Ong Yongzhi (Yong)
Department of Ecoinformatics,
Biometrics & Forest Growth
Georg-August University of Göttingen

Presenting…

MTG & Statistical Analysis

Plant Growth/Branching

1. **Encode**
2. **Manipulate**
3. **Reconstruct**
4. **Analyze**

We will look at…

1. **MTG – Multiscale Tree Graph**
2. **DTD – Digitized Tree Data**
3. **Statistical Analysis of Structures**

1. Multiscale Tree Graph (MTG)

Introduction - What?

A model of plant topological structure.

- Decomposition of plant into elementary Constituents.
- Connection with each other.

A Multiscale Model of Plant Topological Structures –
C. Godin, Y. Caraglio 1997

1. Multiscale Tree Graph (MTG)

Introduction - How?

A text file.

- A header section.
- A body section for the topological structure.

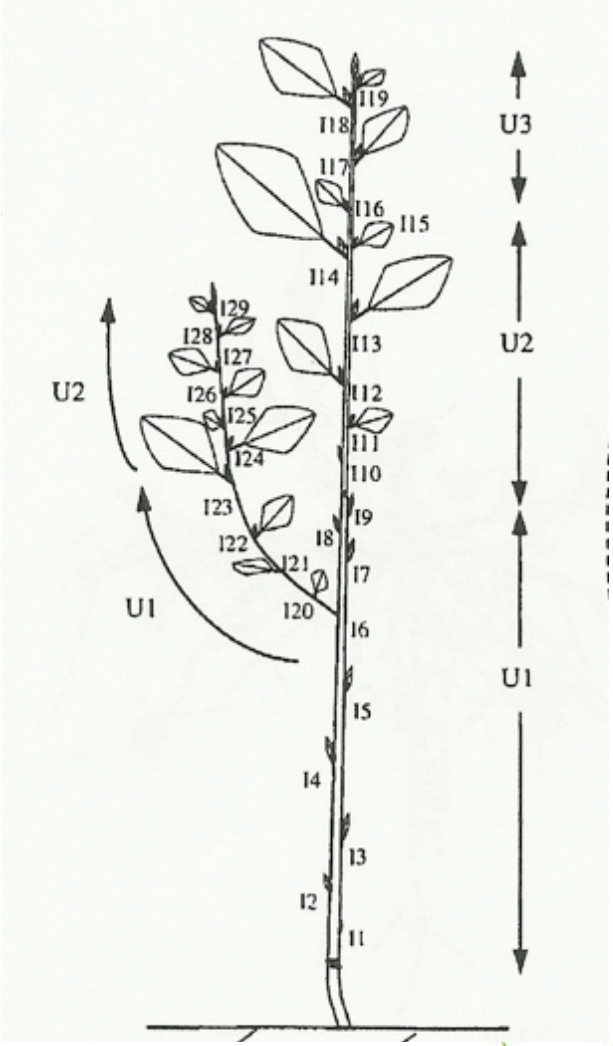
1. Multiscale Tree Graph (MTG)

Introduction - Multiscale?

- Specification of topological elements from various spatial and/or temporal scales.
- E.g. Internodes and growth units.

1. Multiscale Tree Graph (MTG)

Introduction – Multiscale?

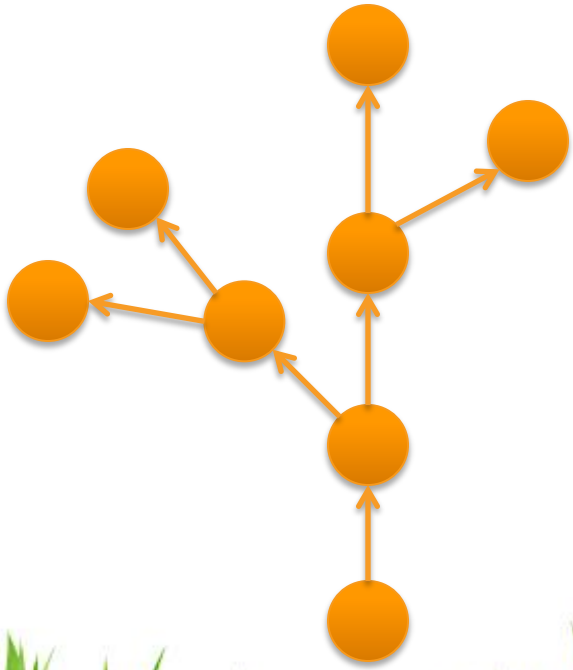


Picture from:
Representing and encoding plant architecture: A Review
- C.Godin, 2000

1. Multiscale Tree Graph (MTG)

Introduction - Tree Graph?

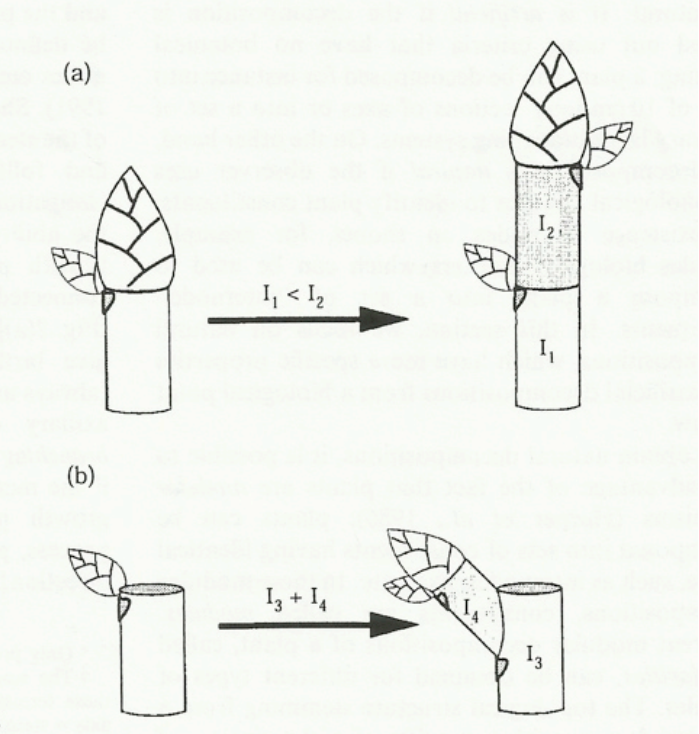
- A *Tree/Graph* data structure.



1. Multiscale Tree Graph (MTG)

Introduction - Tree Graph?

Basic growth processes:

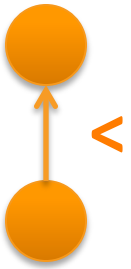


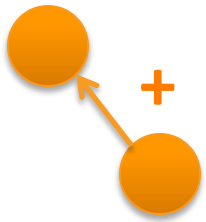
Picture from: Representing and encoding plant architecture: A Review - C.Godin, 2000

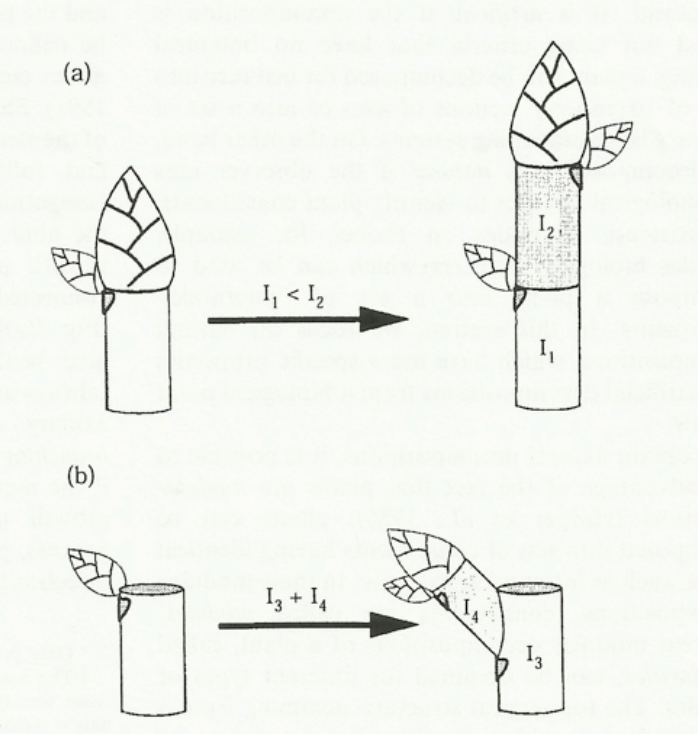
1. Multiscale Tree Graph (MTG)

Introduction - Tree Graph?

Basic growth processes:





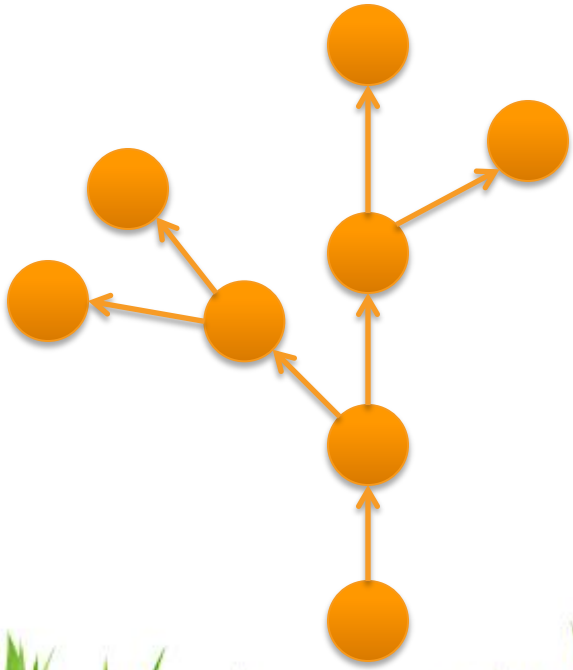


Picture from: Representing and encoding plant architecture: A Review - C.Godin, 2000

1. Multiscale Tree Graph (MTG)

Introduction - Tree Graph?

- A *Tree Graph* data structure naturally represents plant topology.



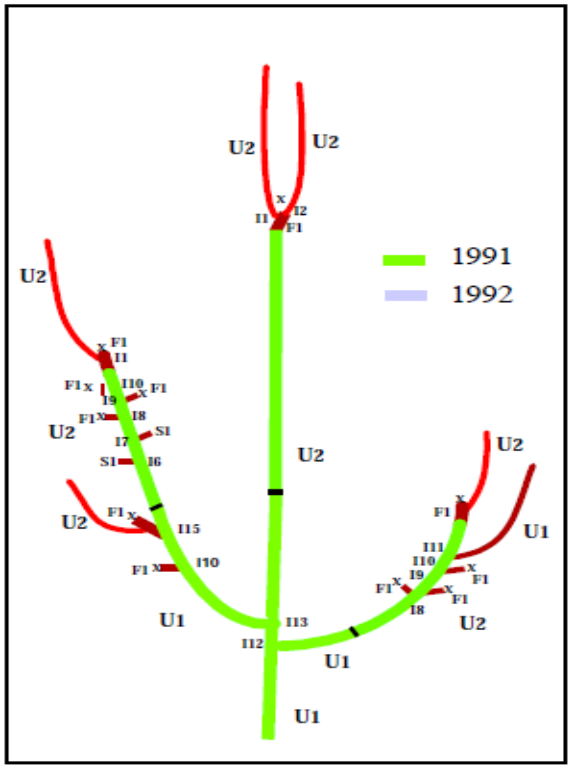
1. Multiscale Tree Graph (MTG)

Encoding - Process Overview

- Plants are broken down into components of different scales.
- Components are given labels that specify their types (E.g. U = growth unit, F = flowering site, S = short shoot, I = internode)

1. Multiscale Tree Graph (MTG)

Encoding - Process Overview



Pictures from:
http://openalea.gforge.inria.fr/, extracted 22 Feb 2012

1. Multiscale Tree Graph (MTG)

Encoding - Process Overview

- From the base of the trunk.
- Symbols representing each vertex and its relationship to its father recorded.

1. Multiscale Tree Graph (MTG)

Encoding

- A plant multiscale topology can be represented by a string of characters.
- The string is made up of a series of labels representing plant components.

/I1<I2<I3<I4+I5<I6

1. Multiscale Tree Graph (MTG)

Encoding

The components are related to each other by:

- ‘/’ - decomposition
- ‘<‘ - successor
- ‘+’ - branching

/I1<I2<I3<I4+I5<I6

1. Multiscale Tree Graph (MTG)

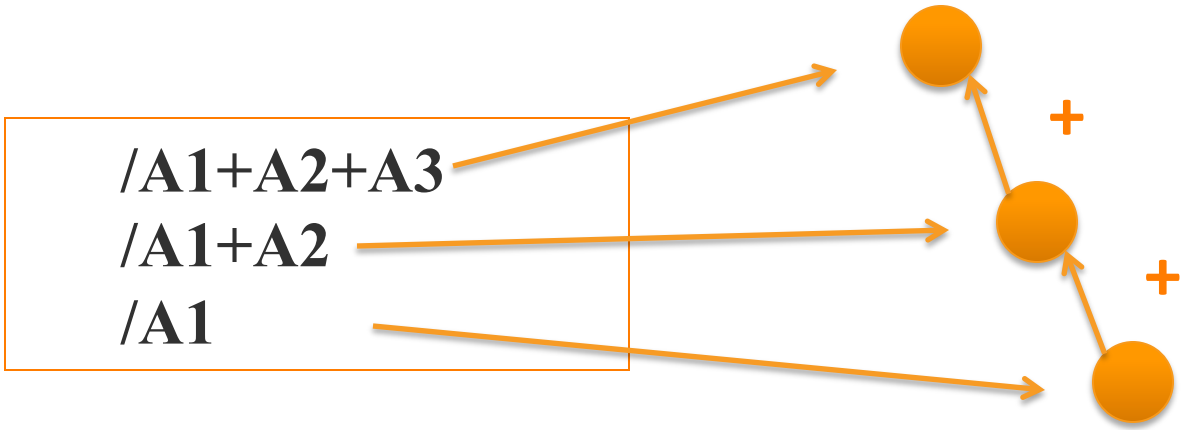
Encoding

The name of an entity is built by concatenating the consecutive entity labels encountered while moving along the plant structure from the plant basis to the considered entity.

/A1
/A1+A2
/A1+A2+A3

1. Multiscale Tree Graph (MTG)

Encoding



1. Multiscale Tree Graph (MTG)

Encoding - Bifurcation

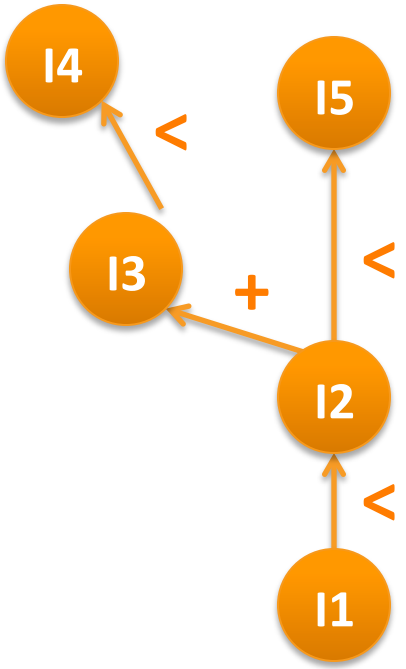
- A square bracket is used when a bifurcation point is encountered.
- A square bracket is closed each time a terminal vertex has been visited.
- The process is then back-tracked to the last bifurcation point.

/I1<I2<[+I3<I4]<I5

1. Multiscale Tree Graph (MTG)

Encoding - Bifurcation

/I1<I2<[+I3<I4]<I5



1. Multiscale Tree Graph (MTG)

Encoding - Decomposition

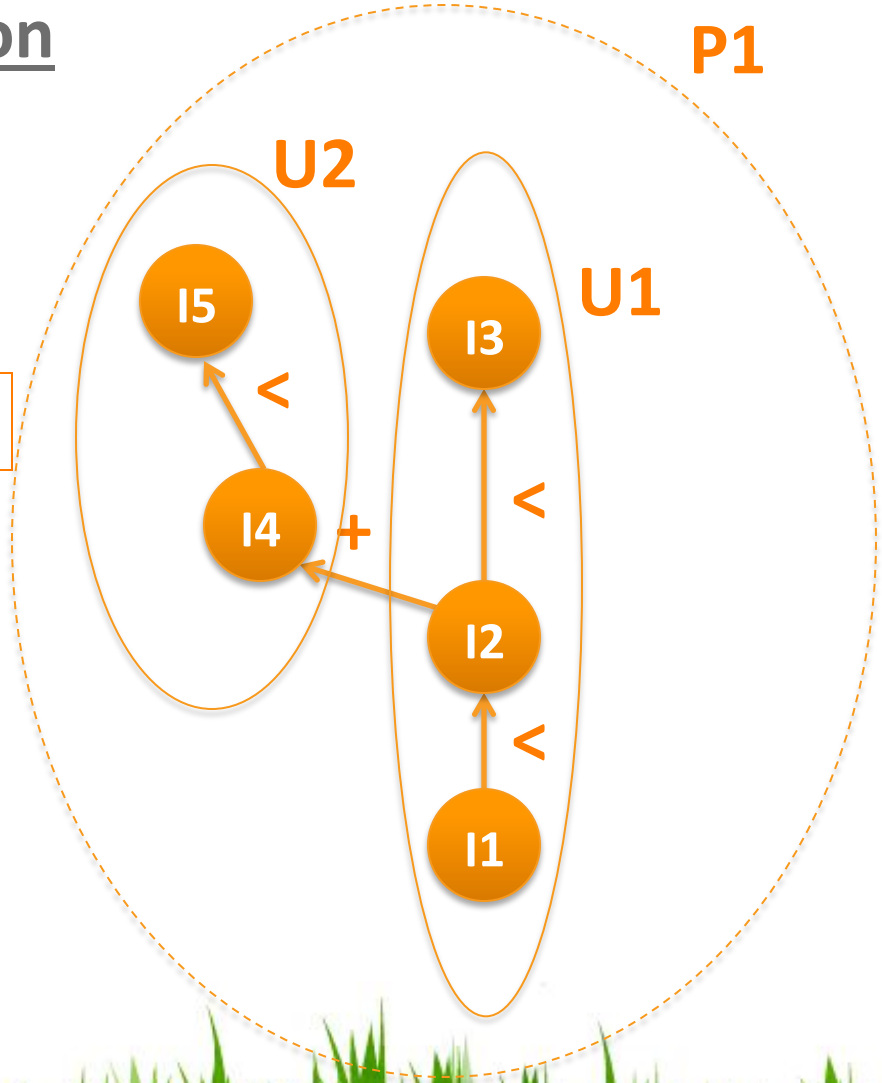
Use of the decomposition symbol is best demonstrated with an example.

/P1/U1/I1<I2[+U2/I4<I5]<I3

1. Multiscale Tree Graph (MTG)

Encoding - Decomposition

/P1/U1/I1<I2[+U2/I4<I5]<I3



1. Multiscale Tree Graph (MTG)

Encoding - Readability

- Long sequences of strings = difficult to read.
- Each square bracket is replaced by a new line plus an indentation level corresponding to the nested level of the square bracket.

/P1/U1/I1<I2[+U2/I4<I5]<I3

/P1/U1/I1<I2
+U2/I4<I5
<I3

1. Multiscale Tree Graph (MTG)

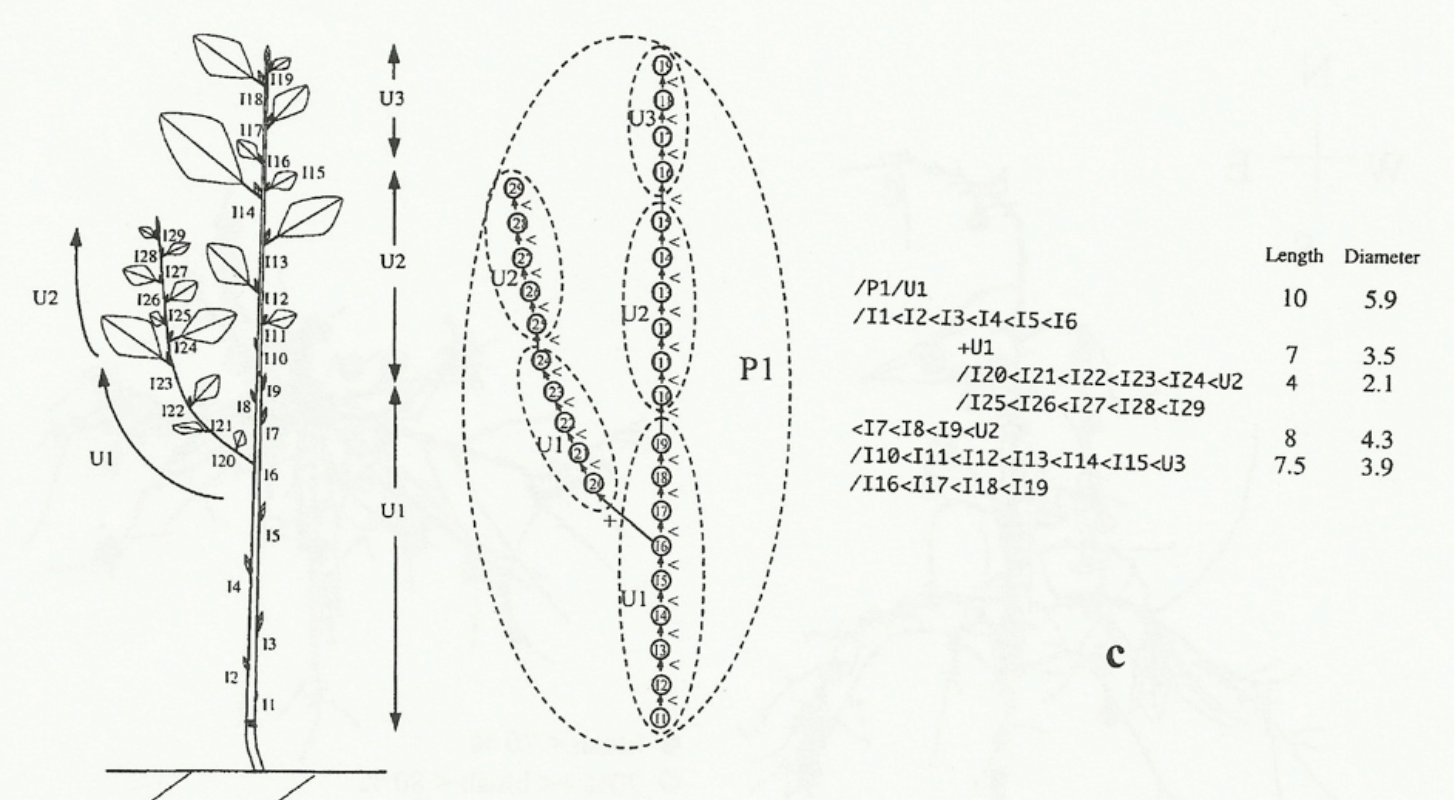
Encoding – Feature values

- Similarly, a new line is used when feature values are assigned to the entity.

	Length	Diameter
/P1/U1/I1	10	10
/P1/U1/I1<I2	8	9
+U2/I4	8	7
+U2/I4<I5	7	6
<I3	8	8

1. Multiscale Tree Graph (MTG)

Encoding – Results



	Length	Diameter
/P1/U1	10	5.9
/I1<I2<I3<I4<I5<I6		
+U1	7	3.5
/I20<I21<I22<I23<I24<U2	4	2.1
/I25<I26<I27<I28<I29		
<I7<I8<I9<U2	8	4.3
/I10<I11<I12<I13<I14<I15<U3	7.5	3.9
/I16<I17<I18<I19		

Picture from: Representing and encoding plant architecture: A Review - C.Godin, 2000

1. Multiscale Tree Graph (MTG)

What about the file?

1. Multiscale Tree Graph (MTG)

File Header

CODE: FORM-A

1. Multiscale Tree Graph (MTG)

File Header

CODE: FORM-A

CLASSES

SYMBOL	SCALE	DECOMPOSITION	INDEXATION	DEFINITION
$	0	FREE	FREE	IMPLICIT
P	1	CONNECTED	FREE	IMPLICIT
U	2	<-LINEAR	FREE	EXPLICIT
I	2	<-LINEAR	FREE	EXPLICIT
E	3	NONE	FREE	IMPLICIT

1. Multiscale Tree Graph (MTG)

File Header

DESCRIPTION:

LEFT	RIGHT	RELTYPE	MAX
U	U,I	+	?
U	U,I	<	1
I	I	+	?
E	E	<	1
E	E	+	1

1. Multiscale Tree Graph (MTG)

File Header

FEATURES:
NAME TYPE

Alias	STRING
Date	DD/MM
State	STRING
len	INT
TopDiameter	REAL
geom	GEOMETRY geom1.geom
appear	APPEARANCE material.app

1. Multiscale Tree Graph (MTG)

File Body

```
MTG:
TOPO
#column1   #column2   #column3   #column4   #column5
/P1/A1
   /U1<U2
      +S1
      +S2
      +A1/U1<U2+S1
      <U3
         +A2/U1<U2<U3
```

1. Multiscale Tree Graph (MTG)

Importing MTG in GroIMP

- GroIMP reads the MTG file and stores the information in 2 types of nodes:
    - MTG root node (de.grogra.mtg.nodes.MTGRoot)
    - MTG node (de.grogra.mtg.MTGNode)

1. Multiscale Tree Graph (MTG)

GroIMP MTG Root Node

- The Root Node contains information of:
  - What is in the file header
- The Root Node also contains:
  - Basic functions

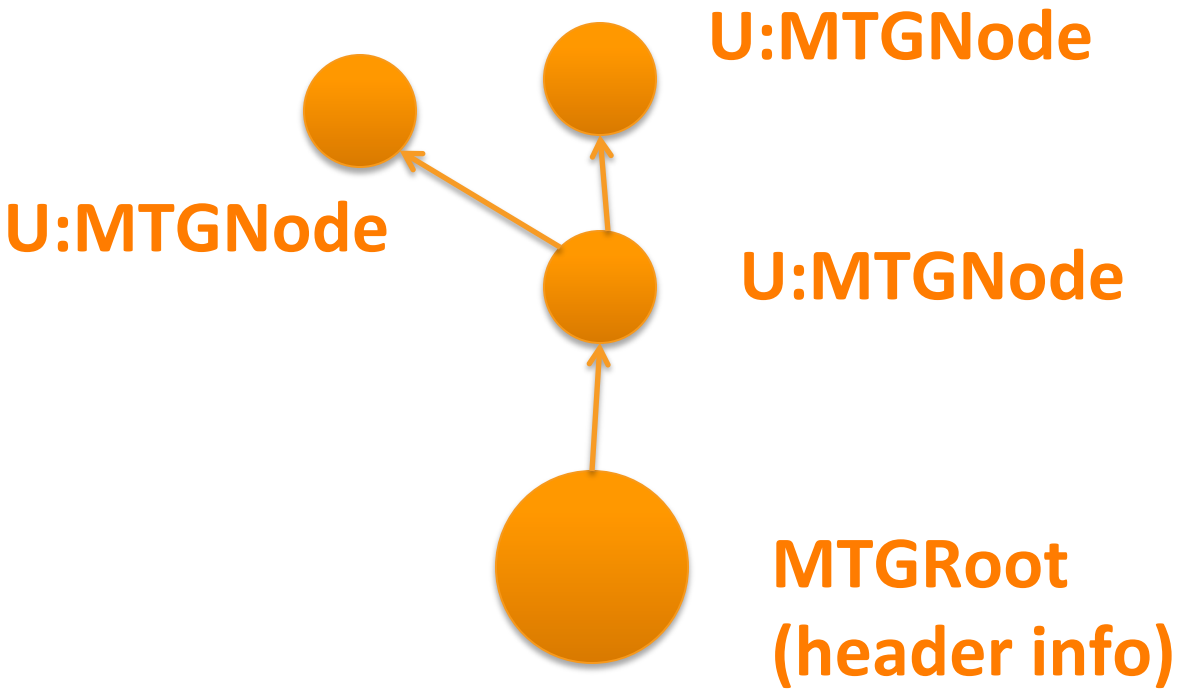
1. Multiscale Tree Graph (MTG)

GroIMP MTG Node

- The Node = element in MTG structure.
- The Node contains - attribute values of the MTG element.
- Contrary to the original implementation, memory is not allocated to attribute values not in the file.

1. Multiscale Tree Graph (MTG)

GroIMP MTG representation



1. Multiscale Tree Graph (MTG)

GroIMP – MTG features

- On-going work to implement more functionality and visualization features of MTG.

- See MTG graph and visualization in GroIMP…

1. Multiscale Tree Graph (MTG)

- Details at:

http://openalea.gforge.inria.fr/doc/vplants/newmtg/doc/_build/html/user/quick_start.html

2. Digitized Tree Data (DTD)

- Standard ASCII text file format.
- Each Growth Unit is described in a separate line.

Information and pictures in this section from:
Shoot Growth and Branching of young Spruce Trees as a function of The two predictors "Shading" und "Growth Density": Data Processing and Analysis with GROGRA (in German) – W.Kurth, G.A.Jürgenson, 1997

2. Digitized Tree Data (DTD)

- Each line begins with an identifier.
- Identifier consists of alpha-numeric characters.
- Each identifier is unique.

C1 L80 #a A25 – W50 D1.1

2. Digitized Tree Data (DTD)

- 2nd specifies length.
- No units specified but normally in mm.
- Begins with ‘L’.

C1 L80 #a A25 – W50 D1.1

2. Digitized Tree Data (DTD)

- 3rd specifies the identifier of the parent Growth Unit.

- Begins with '#'.

C1 L80 #a A25 – W50 D1.1

2. Digitized Tree Data (DTD)

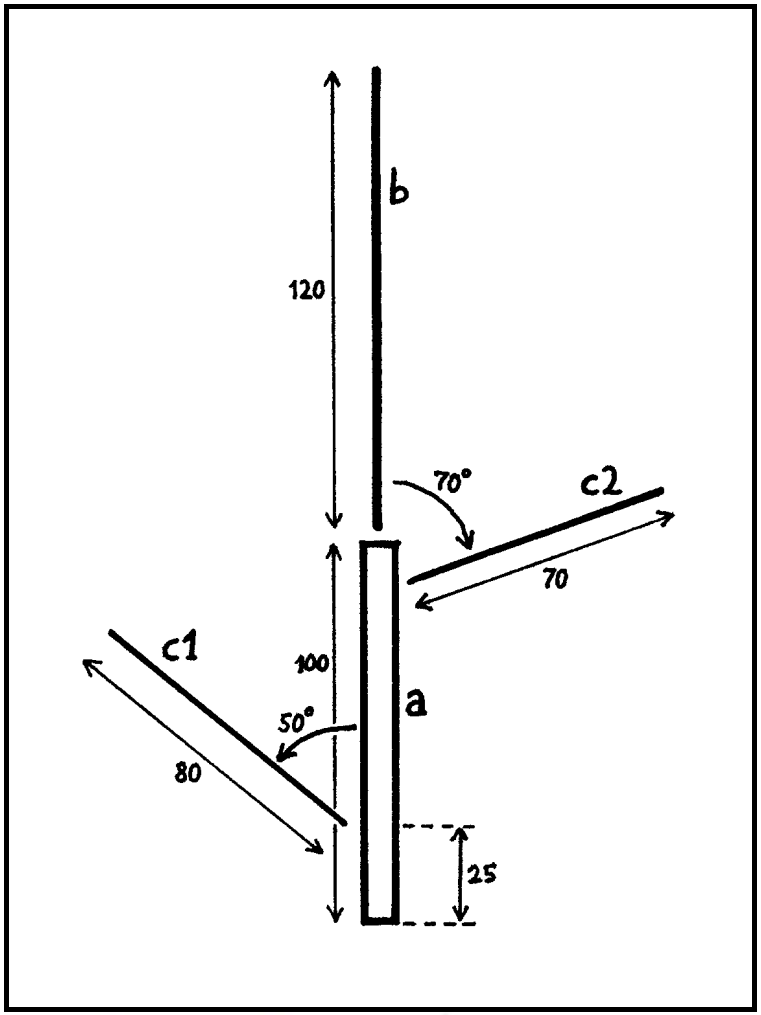
- Followed by optional specifications, e.g.:

  - A : Distance from base of mother unit.
  - B : Number of leaves.
  - C : Color index
  - D : Diameter
  - W : Branching angle
  - etc.

C1 L80 #a A25 – W50 D1.1

2. Digitized Tree Data (DTD)

a	L100 ##	D2.5
b	L120 #a V	D1.2
c1	L80 #a A25 – W50	D1.1
c2	L70 #a A90 + W70	D1.0



2. Digitized Tree Data (DTD)

- Differences with MTG:
  - Each line is independent of previous.
  - Attribute set is fixed.

2. Digitized Tree Data (DTD)

- Details at:

http://wwwuser.gwdg.de/~groimp/grogra.de/documentation/dtd_code.pdf

3. Statistical Analysis

3. Statistical Analysis

Example file:

- Found at:

http://openalea.gforge.inria.fr/doc/vplants/newmtg/doc/_build/html/_downloads/agraf.mtg

3. Statistical Analysis

Branch Length - Computation

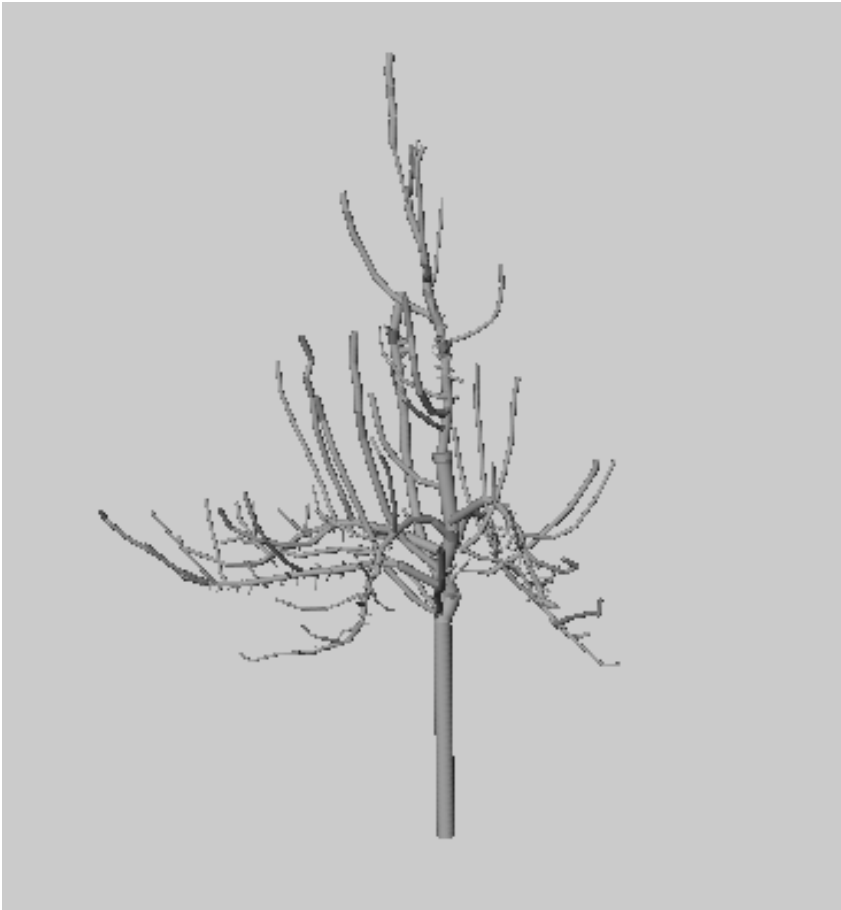
- In XL console:

```
import de.grogra.mtg.nodes.MTGRoot;
import de.grogra.mtg.MTGKeys;

[a:MTGRoot ::> a.plantFrame(3,400);]
```

3. Statistical Analysis

Branch Length – Computation



3. Statistical Analysis

Branch Length - Mean and Std. Deviation

- In XL console:

```
import de.grogra.mtg.nodes.E;

statistics((
(* a:E, (a.getObject(MTGKeys.ATT_LENGTH)!=null)*),
((Float)a.getObject(MTGKeys.ATT_LENGTH)).floatValue()
));
```

3. Statistical Analysis

Branch Length - Analysis

- In XL console:

2935 values in [0.0, 248.68669] :
sum=40246.887,
mean=13.712738,
deviation=11.0813265(80.810455%),
skewness=5.2317533

3. Statistical Analysis

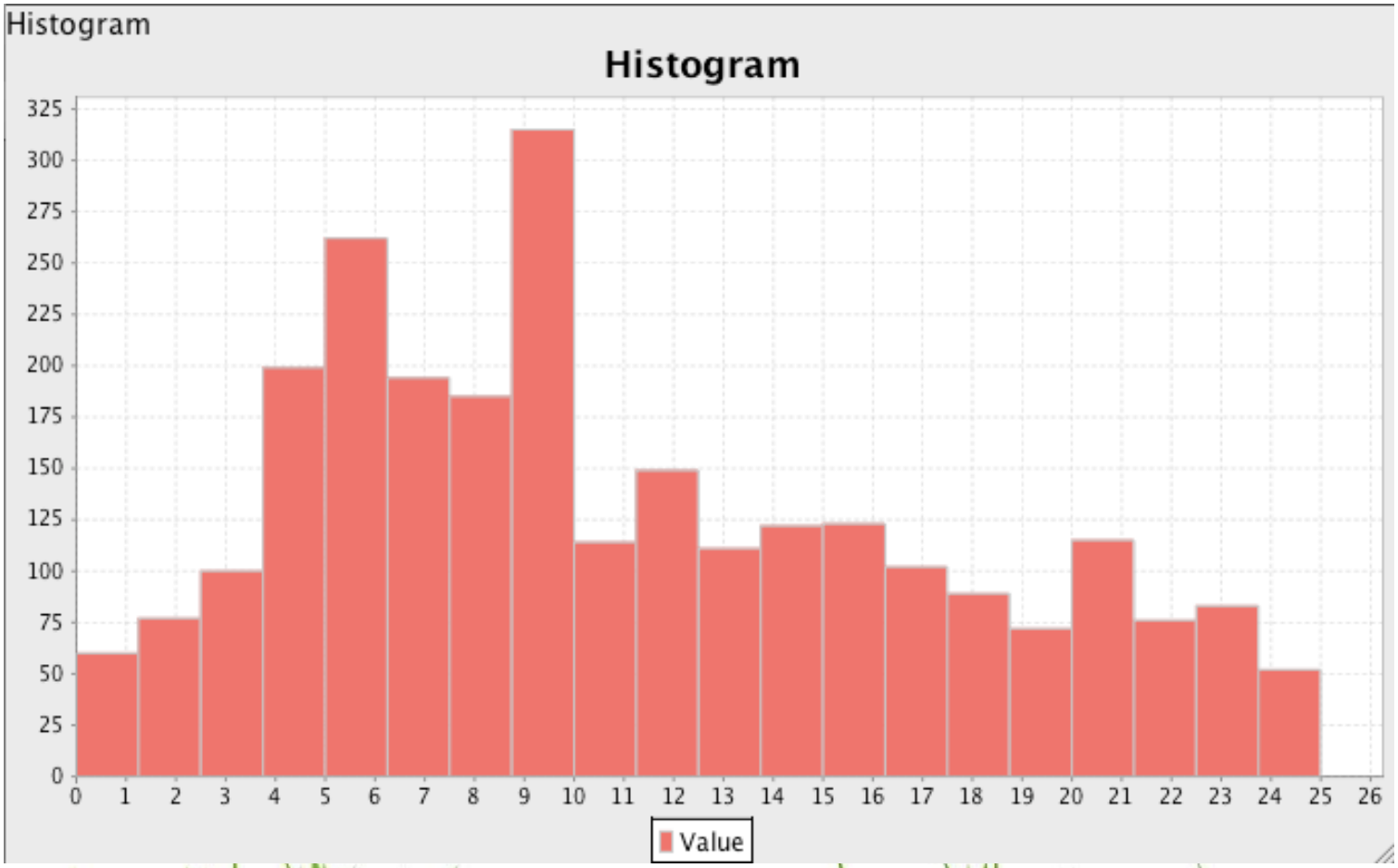
Branch Length – Frequency Distribution

- In XL console:

```
import de.grogra.pf.data.DatasetRef;
import de.grogra.pf.ui.ChartPanel;
DatasetRef myChartHISTO = new DatasetRef("Histogram");
myChartHISTO.clear().setColumnKey(0,"Value");
myChartHISTO.setHistogramBins(0, 0, 25, 20);
//int col, double min, double max, int count
myChartHISTO << (*E*).getLength();
chart(myChartHISTO, HISTOGRAM);
```

3. Statistical Analysis

Branch Length – Frequency Distribution



3. Statistical Analysis

Length-Diameter Correlation

- In XL console:

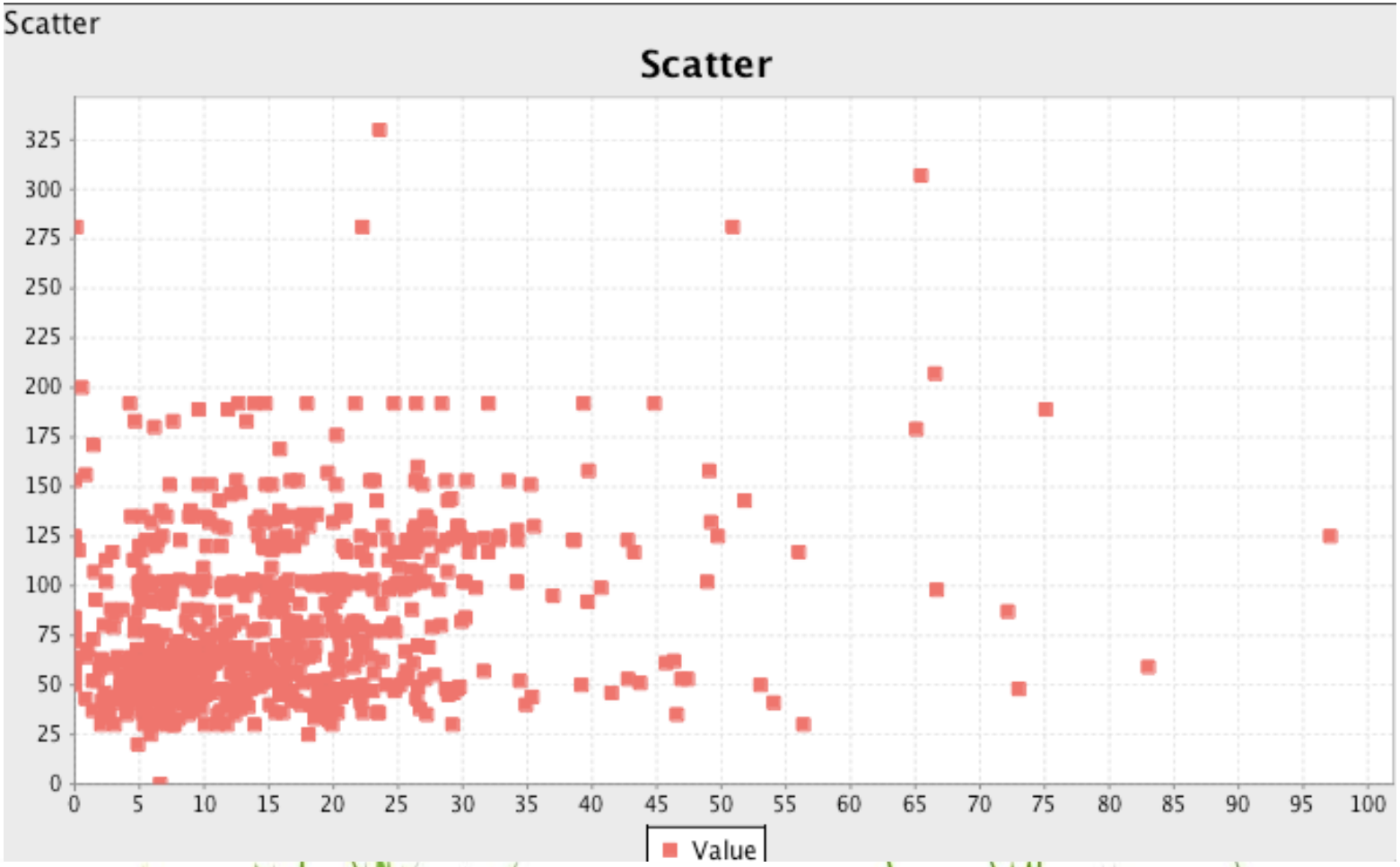
```
DatasetRef myChartXY = new DatasetRef("Scatter");
myChartXY.clear().setColumnKey(0,"Value");

[c:E,(c.getObject(MTGKeys.ATT_TOPDIA)!=null) ::>
myChartXY.addRow().
(set(0, c.getLength().floatValue(),
((Integer)(c.getObject(MTGKeys.ATT_TOPDIA))).
floatValue()));]

chart(myChartXY, SCATTER_PLOT);
```

3. Statistical Analysis

Length-Diameter Correlation



3. Statistical Analysis

Branching Angle

```
statistics(((* c1:E [c2:E],
((c1.getObject(MTGKeys.ATT_DIRECTION_PRI)
!=null)
&&(c2.getObject(MTGKeys.ATT_DIRECTION_PRI)
!=null))*),
((Vector3d)c1.getObject
(MTGKeys.ATT_DIRECTION_PRI)).angle(
((Vector3d)c2.getObject
(MTGKeys.ATT_DIRECTION_PRI)))));
```

3. Statistical Analysis

Branching Angle - Analysis

- In XL console:

443 values in [0.0, 3.1415927] :
sum=550.99335,
mean=1.2437773,
deviation=0.755851(60.770603%),
skewness=0.27995378

3. Statistical Analysis

Branching Angle – Frequency Distribution

- In XL console, clear previous data set:

```
myChartHISTO.clear().setColumnKey(0,"Value");
myChartHISTO.setHistogramBins(0, 0, 3.1415927, 10);
//int col, double min, double max, int count
```

3. Statistical Analysis

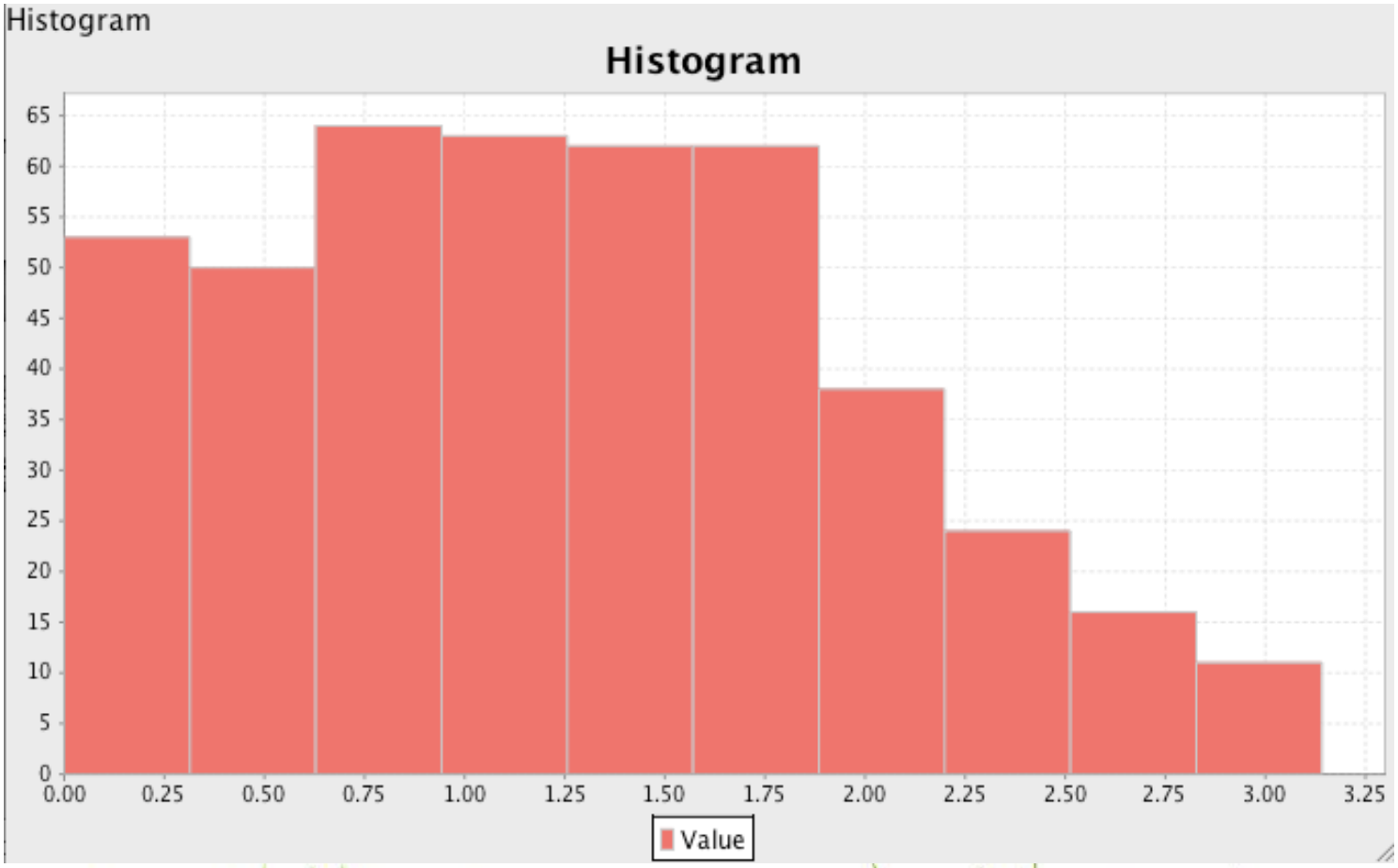
Branching Angle – Frequency Distribution

- In XL console, set angle in data set:

```
[c1:E [c2:E], ((c1.getObject(
MTGKeys.ATT_DIRECTION_PRI)!=null)
&&(c2.getObject(MTGKeys.ATT_DIRECTION_PRI)!=null))
 ::>
myChartHISTO.addRow().set(
0,((Vector3d)c1.getObject(
MTGKeys.ATT_DIRECTION_PRI)).angle(
((Vector3d)c2.getObject(
MTGKeys.ATT_DIRECTION_PRI))));]
```

3. Statistical Analysis

Branching Angle – Frequency Distribution



We have seen…

1. Encode
2. Manipulate
3. Reconstruct
4. Analyze

Thank you!

The End