

Sharing phenotypic data: A coding system and a developmental model

Jim Hanan¹, Bruna Bucciarelli^{2,3} and Carroll P. Vance^{2,3}

¹ARC Centre of Excellence for Integrative Legume Research, ARC Centre for Complex Systems, APMC, University of Queensland, Brisbane, Australia 4072; ²U.S. Department of Agriculture, Agricultural Research Service, ³The Department of Agronomy and Plant Genetics, University of Minnesota, 1991 Upper Buford Circle, St. Paul, MN 55108

This work was supported by the USDA, Agricultural Research Service, CRIS no. 3640-21000-024-00D and the Australian Research Council.

Table of contents

- I. Introduction
- II. Materials and Methods: Plant Growth Conditions and Modeling
- III. The Numerical Nomenclature Coding System
- IV. Chronology and Morphological Development of the Main and Axillary Axes for *Medicago truncatula*
- V. Flower Emergence Along the Main and Axillary Shoots
- VI. *Truncatula* model
- VII. Tables, Figures, Appendix A: Listing of the model
- VIII. References

I. Introduction

The *Medicago truncatula* growth analysis and model presented here provides a standardized method to evaluate phenotypic development. In this report we present: (1) a detailed reproducible baseline description of the temporal growth and developmental pattern of *M. truncatula* from cotyledon to early pod formation; (2) a standardized numerical nomenclature coding system that uses easily identifiable developmental growth stages to define plant growth; and (3) a developmental structural model of *M. truncatula* that allows phenological and geometric data, from different experiments, to be visualized for ease of comparison.

The detailed description of vegetative growth and reproductive organ emergence makes it possible to monitor morphological growth and development because it records plant structures that persist and are visible to the unaided eye. Previous characterizations of *M. truncatula* focused on floral and pod traits (Lesins and Lesins, 1979; Benlloch et al., 2003; Wang and Grusak, 2005). Recently, Moreau et al. (2006) evaluated genotypic variability in *M. truncatula* as a function of thermal time. Their analysis identified changes in *M. truncatula* leaf initiation and appearance on axillary shoots. By contrast, our study details the chronology and sequential appearance of shoot and flower development over the initial 40 day period of growth. Such a detailed account of growth provides an architectural model whereby the development of the whole plant can be assessed.

The development of a numerical nomenclature coding system provides a means to define specific growth stages in plant development thereby making it possible to discern and document specifically where and when changes occur for phenotypically distinct plants. A nomenclature coding system may be used to clearly communicate developmental stages without extensive descriptions and provide a basis to standardize tissue collection for analysis. It may also be incorporated into a component of the *M. truncatula* plant ontology database, specifically the plant morphological aspect of the database (Blake, 2004). Its numerical components are conducive to being queried by computational approaches. The numerical nomenclature coding system presented here is based on the system outlined by Mundermann et al. (2005) for *Arabidopsis*, by which changes in morphological and temporal development can be easily monitored. This coding system documents plant morphological development based on a series of defined growth units. We have defined plant growth according to metamer production along the main and axillary axes. These growth units represent incremental steps in the progression of whole plant development. We have documented timing of metamer production over the initial 40 day growth period thereby allowing assessment of alterations in developmental timing. We have included a component in the system that divides metamer growth into nine sub-stages based on leaf or reproductive organ development (Table I).

An empirical model of *M. truncatula* development was developed using L-systems, a formalism for describing plant development (Lindenmayer 1968). The L-system model provides a basis for modeling of genetic and/or physiological mechanisms underlying *M. truncatula* growth and development. The model code allows a user to enter their own data for plant parameters, thus creating a visual comparison with the model presented here. It is hoped that the research community will contribute models of different mutants and varieties under various conditions to a Model Repository <http://www.cpai.uq.edu.au/MtHandbook/> associated with the *M. truncatula* Handbook, allowing comparison between different phenotypes.

These findings are meant to create a framework for the future analysis of phenotypic alterations of *M. truncatula* due to either genetic mutations or environmental conditions. They have been published as one of the elements in Bucciarelli et al. (2006).

II. Materials and methods

Medicago truncatula seeds of line A17 of cv. Jemalong were chemically scarified with concentrated sulfuric acid for 8 minutes and surface sterilized for 3 min with commercial grade bleach (5.25% sodium hypochlorite). Seeds were given a 2 day germination period at 4° C then transferred to petri plates with moistened filter paper and given a 14 d vernalization period at 4° C. Vernalized seeds with a radical length of 1-1.5 cm were planted in pots containing quartz sand. Pot size was 10cm x 10cm x 35cm. Plants were grown at 24-26° C and a 16h photoperiod (<http://www.isv.cnrs-gif.fr/embo01/manuels/pdf/module1.pdf>) with a light intensity at the sand surface of 430 – 490 mol· m⁻² sec⁻¹. Plants were fertilized 3x per week (150 ml of nutrient solution per fertilizer application) with the following nutrient solution: final concentration: KNO₃, 15 mM; Ca(NO₃)₂ 4H₂O, 12.5 mM; Ca(H₂PO₄)₂, 1mM; MgSO₄ 7H₂O, 1mM; Fe EDTA, 0.01mM; MnCl₂, 0.004 mM; H₃BO₃, 0.02 mM; ZnSO₄ 7H₂O, 0.0004 mM; NaMoO₄, 0.0001 mM; CaSO₄ 5H₂O, 0.0001 mM. Data was collected at 4, 7, 11, 14, 18, 21, 25, 28, 32, and 40 days after planting (dap). Five seedlings were selected for continuous monitoring of shoot growth. These five plants were photographed and various parameters were measured throughout the 40 day period. Plant growth parameters measured included shoot and root fresh and dry weights, shoot and root total lengths, leaf size, and internode lengths. An additional set of plants was grown in parallel for collection and processing of tissue. Three replicated experiments were performed. Images were taken with a Sony DSC-D770 digital camera. Growth angles were calculated from the digital images. All other parameters were measured using a handheld ruler.

The model of *M. truncatula* development was developed using the L-system-based L-studio software from the University of Calgary (Prusinkiewicz et al., 2000a), which can support a wide range of empirical and mechanistic modeling approaches (Prusinkiewicz, 2004; Mundermann et al., 2005). In an L-system, plant components are represented by an alphabet of symbols with associated parameters, arranged in a string to represent plant structure, with branching topology imposed by a hierarchy of brackets. Specific symbols encode local properties, such as component branching angles, widths and colours. Daily growth and developmental changes are captured by applying production rules to all symbols in the current string to produce a new string representing the plant on the next day. The structure can be visualized in schematic (Figure 1) or realistic form (Figure 4, D-E). The rules for production of a new metamer (comprised of an internode, a leaf and an axillary meristem) by an apical meristem, are applied after passage of simulated time equivalent to the plastochron or to the branching delay measured for the meristem at that position in the plant. Maximum lengths and duration of growth for the individual components are drawn from the empirical data according to the nodal position along the axis, then scaled by the ratio of their age to the duration of growth.

III. The Numerical Nomenclature Coding System

Based on the coding system developed for the quantitative modeling of *Arabidopsis* by Mundermann et al. (2005) we propose a numerical nomenclature coding system for *M. truncatula* that follows three principal phases of plant growth from cotyledon to early pod formation; (1) vegetative growth along the main shoot; (2) vegetative growth along the axillary shoots; and (3) the emergence and development of reproductive organs. These particular growth phases were chosen because they divide growth of *M. truncatula* into easily identifiable components and they provide distinct stages of growth for data collection. The proposed nomenclature system documents plant morphological development by following a series of defined growth units representing incremental steps in the progression of whole plant development. It numbers the

developing metameric units sequentially along the main and axillary axes of growth and designates position of flower emergence.

Medicago truncatula produces a procumbent, trailing type of growth habit from the combined elongation of the main and axillary shoots. As shoot apices develop, they produce a series of growth units defined as metamers (m) that consist of an internode, leaf and axillary bud. The main shoot, the primary axis of vegetative growth, initially produces 4 metamers separated by very short immeasurable internodes. Afterward, the internodes of the successively developing metamers elongate producing additional trifoliolate leaves with measurable internodal distances. A spiral type of phyllotaxy is produced in the elongating portion of the shoot. The axillary bud of each metamer has the potential to grow into an axillary shoot. The initiation and growth of these axillary shoots is coordinated with the elongation of the main shoot.

The nomenclature coding system starts with the metamer associated with the unifoliolate leaf as metamer 1 (m1). The units developing above this are numbered in ascending order with m2 associated with the first trifoliolate, etc. The metamers forming axillary shoots developing off of the main axis are coded first according to their main shoot metamer of origin then numbered sequentially starting at the base with 1 (e.g. The axillary shoot associated with m1 that contains 1 unit of growth is designated as m1-1). Floral location is indicated with an “F” preceding the number of its associated metamer of emergence. Flowers emerging from the axils of the main shoot are designated according to the metamer of origin followed by F0 (e.g., floral emergence from metamer 6 is designated as m6-F0). When considering a flower developing along an axillary shoot it is coded according to the main shoot metamer of origin followed by the axillary shoot metamer of emergence preceded by “F” (e.g. The flower emerging from the axillary shoot associated with m1 and from the axil of metamer 3 of the m1 axillary shoot is coded as m1-F3). The numerical coding system identifying units of *M. truncatula* growth and their location is illustrated in Figure 1. Additionally, the nomenclature coding system defines leaf and reproductive organ development into nine sub-stages (Table I) depicted by a decimal added as an extension to the numerical code. These sub-stages were chosen because they are visually identifiable and provide a continuum of growth and development for the organ that they specify. Therefore, to indicate that the m1-1 leaf is fully developed a code of m1-1.9 is used. Various parameters can be measured that may distinguish growth alterations missed by the coding system alone. Regardless of which parameters are monitored, the examination of multiple parameters enhances the potential to detect true phenotypic differences resulting from mutations versus phenotypic plasticity (Coleman et al., 1994; Boyes et al., 2001).

An animation of an empirical model of *M. truncatula* growth and development is presented in Figures 2 and 3. Because our initial study compared growth of *M. truncatula* under nutrient sufficient, phosphorus- and nitrogen stress, Figure 2 and 3 show growth differences of *M. truncatula* under these nutrient conditions over a 40 day period.

IV. Chronology and Morphological Development of the Main and Axillary

Axes for *M. truncatula*

A simple way to orient oneself to the developing plant is to use the unifoliolate leaf as a reference marker. The unifoliolate leaf, associated with metamer one (m1), emerges from the developing apical meristem between the two cotyledons (Fig. 4A). The second leaf (associated with m2) emerges 180 degrees opposite to the unifoliolate. The developing leaves from m1 and m2, plus the cotyledons, produce a cross type of appearance to the developing plant (Fig. 4A). The third leaf (associated with m3) develops 140 degrees relative to the m2 leaf, adjacent to the m1

leaf. The fourth leaf (associated with m4) develops on the same side of the cotyledons as the m2 leaf and approximately 165 degrees from the m3 leaf (Fig. 4B).

Figure 5 depicts a time line for the chronological progression of plant growth throughout the stages of development over a 40 day period. It includes the time interval to reach key developmental stages and the correlation of these stages with the onset of axillary shoot development and floral emergence. An animation of shoot temporal development showing the sequential appearance of shoot structures in a color-coded manner can be viewed in Figure 6.

As the seedling develops, two cotyledons are produced and are fully expanded by 4 days-after-planting (dap). The cotyledons have a succulent characteristic and are blue-green in color. By 7 dap metamer 1 (m1) starts to develop as evidenced by the presence of the unifoliate leaf at a mean developmental stage of m1.4, indicating that the unifoliate leaf blade is folded and the petiole is visible (Table I).

By 14 dap m1 is at full development (m1.9) and m2 is at a mean developmental stage of m2.7, indicating that the leaflets of m2 are almost fully open (Fig. 4A, Table I). By 21 dap the leaf of m3 is fully developed (m3.9) and m4 is at a mean developmental stage of m4.6, indicating that the leaflets of m4 are greater than half open. The leaves associated with metamer 5 and 6 were fully developed by 28 and 32 dap, respectively (Fig 4C). The main shoot developed subsequent metamers between 32 and 40 dap. And by 40 dap, a total of 10-11 metamers were produced along the main shoot. The cotyledons showed no change in appearance throughout this 40 day period. The cotyledons maintained their succulent characteristic and blue-green color.

The first axillary bud to break and initiate growth was associated with m1. The m1 axillary bud break was initiated between 18 and 21 dap and was correlated with the development of m4 along the main shoot (Fig. 4B). By 40 dap the axillary shoot of m1 was at a mean developmental stage of m1-6.4, indicating that 6 metamers were produced on the m1 axillary shoot and the youngest metamer (6) was at a developmental stage of 0.4.

The second axillary bud to break and initiate growth was associated with m2. The m2 axillary bud break was initiated between 21 and 25 dap and was correlated with the development of m5 along the main shoot. By 40 dap the axillary shoot of m2 had a mean developmental stage of m2-6.3 indicating that 6 metamers were produced on the m2 axillary shoot and the youngest metamer (6) was at a developmental stage of 0.3.

The third axillary bud to break and initiate growth was associated with m3. The m3 axillary bud break was initiated between 25 and 28 dap and was correlated with the development of m6 along the main shoot. By 40 dap the m3 axillary shoot was at a mean developmental stage of m3-5.4, indicating that 5 metamers were produced along the m3 axillary shoot and the youngest metamer (5) was at a developmental stage of 0.4.

Subsequently, the m4 axillary shoot developed between 28 and 32 dap and was correlated with the development of m8 along the main shoot. By 40 dap the m4 axillary shoot was at a developmental stage of m4-4.4, indicating that 4 metamers were produced along the m4 axillary shoot and the youngest metamer (4) was at a developmental stage of 0.4.

Axillary bud break from m5 was initiated by 32 dap and was correlated with the development of m9 along the main shoot. By 40 dap the m5 axillary shoot was at the m5-3.4 developmental stage. Axillary buds from m6 and m7 initiated growth just prior to 40 dap and by 40 dap the m6 and m7 axillary shoots contained 1 leaf. It should be noted that axillary shoot elongation began to occur when the third leaf emerged from its growing point.

V. Flower Emergence Along the Main and Axillary Shoots

Flowers began to emerge along the main shoot by 32 dap. The initial location of flower emergence was from the axils of m5, m6 and m7. As the main shoot continued to elongate, flowers emerged from axils above but not below these metamers.

On elongated axillary shoots, flowers were also visible at 32 dap. Flowers emerged from elongated axillary shoots that originated from m1, m2, m3, and m4. The location of flower emergence along the axillary shoots occurred at a specific metamer from the main shoot metamer of origin and was different for the m1 axillary relative to the other axillary shoots. The first flower to have emerged for the m1 axillary was from the m1-3 position. Whereas, the first flower to have emerged from the other axillary shoots is from the m2-2, m3-2 and m4-2 position.

VI. Truncatula model

This is a modified version of the model presented in the paper “A standardized method for analysis of *Medicago truncatula* phenotypic development” by Bruna Bucciarelli, Jim Hanan, Debra Palmquist, and Carroll P. Vance (Bucciarelli et al., 2006). The original model can be found in the supplemental material of that paper. The version described here can be downloaded from the *M. truncatula* Handbook site <http://www.cpai.uq.edu.au/MtHandbook/>. The modifications allow you to record your own phenotype by entering data in the appropriate places and then to run for visual comparison to the Truncatula model.

The model is written in the L-system-based plant modeling language cpfg [Prusinkiewicz et al., 2000b], which runs in the L-studio modeling environment under Windows [Prusinkiewicz et al., 2000c]. A trial version of L-studio can be downloaded from http://www.algorithmicbotany.org/virtual_laboratory. For details of software installation, see the L-studio/vlab documentation. Details of L-system syntax can be found in the cpfg manual in the L-studio package.

The downloaded model is in the form of a zipped archive. Extracting the contents will create a folder containing the following files:

- description.txt - a brief description of user interaction with the model
- plant.l – the cpfg code of the model
- plant.v , plant.a – viewing and animation parameters
- materials.mat – parameters describing colors and materials
- Parameters.pnl – text file describing the model’s control panel
- cotyledon.s, leaf.l.s, leaf.r.s, eleaf.l.s, eleaf.r.s – text files describing surfaces representing cotyledons and left and right hand sides of lateral and terminal leaflets
- LSspecifications – file characterizing the model’s structure as needed by the L-studio software

The plant.l file describes the L-system model, including data that can be edited to create a particular phenotype (see a line-numbered copy in Appendix A). The surface files describe Bezier surface models that are handcrafted using the L-studio surface editor to match observed shapes.

The data that can be modified to create the model are of two kinds: (a) global constants and position-independent data, and (b) plant-type/position dependent data. All sizes are expressed in millimeters, angles in degrees, and times in days.

Position-independent data are specified at the beginning of the text file lsystem.l lines 1-37 in #define statements, as well as within the appropriate productions. All stem and petiole widths, petiolule lengths, phyllotactic angles, divergence angles within trifoliate leaves, and the rate of change of angles are assumed to be position-independent and are estimated from a small sample (data not shown).

Plant-type/position dependent architectural data are specified in initializations of data arrays in the Define: section in the file plant.l lines 39-114. Values that are preceded by a /* Your Model */ comment can be changed to reflect your own data. For nodes along main stem and first order axes these data include:

- The plastochron for the apex at that position (plastochron);
- The flowering time for flower buds at that position (flowertime) where a value of 0 means there is no flowering at that node, and 99 means that the data is not available;
- Delay before development of branches at each main stem node (branchdelay) where 99 indicates no branch developed during the course of the experiment;
- Maximum internode length at that position (internodeLength);
- Duration of expansion of internodes (internodeDuration);
- Maximum petiole length at that position (petioleLength);
- Duration of expansion of petiole (petioleDuration);
- Maximum leaf length at that position (leafLength);
- Duration of expansion of leaf (leafDuration);

Simulation control

Setting the value of STEPS on line 3 will determine how many days the model is run. The value of BRANCHING on line 4 to 1 causes the model to include branches, while 0 causes the model to exclude branches. Plant type is defined by setting PT on line 6 to 0 for the Truncatula model, 1 for your model and 2 for a comparison of both. The age of the plant is represented by the global variable time, which has the initial value of 0 days (line 119) and is incremented during the simulation by 1 day (line 120) in each step. The simulation runs for the derivation length (number of STEPS) as specified on line 116. These parameters can be manipulated using the Parameters panel available through the L-studio Panel interface.

The starting string or axiom is defined in line 123. The module X with parameter PT determines the visualization to be drawn through Decomposition rules in lines 216-218, which layout the appropriate surface representing ground level, a legend, and plant initial apex A with the appropriate plant type parameter.

Production rules

The core of the model consists of production rules that specify the behavior of different model components (modules) in lines 125-272. Visualisation rules in lines 273-315 then detail how the components should be drawn on the screen. The operation of the model can be described in terms of the operation of these modules.

Module A(pt,order,parentnode,node,delay) (lines 125-185) represents apical meristems on the main stem and axillary branches, which create the sequence of metamers that constitute these axes. The parameters have the following meanings:

- pt: plant type for this plant (0 for Truncatula or 1 for “your model”)
- order: branching order of the axis (0 for main stem, 1 for the first-order lateral axes, 2 for second-order lateral axes, etc)
- parentnode: node number on the parent axis where this axis arose
- node: node number of the metamer which will next be created by the apex

- delay: the plastochron, the number of days before the next metamer will be produced by this apex.

While the apex delay is greater than one, the delay is reduced by 1 day in each step, according to the rule in line 131. Once the delay value reaches 1, the A module produces metamers. All metamers have an internode F with parameters current length, growth duration, and maximum size; for order 1 branches there is also a growth delay. The phyllotactic angle is set according to position, if your model has a different phyllotactic angle, you can put a conditional statement if (pt=ptYourModel) {phyllo="your angle";} after the phyllo assignment statement in each of the following productions. The metamer at the first position on the main stem is composed of 2 cotyledons with axillary buds B (lines 134-143) with the phyllotactic angle to the next leaf set to 90 degrees. The metamer at the second position produces a unifoliate leaf with an axillary bud B (lines 145-163) with a phyllotactic angle to the next leaf of 140 degrees. Metamers at all other positions (lines 165-185) produce trifoliate leaves with axillary bud B, axillary reproductive bud b, and with phyllotactic angle to the next leaf of 165 degrees. For each type of metamer, internode length, growth duration, branch delay, and plastochron are set by accessing values for the appropriate data arrays according to the current plant type, parent node, and the node being produced.

Module B(pt,order,parentnode,delay) (lines 187-189) represents dormant vegetative axillary meristems. Once the delay before branching has expired, the meristem will be transformed into an active meristem A by the production in line 188. Line 189 controls the reduction of delay. Note that axes above first order are not modelled. The parameters have the following meanings:

- pt: plant type for this plant (0 for Truncatula or 1 for “your model”)
- order: branching order of the axis (1 for the first-order lateral axes, 2 for second order lateral axes, etc)
- parentnode: node number on the parent axis where this axis arose
- delay: the number of days before this apex will start to produce a branch.

Module b(pt,order,parentnode) (lines 191-192) represents a reproductive bud. The parameters have the following meanings:

- pt: plant type for this plant (0 for Truncatula or 1 for “your model”)
- order: branching order of the axis (1 for the first-order lateral axes, 2 for second order lateral axes, etc)
- parentnode: node number on the parent axis where this axis arose.

The bud will produce a reproductive structure R once the specified flowering time is reached.

Growth of internodes, petioles and leaves are controlled by productions in lines 194-202. After an initial delay (lines 195-196) stem segments grow in a linear fashion, based on a declining duration and increasing length. This is an abstraction from the more likely sigmoidal pattern of growth and different growth functions could be specified here. Widths, encoded with the ! (line 203), and angles (lines 205-208) encoded with & for pitch down, + for right turn and – for left turn, grow at an arbitrary exponential rate (assigned in the productions initiating them) until they reach their maximum.

Colour changes for your model can be specified in the production in line 211.

Module L(*pt,type,node*) represents the leaves of different types. The parameters have the following meanings:

- *pt*: plant type for this plant (0 for *Truncatula* or 1 for “your model”)
- *type*: leaf type determining leaflet shape and arrangement (1 for the cotyledons: lines 221-229, 2 for unifoliate: lines 230-243, 3 for trifoliate: lines 244-267)
- *node*: node number on the parent axis where this leaf appears.

Each production specifies the appropriate placement of leaflet shapes, specified with ~ modules, according to the leaf type, with a petiole *P* and petiolules specified with *F* modules. Leaf angle parameters are adjusted by plant type in the case of trifoliate leaves. Leaflet and petiole lengths and growth durations are assigned according to the plant type and nodal location of the leaf. The @*v* module aligns the leaf horizontally.

Module P(*segs,start,duration,max*) (lines 269-271) represents a petiole. Petioles are broken down into the number of segments *segs*, each represented by an *F* module so that tropism effects can be used to give them a naturally bending appearance. The degree of bending is set by the *L* productions using the _ (underscore) module, with the value chosen by visual inspection.

Visualisation rules (lines 273-315) do not change the string representing the plant from step to step, but are used during interpretation of the specified modules to create the image. In lines 275-280, the surfaces are oriented according to their age so that the leaflets will open over time. The visualization productions in lines 282-285 produces labels associated with selected apices.

Module R(*pt,order,node,birthday*) (lines 287-303) represents a reproductive structure. The parameters have the following meanings:

- *pt*: plant type for this plant (0 for *Truncatula* or 1 for “your model”)
- *order*: branching order of the axis (1 for the first-order lateral axes, 2 for second order lateral axes, etc)
- *node*: node number on the parent axis where this axis arose
- *birthday*: day of first appearance of the reproductive structure.

The stage of growth is calculated from the birthday, then the appropriate visualization size, colour and shape are chosen. If your model should have a different visualisation, duplicating these lines, and conditioning them on the value of *pt* will allow you to create your own visualisation.

Module SURFACE (lines 305-309) draws a red-brown rectangle representing the surface of the ground.

Module Legend (lines 311-315) positions and prints the legend appropriate for the plant type, along with the current day since the beginning of the simulation.

VII. Table, Figures Appendix A: Listing the model

Table I. Decimal component of the numerical nomenclature coding system of <i>Medicago truncatula</i> defining leaf and reproductive organ development into nine sub-stages of growth.			
Decimal Code ^a	Leaf Developmental Stages ^b	Decimal Code ^a	Reproductive Organ Developmental Stages ^b
0.1	Bud break	0.1	Bud stage
0.2	Blade difficult to discern, visible	0.2	Petals visible, green
0.3	Blade discernible, small	0.3	Petals visible, yellow (closed)
0.4	Blade folded, petiole visible	0.4	Petals open
0.5	Blade half open	0.5	Petals senescent
0.6	Blade greater than half open	0.6	Pod visible
0.7	Blade almost fully open	0.7	Pod small (1-5 mm long)
0.8	Blade fully open, green	0.8	Pod medium (> 5 mm long), green
0.9	Blade fully open, blue-green	0.9	Pod full size, brown
^a Decimal numerical code for organ development from emergence to full development.			
^b Definition of decimal code for stages of organ development.			

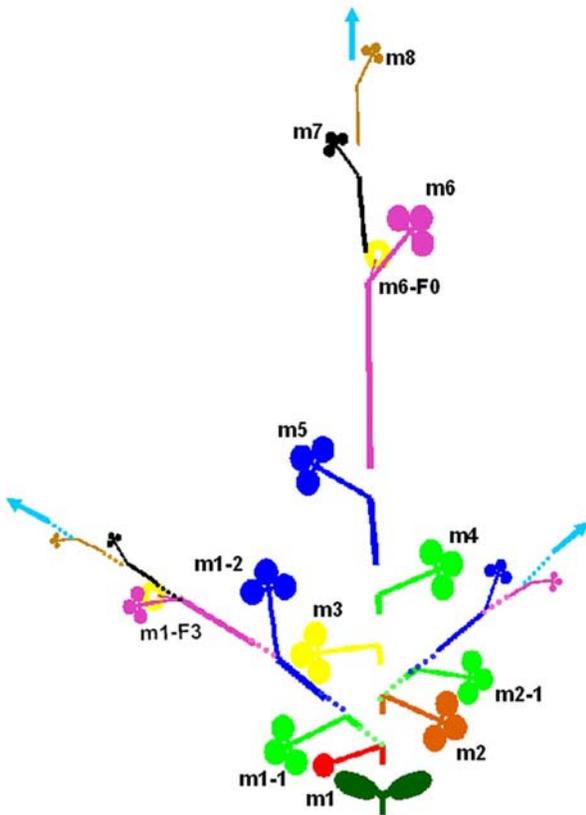


Figure 1: Schematic diagram of *Medicago truncatula* illustrating the numerical nomenclature coding system. Metamers are labeled along the main shoot and some are labeled along the axillary shoots. The position of flower emergence is designated with an “F” as part of the coding system. Structures sharing a common color (i.e. pink) appear simultaneously.

Figures 2 and 3 can be found at <http://www.cpai.uq.edu.au/MtHandbook/>

Figure 2 and 3: Animation of the empirical model of *Medicago truncatula* growth under nutrient sufficient (N+P+), phosphorus-deficient (N+P-) and nitrogen-deficient (N-P+) conditions over 40 days. (Fig. 2) side view, (Fig. 3) overhead view. The model has been developed using the L-system-based L-studio software. Labeling of metamer appearance is shown for the main shoot and the first two axillary shoots. Flower appearance is depicted by yellow spheres, and pods are depicted by green crescents.

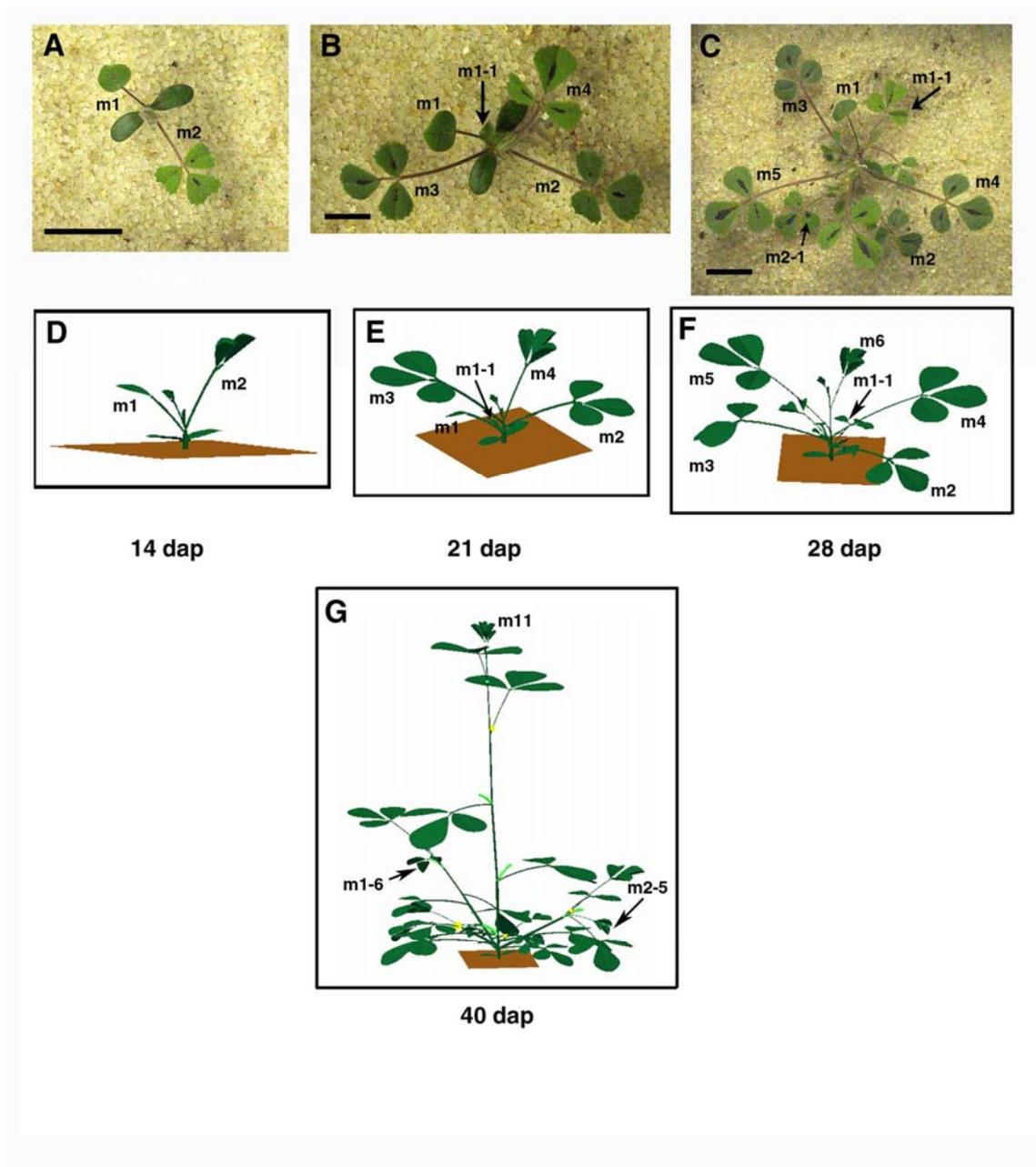


Figure 4: Shoot development of *Medicago truncatula* at 14 (A), 21 (B) and 28 (C) days after planting (dap). D, the model rendition of (A). E, the model rendition of (B). F, the model rendition of (C). G, the model rendition of plant growth at 40 dap. Leaves associated with metamer growth-units are labeled using the numerical nomenclature coding system illustrated in Figure 1. Flowers are depicted by yellow spheres and pods are depicted by green crescents. Scale bar = 1cm.

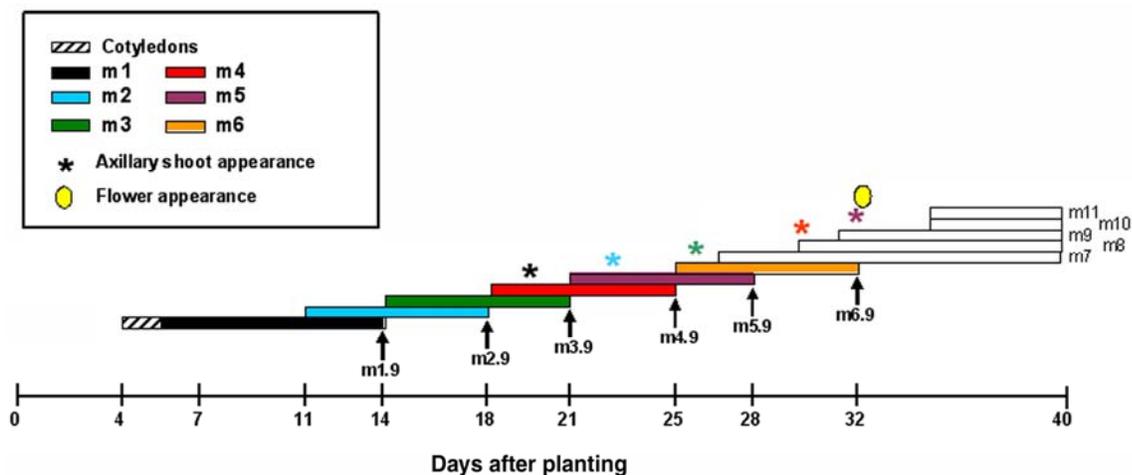


Figure 5: Chronology of metamer appearance along the main shoot of *Medicago truncatula* over a 40 day period. The first six metamers are color-coded as shown in the legend. Time of axillary shoot appearance is depicted by an asterisk (*), color-coded according to its metamer of origin. Time of flower appearance is indicated by a yellow oval. The nomenclature code of most metamers, at completion of growth, is labeled and indicated with an arrow.

Figure 6: Animation Schematic.avi

Figure 6: A schematic visualization of *Medicago truncatula* shoot temporal development showing the sequential appearance of shoot structures in a color-coded manner. The animation represents plant growth under nutrient sufficient conditions over a 40 day period.


```

69. /* m1 */ 99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,
70. /* m2 */ 99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,
71. /* m3 */ 99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,
72. /* m4 */ 99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,
73. /* m5 */ 99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99);
74.
75. array branchdelay[ptAll][15] = {
76. /* Truncatula */ 99,13,12,12,12,11,99,99,99,99,99,99,99,99,99,
77. /* Your Model */ 99,99,99,99,99,99,99,99,99,99,99,99,99,99,99);
78.
79. array internodeLength[ptAll][6][15] = {
80. /* Truncatula */
81. /* ms */ 1,1,1,1,1,2.22,15.5,22.25,20,20,20,20,20,20,20,20,20,
82. /* m1 */ 1.89,5.67,16.67,12.67,13,13,13,13,13,13,13,13,13,13,13,
83. /* m */ 4.67,16.44,25.78,11.33,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,
84. /* m3 */ 5.11,19.33,14.22,3.11,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,
85. /* m4 */ 3.89,10.0,2.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,
86. /* m5 */ .1,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,.1,
87. /* Your Model */
88. /* ms */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
89. /* m1 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
90. /* m2 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
91. /* m3 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
92. /* m4 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
93. /* m5 */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
94.
95. array internodeDuration[ptAll][15] = {
96. /* Truncatula */ 6,8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
97. /* Your Model */ 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1);
98.
99. array petioleLength[ptAll][15] = {
100. /* Truncatula */ 0,8,7,14,3,19,2,20,4,22,1,17,6,17,17,17,17,17,17,
101. /* Your Model */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
102.
103. array petioleDuration[ptAll][15] = {
104. /* Truncatula */ 6,8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
105. /* Your Model */ 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1);
106.
107. array leafLength[ptAll][15] = {
108. /* Truncatula */ 5,4,4,8,4,9,8,11,8,12,9,15,2,15,15,15,15,15,15,
109. /* Your Model */ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
110.
111. array leafDuration[ptAll][15] = {
112. /* Truncatula */ 6,8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
113. /* Your Model */ 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1);
114. }
115.
116. derivation length: STEPS
117.
118. /* keep track of simulation time: 1 step = 1 day */
119. Start: {time=0;}
120. StartEach: {time=time+1;}
121.
122. /* Axiom uses X decomposition to select between particular plant type, or to display all */
123. Axiom: X(PT)
124.
125. /* Main stem apex A goes through 3 phases producing cotyledons, unifoliate leaf
126. and then trifoliates. This is dependant on its current nodal position on the branch as
127. recorded in the parameter node, with delay controlled by final parameter d
128. Parent node pn (0 for main stem) is used to access appropriate parameters */
129.
130. /* handle apex plastochron delay */
131. A(pt,order,pn,node,d) : d>1 --> A(pt,order,pn,node,d-1)
132.
133. /* producing cotyledons and axillary buds */
134. A(pt,order,pn,node,d) : pn==0 && node<A_UNI && d==1
135. {phyllo=90;
136. internodemax=internodeLength[pt][pn][node];
137. internodeduration=internodeDuration[pt][node];
138. internodestart=internodemax/internodeduration;

```

```

139. } -->
140. ,(32+pt,pt)!(A_WIDTHS,A_WIDTHR,A_WIDTHM)F(internodestart,internodeduration,internodemax)
141. [^(COT_Angle)L(pt,1,pn,0)]&[(COT_Angle/2)B(pt,order+1,node,branchdelay[pt][node])]/(180)
142. [^(COT_Angle)L(pt,1,pn,0)]&[(COT_Angle/2)B(pt,order+1,node,branchdelay[pt][node])]
143. /(phylo)^(0)!(A_WIDTHS,A_WIDTHR,A_WIDTHM)A(pt,order,pn,node+1,plastochron[pt][pn][node])
144.
145. /* producing Unifoliate leaf */
146. A(pt,order,pn,node,d) : pn==0 && node==A_UNI && d==1
147. {phylo=140;
148. internodemax=internodeLength[pt][pn][node];
149. internodeduration=internodeDuration[pt][node];
150. internodestart=internodemax/internodeduration;
151. /* determine angle change according to nutrient status */
152. if (pt==ptTruncatula) {
153.   anglemax=LEAF_AngleMax;
154.   anglerate=LEAF_AngleRate;
155. } else { /* Your Model */
156.   anglemax=LEAF_AngleMax;
157.   anglerate=LEAF_AngleRate;
158. }
159. } -->
160. F(internodestart,internodeduration,internodemax)
161. [&(LEAF_AngleStart,anglerate,anglemax)L(pt,2,pn,node)]
162. [_(.05)&(anglemax*4/5)/(90)B(pt,order+1,node,branchdelay[pt][node])]
163. /(phylo)^(0)!(A_WIDTHS,A_WIDTHR,A_WIDTHM)A(pt,order,pn,node+1,plastochron[pt][pn][node])
164.
165. /* producing Trifoliate leaves */
166. A(pt,order,pn,node,d) : d==1
167. {phylo=165;
168. internodemax=internodeLength[pt][pn][node];
169. internodeduration=internodeDuration[pt][node];
170. internodestart=internodemax/internodeduration;
171. /* determine angle change according to nutrient status */
172. if (pt==ptTruncatula) {
173.   anglemax=LEAF_AngleMax;
174.   anglerate=LEAF_AngleRate;
175. } else { /* Your Model */
176.   anglemax=LEAF_AngleMax;
177.   anglerate=LEAF_AngleRate;
178. }
179. } -->
180. /* internode growth delay for order 1 branches less than node 3: order*9*(node<=3) */
181. F(internodestart,internodeduration,internodemax,order*9*(node<=3))
182. [&(LEAF_AngleStart,anglerate,anglemax)L(pt,3,pn,node)]
183. [_(.02)&(anglemax*4/5)/(90)B(pt,order+1,node,branchdelay[pt][node])]
184. [&(anglemax/2)b(pt,pn,node)]
185. /(phylo)&(5)!(A_WIDTHS,A_WIDTHR,A_WIDTHM)A(pt,order,pn,node+1,plastochron[pt][pn][node])
186.
187. /* no branching at cotyledonary node or at higher than order 1 or if branching turned off */
188. B(pt,order,node,d) : BRANCHING && node>0 && order<2 && d==1 --> A(pt,order,node,1,1)
189. B(pt,order,node,d) : d>1 --> B(pt,order,node,d-1)
190.
191. /* flower bud initiation */
192. b(pt,pn,node) : time==flowertime[pt][pn][node] --> R(pt,pn,node,time)
193.
194. /* internode and petiole length and width, lamina growth */
195. F(l,d,m,delay) : delay<=1 --> F(l,d,m)
196. F(l,d,m,delay) : delay>1 --> F(l,d,m,delay-1)
197. F(l,d,m) : l<m {l=l+(m-l)/d; if (l>m) {l=m;}} --> F(l,d-1,m)
198. ~c(pt,l,d,m) : l<m {l=l+(m-l)/d; if (l>m) {l=m;}} --> ~c(pt,l,d-1,m)
199. ~l(pt,l,d,m) : l<m {l=l+(m-l)/d; if (l>m) {l=m;}} --> ~l(pt,l,d-1,m)
200. ~r(pt,l,d,m) : l<m {l=l+(m-l)/d; if (l>m) {l=m;}} --> ~r(pt,l,d-1,m)
201. ~L(pt,l,d,m) : l<m {l=l+(m-l)/d; if (l>m) {l=m;}} --> ~L(pt,l,d-1,m)
202. ~R(pt,l,d,m) : l<m {l=l+(m-l)/d; if (l>m) {l=m;}} --> ~R(pt,l,d-1,m)
203. !(w,r,m) : w<m {w=w*r; if (w>m) {w=m;}} --> !(w,r,m)
204.
205. /* angle changes over time */
206. &(a,r,m) : a<m {a=a*r; if (a>m) {a=m;}} --> &(a,r,m)
207. +(ang,rate,max) : ang<max {ang=ang*rate; if (ang>max) {ang=max;}} --> +(ang,rate,max)
208. -(ang,rate,max) : ang<max {ang=ang*rate; if (ang>max) {ang=max;}} --> -(ang,rate,max)

```

```

209.
210. /* colour change for your model at specified time points could be invoked here */
211. ,(c,pt) : pt==ptYourModel && (time==99) --> ,(c,pt)
212.
213. decomposition
214. maximum depth: 20
215. /* set up desired model for display */
216. X(pt) : pt==ptAll --> [x(ptTruncatula)]-(90)f(100)+(90)[x(ptYourModel)]
217. X(pt) : pt!=ptAll --> x(pt)
218. x(pt) --> [SURFACE Legend(pt) \ (15) A(pt,0,0,0,1)]
219.
220. /* leaf development */
221. /* type 1 is used for cotyledons */
222. L(pt,t,pn,node) : t==1
223. {
224.   leafmax=leafLength[pt][node];
225.   leafduration=leafDuration[pt][node];
226.   leafstart=leafmax/leafduration;
227. } -->
228. I(PET_WIDTHS,PET_WIDTHHR,PET_WIDTHM)
229. F(PStart,PDuration,PMax)-c(pt,leafstart,leafduration,leafmax)
230. /* type 2 is used for Unifoliate leaves */
231. L(pt,t,pn,node) : t==2
232. {
233.   segs=5;
234.   leafmax=leafLength[pt][node];
235.   leafduration=leafDuration[pt][node];
236.   leafstart=2*leafmax/leafduration;
237.   petmax=petioleLength[pt][node];
238.   petduration=petioleDuration[pt][node];
239.   petstart=0;
240. } -->
241. I(PET_WIDTHS,PET_WIDTHHR,PET_WIDTHM)
242. _(.07)P(segs,petstart/segs,petduration,petmax/segs)_ (0)
243. &(UNI_Angle)[-l(pt,leafstart,leafduration,leafmax)][~r(pt,leafstart,leafduration,leafmax)]
244. /* type 3 is used for Trifoliate leaves */
245. L(pt,t,pn,node) : t==3
246. {
247.   segs=5;
248.   leafmax=leafLength[pt][node];
249.   leafduration=leafDuration[pt][node];
250.   leafstart=leafmax/leafduration;
251.   petmax=petioleLength[pt][node];
252.   petduration=petioleDuration[pt][node];
253.   petstart=petmax/petduration;
254.   /* adjust leaflet angle for N- */
255.   angle1=TRI_Angle1;
256.   if (pt==ptYourModel) {
257.     angle1=TRI_Angle1;
258.   }
259. } -->
260. I(PET_WIDTHS,PET_WIDTHHR,PET_WIDTHM)
261. _(.07)P(segs,petstart/segs,petduration,petmax/segs)_ (0)@v
262. [+ (20,TRI_ARate,angle1)F(PStart,PDuration,PMax)@v&(5)
263. [-l(pt,leafstart,leafduration,leafmax)][~r(pt,leafstart,leafduration,leafmax)]]
264. [- (20,TRI_ARate,angle1)F(PStart,PDuration,PMax)@v&(5)
265. [-l(pt,leafstart,leafduration,leafmax)][~r(pt,leafstart,leafduration,leafmax)]]
266. &(TRI_Angle2)F(PStart*2,PDuration,PMax*2)
267. &(TRI_Angle3)[-L(pt,leafstart,leafduration,leafmax)][~R(pt,leafstart,leafduration,leafmax)]
268.
269. /* petiole segmentation for bending via tropism */
270. P(segs,start,duration,max) : segs>0 --> F(start,duration,max)P(segs-1,start,duration,max)
271. P(segs,start,duration,max) --> *
272.
273. homomorphism /* rules for visualisation */
274.
275. /* opening of leaves */
276. ~c(pt,l,d,m) --> ~c(l)
277. ~l(pt,l,d,m) : pt==ptTruncatula --> \ (90-l/m*90)-l(l)
278. ~r(pt,l,d,m) : pt==ptTruncatula --> / (90-l/m*90)-r(l)

```

```

279. ~L(pt,l,d,m) : pt==ptTruncatula--> /(90-l/m*90)~L(l)
280. ~R(pt,l,d,m) : pt==ptTruncatula--> \ (90-l/m*90)~R(l)
281.
282. /* apex visualisation */
283. A(pt,order,pn,node,d) : pn==0 --> ~l(2,5);(48)@L(" -- m%.f",node-1)
284. A(pt,order,pn,node,d) : pn>2 --> ~l(2,5)
285. A(pt,order,pn,node,d) --> ~l(2,5) ;(48) [@L(" -- m%.f-%.f",pn,node-1)]
286.
287. /* Reproductive visualisation */
288. R(pt,order,node,birthday) : time-birthday<6
289. {
290.   stage=(time-birthday+1)*.1;
291.   size=stage*5;
292.   colour=8; /* bud and green petals */
293.   if (stage>=.3) {colour=colour+1;} /* yellow flower */
294. }
295. --> F(size);(colour)@O(size)
296. R(pt,order,node,birthday) : *
297. {
298.   stage=(time-birthday+1)*.1;
299.   if (stage>.8) {stage=.8;} /* currently limited */
300.   size=stage*2;
301.   colour=10; /* pod */
302. }
303. --> _(.2);(colour)F(size)F(size)F(size)F(size)
304.
305. #define SURFACESIZE 20
306.
307. SURFACE --> [[f(0)];(1)
308.   {&(91)g(SURFACESIZE/2)+(90)f(SURFACESIZE/2)+(90)f(SURFACESIZE)
309.     +(90)f(SURFACESIZE)+(90)f(SURFACESIZE)}]
310.
311. Legend(pt) : pt==ptTruncatula -->
312.   [[f(5)[^(90)f(35)@L("Truncatula")+ (90)f(30);(48)@L("Day %.f", time)]
313.
314. Legend(pt) : pt==ptYourModel -->
315.   [[f(5)[^(90)f(35)@L("Your Model")+ (90)f(30);(48)@L("Day %.f", time)]
316.
317. endsystem

```

VII. References

- Benlloch R, Navarro C, Beltran JP, Canas LA (2003) Floral development of the model legume *Medicago truncatula*: ontogeny studies as a tool to better characterize homeotic mutations. *Sex Plant Reprod.* **15**: 231-241.
- Blake J (2004) Bio-Ontologies – fast and furious. *Nat. Biotechnol.* **22**: 773-774
- Boyes DC, Zayed AM, Ascenzi R, McCaskill AJ, Hoffman NE, Davis KR, Gortlach J (2001) Growth stage-based phenotypic analysis of *Arabidopsis*: A model for high throughput functional genomics in plants. *Plant Cell* **13**: 1499-1510.
- Bucciarelli B, Hanan J, Palmquist D, Vance CP (2006) A standardized method for analysis of *Medicago truncatula* phenotypic development. *Plant Physiol.* **142**: 207-219.
- Coleman JS, McConaughay KDM, Ackerly DD (1994) Interpreting phenotypic variation in plants. *TREE* **9**: 187-191.
- Lesins KA, Lesins I (1979) Genus *Medicago* (Leguminosae). Dr. W. Junk bv Publishers, The Hague.
- Lindenmayer A. (1968) Mathematical models for cellular interaction in development: Parts I and II. *J. Theor. Biol.* **18**: 280–315.
- Moreau D, Salon C, Munier-Jolan N (2006) Using a standard framework for the phenotypic analysis of *Medicago truncatula*: an effective method for characterizing the plant material used for functional genomics approaches. *Plant Cell Environ.* **29**: 1087-1098.
- Mundermann L, Erasmus Y, Lane B, Coen E, Prusinkiewicz P (2005) Quantitative modeling of *Arabidopsis* development. *Plant Physiol.* **139**: 960-968.
- Prusinkiewicz P, Hanan JS, Karwowski MR (2000a) L-studio/cpfg: A software system for modeling plants. In M Nagl, A Schurr, M Munch eds, Lecture notes in computer science 1779: Applications of graph transformation with industrial relevance. Springer-Verlag Press, Berlin, pp 457-464.
- Prusinkiewicz P, Hanan JS, and Mèch R. (2000b) An L-system-based plant modeling language. In M. Nagl and A. Schurr and M. Munch (eds.), Applications of graph transformations with industrial relevance, Lecture Notes in Computer Science 1779, p 395-410, Springer-Verlag, Berlin.
- Prusinkiewicz P, Karwowski R, Mech R, Hanan J (2000c) L-studio/cpfg: A software system for modeling plants. Proceedings of Applications Of Graph Transformations With Industrial Relevance, Lecture Notes In Computer Science, 1779: 457-464, Springer-Verlag, Berlin.
- Prusinkiewicz P (2004) Modeling plant growth and development. *Curr. Opin. Plant Biol.* **7**: 79-83.
- Wang HL, Grusak MA (2005) Structure and development of *Medicago truncatula* pod wall and seed coat. *Ann Bot.* **95**: 737-747.