reliable. As properties, annotations are added as elements located within the corresponding scene, plant, or root element. Each annotation includes a list of one or more points, representing the point, line, or region of interest to which the annotation applies. A `<value>` element provides the text or numerical content of the annotation, and finally, a `<software>` element specifies the software used to add that annotation.

Root Development

Root developmental processes (e.g. growth rate) are often analyzed using time-lapse image sequences. Preserving time series information is an important factor in many root phenotyping applications and requires maintaining an explicit link between successive images of the same plant. The RSML format allows images of a time-lapse sequence to be linked through the use of the `<time-sequence>` element in the metadata. The `<index>` element indicates its position in the time series.

Software that provides an explicit mapping between geometries in a time series can use the metadata to indicate this. Some software use such information to calculate a change in parameters over time (e.g. elongation rate). Others (SmartRoot and RootSystemAnalyser) use previous time-step information to initialize subsequent reconstructions, improving root-tracing efficiency by focusing on incremental information in the following image.

RSML Thesaurus

The RSML format does not impose a restricted set of properties or functions. However, a thesaurus has been defined to promote the use of standardized terms. Any supporting software is not required to process these terms. So, unlike the main RSML definition, new terms may be added to this thesaurus without changing the format itself. Addition to the thesaurus follows a traditional open-source protocol as described on the RSML Web site (<http://rootssystemml.github.io>).

Open-Source Packages for RSML Analysis

Five open-source application programming interfaces have been created to read and parse RSML data files from within C#, Excel (Fig. 3A), R (Fig. 3B), ImageJ (Fig. 3C), and Python (Fig. 3D). The aim of these packages is not to carry the analysis of the root system data but to provide end users with commonly used data structures within popular data-analysis pipelines. These packages have been released as open source to allow users to adapt them to their needs. They are available through the RSML Web site.

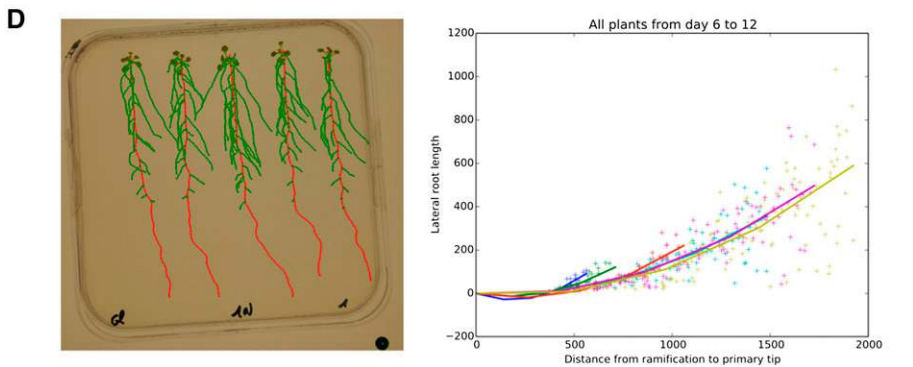
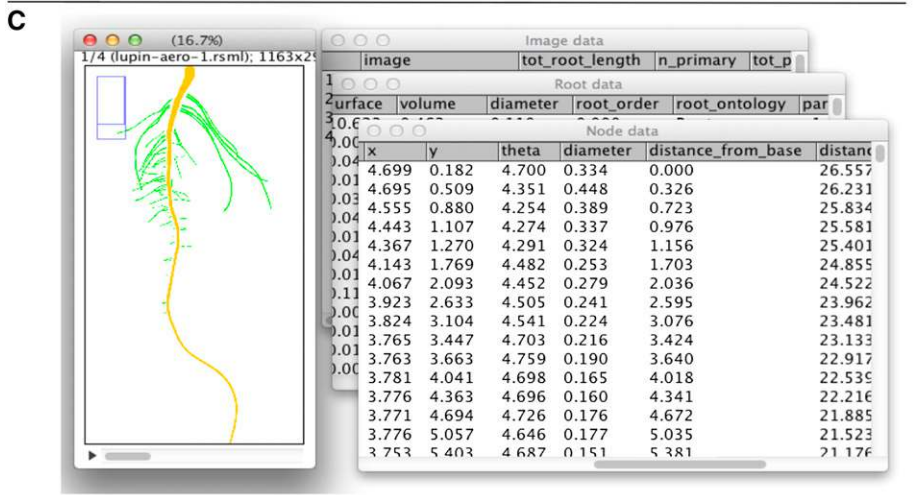
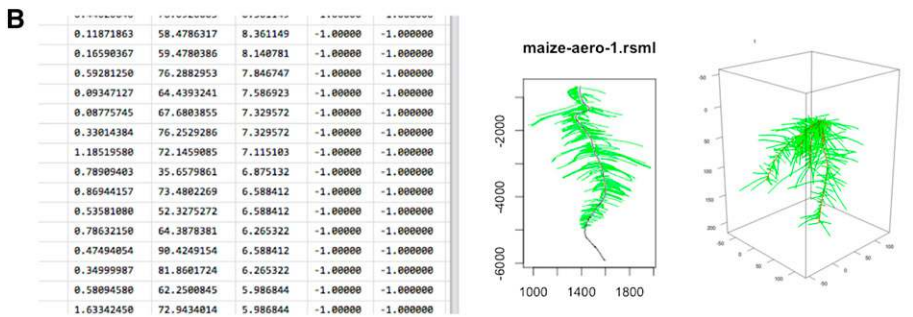
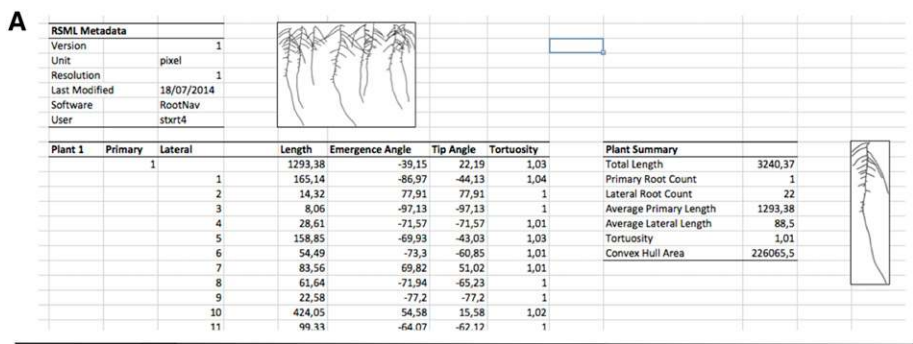
RSML Enables Common Pipelines for Root System Analysis and Modeling

The RSML format provides plant researchers with a central paradigm connecting image analysis tools, data-analysis pipelines, and modeling platforms (Fig. 4). Data generated by RSML-compliant tools can be reused in others, facilitating data transfer between researchers and groups. We provide here three examples where RSML is used to interface with different analysis pipelines.

In the first example, RSML was used to transfer root architecture information between root image analysis tools (Fig. 5). A root image, containing several plants (Fig. 5A), was traced using RootNav (Fig. 5B). RootNav features an efficient root-tracing algorithm but does not compute a measure of root diameter along each root, a measurement that might be required in some experiments. The RSML file generated by RootNav was imported in SmartRoot, which automatically computes diameter measurements upon loading an RSML file that does not contain that information (Fig. 5C). The resulting RSML file was imported into the R statistical computing environment (R Core Team, 2008) for analysis. The profile of lateral root length along the primary root axis (Fig. 5D) was computed using the tracing originally performed by RootNav. The primary and lateral root diameter distributions (Fig. 5E) were computed using the data computed by SmartRoot. This example illustrates the complementarity of existent root image analysis tools and how the RSML format enables this complementarity to be exploited by researchers.

Today's science faces an increasing demand for reproducibility and standardized analysis pipelines. We believe that the existence of a standard format for root architecture will enable easier reproducibility among researchers and allow the comparison of multiple data sets, even those coming from different sources. In the second example, the image shown in Figure 6A was analyzed using RhizoScan, RootSystemAnalyser, RootNav, and SmartRoot. The RSML files generated by the different tools were exported into a single data file and analyzed using R (R Core Team, 2008). Supplemental Figure S2 shows the comparison of the measured primary root length, lateral root length, lateral insertion angles, and lateral insertion positions for each software. This example illustrates how a shared format can streamline the validation of new algorithms and the creation of benchmark data sets with which to validate them.

In the third example, we illustrate the use of RSML for data storage and sharing between modeling tools. Figure 6A shows the visual output of an *Anagallis femina* root system simulated by RootBox (Leitner et al., 2010). The simulated root architecture was stored as an RSML data file and converted into the MTG data structure (Godin and Caraglio, 1998) in the OpenAlea platform (Pradal et al., 2008). Taking advantage of the geometric modules of PlantGL (Pradal et al., 2009), the 2D or 3D convex hull of the root system can easily be calculated (Fig. 6B). The same RSML file was used in



R-SWMS (Javaux et al., 2008) to simulate water flow in the soil-root system, hence allowing the testing of the functional performance (in this case, water uptake) of the simulated root system.

These three examples highlight the potential role of the RSML format as a cornerstone in analysis pipelines and show how it can hasten data (both simulated and experimental) exchange between researchers.

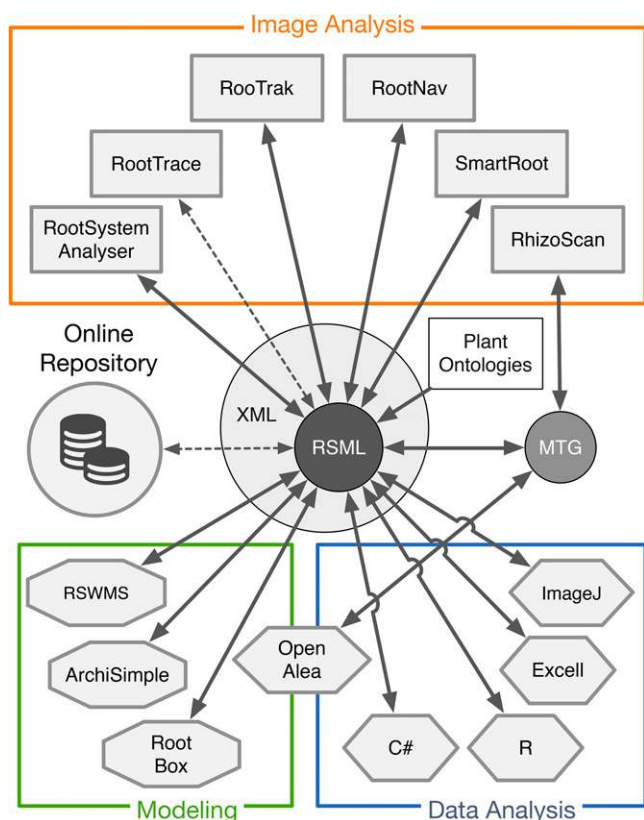


Figure 4. Analysis pipeline enabled by the RSML format. Dotted arrows represent connections that are not yet implemented.

RSML Promotes the Use of a Central Repository for Root Architecture Data

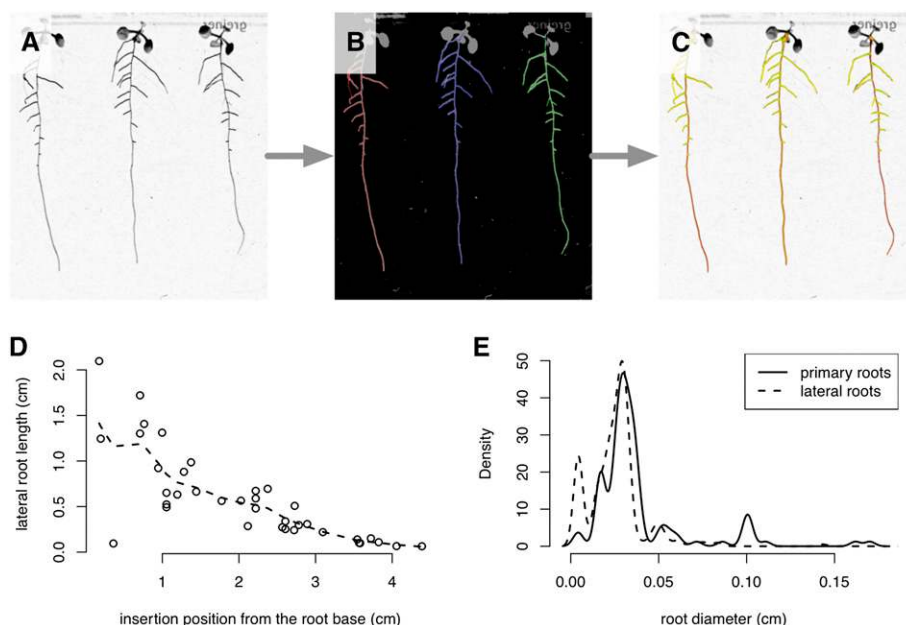
Many experiments on root architecture, and so a great number of software tools developed to analyze

them, focus on the limited number of root architecture parameters that can be calculated without an explicit root model. To extract other parameters of interest (e.g. lateral root length), a complete tracing of the root system is often required, including a hierarchical model of the root structure. However, the tracing of a complete root system can be time consuming. Therefore, it is highly beneficial to reuse previous root data sets in the quantification of other traits demanded in different experiments. Due to the lack of compatibility between many historic data sets, this reanalysis is often possible only by reconstructing the complete data set, which is, at best, time consuming and, at worst, impossible. By storing root architecture in a common format, desired root traits can be calculated quickly over large data sets captured with a variety of software, regardless of the traits that were considered when that data set was first analyzed.

We believe that the adoption of RSML will encourage the creation of central repositories for root architecture data, similar to those that exist in other domains. In molecular biology, it is even mandatory to upload gene expression data sets to a database such as EBI-ArrayExpress (Rustici et al., 2013) prior to publication. Those publicly accessible repositories are frequently queried by the scientific community. The development of a similar public database for costly and valuable root architecture data would increase the pace and efficiency of root research.

This central repository also can be used as a benchmark to compare and evaluate computer programs used to reconstruct architectural data. New algorithms and software could be assessed to ensure that the data sets produced are scientifically valid. This benchmark will also greatly accelerate the impact and the adoption of new, independently developed algorithms for the automatic digital reconstruction of root system architecture.

Figure 5. Example workflow enabled by the RSML format. A, Original image of Arabidopsis plants grown in a petri dish. B, Screenshot of a root tracing done using RootNav. C, The RSML generated by RootNav was opened using SmartRoot, which computed the root diameter (which is not calculated by RootNav). D, R-generated graph showing lateral root length depending on the insertion position from the primary root base. These data were computed by RootNav. The dashed line represents the moving average across the data set. E, R-generated histograms comparing the diameter of the primary and lateral roots. These data were computed by SmartRoot.



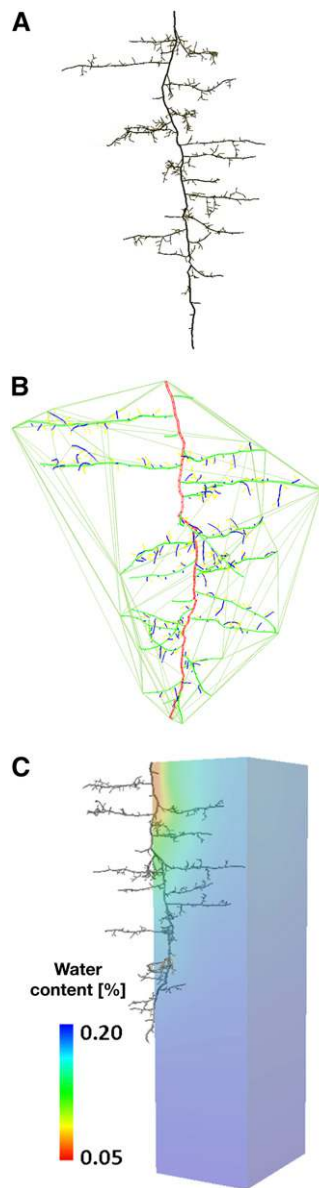


Figure 6. Using the RSML format to share data between modeling tools. A, Root system of *A. femina*, simulated using RootBox (Leitner et al., 2010). B, The RSML file exported by RootBox was opened in PlantGL (Pradal et al., 2009) to compute the convex hull of the root system (yellow lines). C, The same RSML file was used by R-SWMS (Javaux et al., 2008) to simulate soil water depletion in 24 h.

In neuroscience, the DIADEM challenge (for digital reconstructions of axonal and dendritic morphology) addresses a similar need (Parekh and Ascoli, 2013).

Using the RSML Format to Store Root System Data without Architectural Information

The RSML format was initially defined as an efficient storage mechanism for detailed root system representations. As such, the explicit topology and geometry of the root system can be encoded. However, it is important to

note that the RSML format also can be used for the storage of root system data that do not contain geometrical and/or topological information. As an example, the recently developed DIRT image-analysis toolbox (Bucksch et al., 2014) is able to extract multiple metrics from images of field-grown crown roots. The RSML format still could be used in this case by taking advantage of its underlying multiscale formalism (i.e. the fact that properties may be set at any level of detail, such as scene, plant, or root).

CONCLUSION

The RSML presented here facilitates the sharing of root architectures between software, experiments, and research groups. RSML accommodates a wide array of root architecture complexity (ranging from 2D projections of single roots to 3D representations of complete root systems) at varying levels of detail. The RSML format stores root topology (parent-child relationships), morphological properties (positions in space and time, length), and virtually any type of additional information used to describe root segments (e.g. diameter, color, and age) separately, linking them to form a coherent representation.

The RSML format is currently implemented within five root image analysis software (RhizoScan, RootNav, RooTrak, RootSystemAnalyser, and SmartRoot), three functional-structural root models (RootBox, ArchiSimple, and R-SWMS), and is based on the MTG format used in the OpenAlea platform. Five open-source packages were developed for the analysis and visualization of RSML data files (in C#, R, Excel, Java, and Python). We believe that the availability of RSML will encourage the creation of common analysis pipelines for root architecture information, enabling better data sharing between root researchers, and will facilitate the creation of a shared database of root architecture information.

A complete description of the RSML, along with examples and application programming interface packages, are available at <http://rootssystemml.github.io>.

Supplemental Data

The following supplemental materials are available.

Supplemental Figure S1. Schematic representation of the Root System Markup Language structure.

Supplemental Figure S2. Comparison of root system traits obtained with different tools.

Supplemental Data S1. Example of RSML data file.

Received November 13, 2014; accepted January 21, 2015; published January 22, 2015.

LITERATURE CITED

Al-Ghazi Y, Muller B, Pinloche S, Tranbarger TJ, Nacry P, Rossignol M, Tardieu F, Doumas P (2003) Temporal responses of Arabidopsis root

- architecture to phosphate starvation: evidence for the involvement of auxin signalling. *Plant Cell Environ* 26: 1053–1066
- Armengaud P, Zambaux K, Hills A, Sulpice R, Pattison RJ, Blatt MR, Amtmann A** (2009) EZ-Rhizo: integrated software for the fast and accurate measurement of root system architecture. *Plant J* 57: 945–956
- Avraham S, Tung CW, Ilic K, Jaiswal P, Kellogg EA, McCouch S, Pujar A, Reiser L, Rhee SY, Sachs MM, et al** (2008) The Plant Ontology Database: a community resource for plant structure and developmental stages controlled vocabulary and annotations. *Nucleic Acids Res (Suppl 1)* 36: D449–D454
- Barthélémy D, Caraglio Y** (2007) Plant architecture: a dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. *Ann Bot (Lond)* 9: 375–407
- Boudon F, Pradal C, Cokelaer T, Prusinkiewicz P, Godin C** (2012) L-py: an L-system simulation framework for modeling plant architecture development based on a dynamic language. *Front Plant Sci* 3: 76
- Bucksch A, Burridge J, York LM, Das A, Nord E, Weitz JS, Lynch JP** (2014) Image-based high-throughput field phenotyping of crop roots. *Plant Physiol* 166: 470–486
- Clark RT, Famoso AN, Zhao K, Shaff JE, Craft EJ, Bustamante CD, McCouch SR, Aneshansley DJ, Kochian LV** (2013) High-throughput two-dimensional root system phenotyping platform facilitates genetic analysis of root growth and development. *Plant Cell Environ* 36: 454–466
- Clark RT, MacCurdy RB, Jung JK, Shaff JE, McCouch SR, Aneshansley DJ, Kochian LV** (2011) Three-dimensional root phenotyping with a novel imaging and software platform. *Plant Physiol* 156: 455–465
- Comas LH, Becker SR, Cruz VM, Byrne PF, Dierig DA** (2013) Root traits contributing to plant productivity under drought. *Front Plant Sci* 4: 442
- Danjon F, Khuder H, Stokes A** (2013) Deep phenotyping of coarse root architecture in *R. pseudoacacia* reveals that tree root system plasticity is confined within its architectural model. *PLoS ONE* 8: e83548
- Danjon F, Reubens B** (2007) Assessing and analyzing 3D architecture of woody root systems, a review of methods and applications in tree and soil stability, resource acquisition and allocation. *Plant Soil* 303: 1–34
- Danjon F, Sinoquet H, Godin C, Colin F, Drexhage M** (1999) Characterisation of structural tree root architecture using 3D digitising and AMApmod software. *Plant Soil* 211: 241–258
- Diener J, Nacry P, Périn C, Dievart A, Draye X, Boudon F, Gaujon A, Godin C** (2013) An automated image-processing pipeline for high-throughput analysis of root architecture in OpenAlea. In R Sievanen, E Nikinmaa, C Godin, A Lintunen, and P Nygren, eds, 7th International Conference on Functional-Structural Plant Models. Saariselka, Finland, pp 85–87
- Draye X, Kim Y, Lobet G, Javaux M** (2010) Model-assisted integration of physiological and environmental constraints affecting the dynamic and spatial patterns of root water uptake from soils. *J Exp Bot* 61: 2145–2155
- Dunbabin VM, Postma JA, Schnepf A, Pagès L, Javaux M, Wu L, Leitner D, Chen YL, Rengel Z, Diggle AJ** (2013) Modelling root-soil interactions using three-dimensional models of root growth, architecture and function. *Plant Soil* 372: 93–124
- Fournier C, Pradal C, Louarn G, Combes D, Soulié JC, Luquet D, Boudon F, Chelle M** (2010) Building modular FSPM under OpenAlea: concepts and applications. In T DeJong, D Da Silva, 6th International Workshop on Functional-Structural Plant Models. Davis, CA, pp 109–112
- French A, Ubeda-Tomás S, Holman TJ, Bennett MJ, Pridmore T** (2009) High-throughput quantification of root growth using a novel image-analysis tool. *Plant Physiol* 150: 1784–1795
- Galkovskiy T, Mileyko Y, Bucksch A, Moore B, Symonova O, Price CA, Topp CN, Iyer-Pascuzzi AS, Zurek PR, Fang S, et al** (2012) GiA Roots: software for the high throughput analysis of plant root system architecture. *BMC Plant Biol* 12: 116
- Garin G, Fournier C, Andrieu B, Houlès V, Robert C, Pradal C** (2014) A modelling framework to simulate foliar fungal epidemics using functional-structural plant models. *Ann Bot (Lond)* 114: 795–812
- Godin C, Caraglio Y** (1998) A multiscale model of plant topological structures. *J Theor Biol* 191: 1–46
- Godin C, Costes E, Sinoquet H** (1999) A method for describing plant architecture which integrates topology and geometry. *Ann Bot (Lond)* 84: 343–357
- Godin C, Costes E, Sinoquet H** (2005). Plant architecture modelling. In CGN Turnbull, ed, *Plant Architecture and Its Manipulation*. Wiley-Blackwell, Oxford, pp 238–288
- Godin C, Sinoquet H** (2005) Functional-structural plant modelling. *New Phytol* 166: 705–708
- Griffon S, de Coligny F** (2014) AMApstudio: an editing and simulation software suite for plants architecture modelling. *Ecol Modell* 290: 3–10
- Gruber BD, Giehl RFH, Friedel S, von Wirén N** (2013) Plasticity of the *Arabidopsis* root system under nutrient deficiencies. *Plant Physiol* 163: 161–179
- Guédon Y, Barthélémy D, Caraglio Y, Costes E** (2001) Pattern analysis in branching and axillary flowering sequences. *J Theor Biol* 212: 481–520
- Hammer G, Dong Z, McLean G, Doherty A, Messina C, Schussler J, Zinselmeier C, Paszkiewicz S, Cooper M** (2009) Can changes in canopy and/or root system architecture explain historical maize yield trends in the U.S. Corn Belt? *Crop Sci* 49: 299–312
- Iyer-Pascuzzi AS, Symonova O, Mileyko Y, Hao Y, Belcher H, Harer J, Weitz JS, Benfey PN** (2010) Imaging and analysis platform for automatic phenotyping and trait ranking of plant root systems. *Plant Physiol* 152: 1148–1157
- Javaux M, Schroeder T, Vanderborcht J, Vereecken H** (2008) Use of a three-dimensional detailed modeling approach for predicting root water uptake. *Vadose Zone J* 7: 1079–1088
- Kellermeier F, Armengaud P, Seditas TJ, Danku J, Salt DE, Amtmann A** (2014) Analysis of the root system architecture of *Arabidopsis* provides a quantitative readout of crosstalk between nutritional signals. *Plant Cell* 26: 1480–1496
- Le Bot J, Serra V, Fabre J, Draye X, Adamowicz S, Pagès L** (2010) DART: a software to analyse root system architecture and development from captured images. *Plant Soil* 326: 261–273
- Leitner D, Felderer B, Vontobel P, Schnepf A** (2014) Recovering root system traits using image analysis exemplified by two-dimensional neutron radiography images of lupine. *Plant Physiol* 164: 24–35
- Leitner D, Klepsch S, Bodner G, Schnepf A** (2010) A dynamic root system growth model based on L-systems. *Plant Soil* 332: 117–192
- Lobet G, Couvreur V, Meunier F, Javaux M, Draye X** (2014) Plant water uptake in drying soils. *Plant Physiol* 164: 1619–1627
- Lobet G, Draye X, Périlleux C** (2013) An online database for plant image analysis software tools. *Plant Methods* 9: 38
- Lobet G, Pagès L, Draye X** (2011) A novel image-analysis toolbox enabling quantitative analysis of root system architecture. *Plant Physiol* 157: 29–39
- Lynch J** (1995) Root architecture and plant productivity. *Plant Physiol* 109: 7–13
- Lynch JP** (2007) Roots of the second green revolution. *Aust J Bot* 55: 493–512
- Mairhofer S, Zappala S, Tracy S, Sturrock C, Bennett MJ, Mooney SJ, Pridmore TP** (2013) Recovering complete plant root system architectures from soil via x-ray μ -computed tomography. *Plant Methods* 9: 8
- Mairhofer S, Zappala S, Tracy SR, Sturrock C, Bennett M, Mooney SJ, Pridmore T** (2012) RooTrak: automated recovery of three-dimensional plant root architecture in soil from x-ray microcomputed tomography images using visual tracking. *Plant Physiol* 158: 561–569
- Mooney SJ, Pridmore TP, Helliwell J, Bennett MJ** (2012) Developing x-ray computed tomography to non-invasively image 3-D root systems architecture in soil. *Plant Soil* 352: 1–22
- Mündermann L, Erasmus Y, Lane B, Coen E, Prusinkiewicz P** (2005) Quantitative modeling of *Arabidopsis* development. *Plant Physiol* 139: 960–968
- Negrón C, Contador L** (2013) Systematic analysis of branching patterns of three almond cultivars with different tree architectures. *J Am Soc Hortic Sci* 138: 407–415
- Pace J, Lee N, Naik HS, Ganapathysubramanian B, Lübberstedt T** (2014) Analysis of maize (*Zea mays* L.) seedling roots with the high-throughput image analysis tool ARIA (Automatic Root Image Analysis). *PLoS ONE* 9: e108255
- Pagès L, Bécel C, Boukcim H, Moreau D, Nguyen C, Voisin AS** (2014) Calibration and evaluation of ArchiSimple, a simple model of root system architecture. *Ecol Modell* 290: 76–84
- Parekh R, Ascoli GA** (2013) Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron* 77: 1017–1038
- Péret B, Li G, Zhao J, Band LR, Voß U, Postaire O, Luu DT, Da Ines O, Casimiro I, Lucas M, et al** (2012) Auxin regulates aquaporin function to facilitate lateral root emergence. *Nat Cell Biol* 14: 991–998
- Plant Ontology Consortium** (2002) The Plant Ontology Consortium and plant ontologies. *Comp Funct Genomics* 3: 137–142

- Postma JA, Lynch JP** (2011) Root cortical aerenchyma enhances the growth of maize on soils with suboptimal availability of nitrogen, phosphorus, and potassium. *Plant Physiol* **156**: 1190–1201
- Pound MP, French AP, Atkinson JA, Wells DM, Bennett MJ, Pridmore T** (2013) RootNav: navigating images of complex root architectures. *Plant Physiol* **162**: 1802–1814
- Pradal C, Dufour-Kowalski S, Boudon F, Fournier C, Godin C** (2008) OpenAlea: a visual programming and component-based software platform for plant modelling. *Funct Plant Biol* **35**: 751–760
- Pradal C, Boudon F, Noguier C, Chopard J, Godin C** (2009) PlantGL: a Python-based geometric library for 3D plant modelling at different scales. *Graph Models* **71**: 1–21
- R Core Team** (2008) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna
- Rustici G, Kolesnikov N, Brandizi M, Burdett T, Dylag M, Emam I, Farne A, Hastings E, Ison J, Keays M, et al** (2013) ArrayExpress update: trends in database growth and links to data analysis tools. *Nucleic Acids Res* **41**: D987–D990