NEURON

http://fourcoffees.com/project/neuron/

Neuron is a set of Perl programs and modules allowing for the statistical analysis of virtual neurons. With it, one can measure certain standalone physical attributes of these neurons – of the complete dendritic tree as well as only parts of it – as well as generate histograms.

Gunther Schmidl
gschmidl@gmx.at

Structure of *Neuron*

The following files make up the functionality of Neuron:

Modules

- 🗁 Neuron
  - 🗁 Morphometrics
    - 🗎 Bifurcation.pm – Statistics on bifurcations
    - 🗎 Compartment.pm – Statistics on compartments
    - 🗎 EntireNeuron.pm – Statistics on the entire neuron
    - 🗎 Segment.pm – Statistics on segments
    - 🗎 Tools.pm – Initialisation and generic routines
  - 🗁 Statistics
    - 🗎 StdDev.pm – Calculate Standard Deviation
  - 🗁 SWC
    - 🗎 Read.pm – Read and parse a SWC file
    - 🗎 Write.pm – Write a SWC file
  - 🗁 XML
    - 🗎 Write.pm – Write data in XML format, including DTD

Programs

- 🗎 MorphometricsList.pl – Calculate all morphometrics for a neuron
- 🗎 SimpleHistogram.pl – Generate simple histograms from neuronal data

While the programs can be used standalone, the modules provide exported functions and can not be used directly.

Since the code is written in Perl, modules are referenced in the standard way through subdirectories, so the XML writer would be addressed as Neuron::XML::Write, while the Tools module would be addressed as Neuron::Morphometrics::Tools.

Program Reference

MorphometricsList

Usage: `MorphometricsList <inputfile>`

Parameters:

`inputfile`	data in SWC format[*]

- This simple program outputs all available morphometric data for a given SWC file. Output will look similar to the example on page 4.

SimpleHistogram

Usage: `SimpleHistogram [-l<n> -h<n> -w<var>] [-s<n> | -z<n>] [-t] [-m<var> | -m?] <inputfile>`

Parameters:

`inputfile`	data in SWC format[*]
`-m`*var*	measurement (use -m? for a full list)[*]
`-h`*n*	upper limit for variable[**]
`-l`*n*	lower limit for variable[**]
`-w`*var*	variable for limit (order, distance, or edistance)[**]
`-s`*n*	number of bins[***]
`-z`*n*	size of bins[***]
`-t1`	generate tab-delimited instead of comma-delimited output

- This program allows the generation of simple histograms, which divide the range of a selected morphometric into a number of bins; either this number or the bin size can be chosen through the `-s` and `-z` parameters. Example output can be seen on page 5.
- The morphometric can be limited to a range of order, distance or Euclidian distance if required; this is achieved using the `-h`, `-l` and `-w` parameters.
- Standard output is in the CSV (comma-separated value) format; using the `-t1` parameter, output can be forced to tabulation-separated value format.

Possible values for `-m`:

SEGMENTS			
contraction	segment contraction	distance	path distance from soma
edistance	Euclidian distance from soma	elength	Euclidian length
endd	final diameter	fragmentation	number of compartments
initiald	initial diameter	length	path length
order	branch order	taper	segment taper
BIFURCATIONS			
btangle	bifurcation torque angle	caangle	compartment amplitude angle
dratio	parent/child ratio	kidratio	daughter ratio
pc	power correction	rall	Rall’s value
saangle	segment amplitude angle	tilt	tilt

[*] required parameter
[**] if used at all, all these parameters must be used together
[***] these parameters are mutually exclusive

Sample Output from MorphometricsList

```
=========================================================
STATISTICS ON ENTIRE NEURON
=========================================================
Number of stems        = 3
Number of terminals    = 71
Minimum X              = -174.363
Maximum X              = 195.311
Minimum Y              = -245.718
Maximum Y              = 498.191
Minimum Z              = -142
Maximum Z              = 109.45
Minimum branch order   = 0
Maximum branch order   = 10
Average branch order   = 5.44704570791527
 standard deviation    = 2.35023980509486
Volume                 = 11711.4661192198
Area                   = 35231.7723155906
=========================================================
STATISTICS ON COMPARTMENTS
=========================================================
Number of compartments = 2691
Minimum diameter       = 0.16
Maximum diameter       = 13.2
Average diameter       = 0.823730955035304
Average volume         = 4.35208700082489
Average area           = 13.092446048157
Min path distance      = 0
Max path distance      = 719.976852530749
Avg path distance      = 315.399314202408
Min eucl. distance     = 0
Max eucl. distance     = 513.267715238354
Avg eucl. distance     = 211.246640182865
Min comp. length       = 0
Max comp. length       = 54.5492265481372
Avg comp. length       = 5.36759427076911
Min Burke's Taper Rate = -0.347918001507942
Max Burke's Taper Rate = 5.3781512605042
Avg Burke's Taper Rate = 0.0129003712481163
 standard deviation    = 0.556645264676897
=========================================================
STATISTICS ON BIFURCATIONS
=========================================================
Number of bifurcations = 68
Average daughter ratio = 0.716219985137401
 standard deviation    = 0.252836348296231
Asymmetry              = 0.510234042777143
Avg comp. ampl. angle  = 94.9669073070367
 standard deviation    = 43.9370510942587
Avg seg. ampl. angle   = 44.1554988379234
 standard deviation    = 25.6253447010115
Avg bifurcation torque = 123.709540213384
 standard deviation    = 32.247645100205
Tilt                   = 86.1429126028451
 standard deviation    = 46.6226626639095
Rall's Value           = 2.436075
 standard deviation    = 0.843875031847133
Power Correction       = 0.883370554525263
 standard deviation    = 0.490862298213262
P/C diameter ratio     = 1.44740524518819
 standard deviation    = 0.580092062308889
=========================================================
STATISTICS ON SEGMENTS
=========================================================
Number of segments     = 140
Min Segment Taper      = 0.325757575757576
Max Segment Taper      = 1.66666666666667
Avg Segment Taper      = 0.922672318238526
 standard deviation    = 0.242499327889654
Min Fragmentation      = 1
Max Fragmentation      = 66
Avg Fragmentation      = 19.3525179856115
 standard deviation    = 13.6433836756202
Contraction            = 0.615466397143374
 standard deviation    = 0.225738943314012
Min eucl. seg. length  = 0
Max eucl. seg. length  = 161.299773636249
```

```
Avg eucl. seg. length  = 59.1871781727122
 standard deviation    = 48.221836874595
Min seg. path length   = 0
Max seg. path length   = 275.149772199891
Avg seg. path length   = 100.609131684065
 standard deviation    = 74.3034654062193
Min eucl. distance     = 2.394
Max eucl. distance     = 435.608207046194
Avg eucl. distance     = 146.852792459743
 standard deviation    = 92.8675577916519
Min path distance      = 2.394
Max path distance      = 604.854527056673
Avg path distance      = 211.624973984047
 standard deviation    = 121.319601593514
```

Sample Output from SimpleHistogram

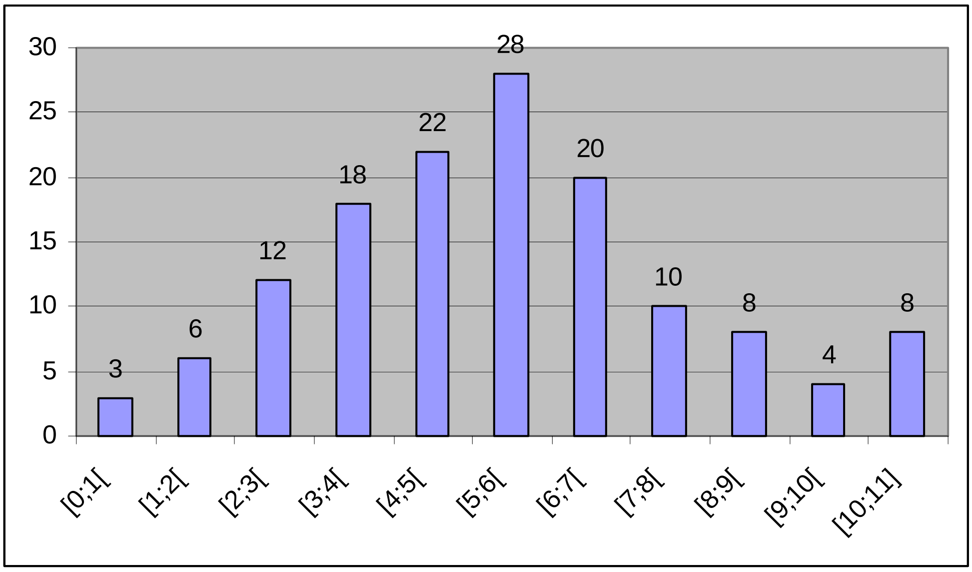
SimpleHistogram was run on an example neuron to create bins of size 1 on the “branch order” morphometric, thus listing the number of segments per branch order:

```
>SimpleHistogram.pl -morder -z1 s.swc
SimpleHistogram 1.0 (Neuron 0.1)
Initialising... done.
Sampling between order of 0 and 10 (11 bins of size 1).
Output file is s.csv.
All done.
```

This is the output file in CSV format:

```
[0;1[,3
[1;2[,6
[2;3[,12
[3;4[,18
[4;5[,22
[5;6[,28
[6;7[,20
[7;8[,10
[8;9[,8
[9;10[,4
[10;11],8
```

This is an example diagram of the output, produced with Microsoft Excel:



The SWC Format

Neuron deals with virtual neuron files in the SWC format.

SWC is a whitespace-delimited format for describing the structure of a neuron in the simplest possible way. Each line describes the properties of a single *compartment*, an arbitrarily sized piece of a *segment*. Many of these segments, each of which may either branch into two further segments at a *bifurcation* or end at a *terminal* (both of which are also represented as compartments), make up the entire neuron.

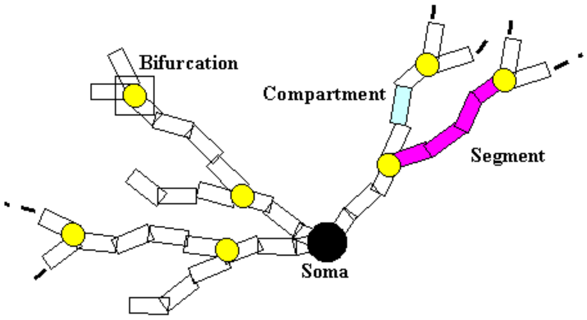


Fig. 1: Definitions

The format of a line in an SWC file is as follows:

```
51   3   -91.73   -93.645   -20.25   0.24   50
```

In order, these numbers mean:

1) The number of the compartment, starting from 1 (which represents the soma)
2) A flag providing the location of the compartment (irrelevant for *Neuron*)
3) X coordinate of compartment
4) Y coordinate of compartment
5) Z coordinate of compartment
6) Diameter of compartment
7) Parent compartment (the soma's parent is defined as –1)

Module Reference

This reference lists all the functions exported by the various modules and explains briefly how they work.

All morphometric functions not in Neuron::Morphometrics::EntireNeuron allow the limitation of measurements to a range of branch order, path distance or Euclidian distance.

The optional parameters *min* and *max* define limits in the parameter specified by *which*: branch order ("*order*"), path distance ("*distance*") or Euclidian distance ("*edistance*"). If parameters are given at all, both *min* and *max* must be given.

Neuron::Morphometrics::Bifurcation

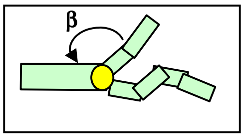
avgAsymmetry(neuron [, min , max, which])

Return average asymmetry of neuron. This is defined as |l-d| / (l-d+2), where l and d are the number of terminals on either of a bifurcation.

avgDaughterRatio(neuron [, min , max, which])

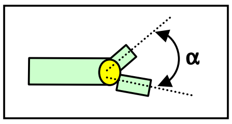
Return average daughter ratio of neuron. This is defined as the diameter of one daughter divided by the diameter of the other.

bifurcationTorqueAngle(neuron [, min , max, which])



Return average bifurcation torque angle β of bifurcations. The returned angle is the one between the parent and the vector that is in the middle of the two daughter vectors[1].

compartmentAmplitudeAngle(neuron [, min , max, which])



Return average amplitude angle of bifurcations α with respect to their daughter compartments[1].

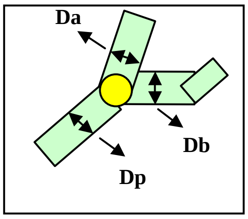
diameterRatio(neuron [, min , max, which])

Return average parent/child diameter ratio (the average of the parent’s diameter divided by each of the childs’ diameters).

numBifurcations(neuron [, min , max, which])

Return number of bifurcations.

powerCorrection(neuron, rall [, min , max, which])

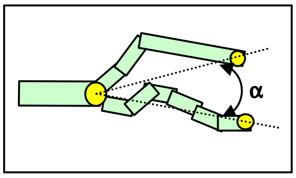


Return average power correction for a given Rall's Value v.
This is defined as $D_p^v = D_a^v + D_b^v$.

Rall(neuron [, min , max, which])

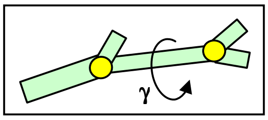
Return Rall's Value v, which is defined as v | $D_p^v = D_a^v + D_b^v$
Since this value is brute-forced, it takes up to several seconds to calculate for large Neurons[2].

segmentAmplitudeAngle(neuron [, min , max, which])



Return average amplitude angle of bifurcations α’ with respect to their daughter segments[1].

tilt(neuron [, min , max, which])



Return average tilt γ between bifurcations[1,3].

---

[1] Note: Sometimes the divisor of the will come out as 0, and the bifurcation will not be included in calculations.
[2] *Note*: Sometimes there is no local minimum that fits the definition of a valid Rall’s value, and that bifurcation will not be included in calculations. This is reflected by setting the value to –1 in `@FORKLIST`.
[3] *Note*: if a bifurcation appears directly at the soma, tilt can not be calculated for it, and that bifurcation will not be included in calculations. This is reflected by setting the value to –1 in `@FORKLIST`.

Neuron::Morphometrics::Compartment

avgArea(neuron [, min , max, which])

Returns average compartment area.

avgVolume(neuron [, min , max, which])

Returns average compartment volume.

(min|max|avg)BurkeTaperRate(neuron [, min , max, which])

Returns minimal, maximal or average Burke's Taper Rate of compartments. This is defined as (daughter diameter – parent diameter) / (daughter length + parent length)[4].

(min|max|avg)CompartmentEuclidianDistance(neuron [, min [, max], which])

Returns minimal, maximal or average euclidian distance of compartments from soma.

(min|max|avg)CompartmentLength(neuron [, min , max, which])

Returns minimal, maximal or average length (path length equals euclidian length) of compartments.

(min|max|avg)CompartmentPathDistance(neuron [, min , max, which])

Returns minimal, maximal or average path distance of compartments.

(min|max|avg)Diameter(neuron [, min , max, which])

Returns minimal, maximal or average diameter of compartments.

Neuron::Morphometrics::EntireNeuron

numStems(neuron)

Returns the number of stems in the neuron.

numTerminals(neuron [, min , max, which])

Returns the number of terminal tips in the neuron.

area(neuron)

Returns the neuron's surface area.

volume(neuron)

Returns the neuron's volume.

(min|max)(X|Y|Z)

Returns minimal or maximal X, Y or Z value[5].

---

[4] *Note*: this value can be negative if the daughter diameter is larger than the parent diameter.

[5] *Note*: any of these values can be negative.

Neuron::Morphometrics::Segment

numSegments(neuron [, min , max, which])

Return number of segments.

contraction(neuron [, min , max, which])

Returns average contraction of segments. This is defined as the ratio of Euclidian distance to path distance of the segment.

(min|max|avg)Fragmentation(neuron [, min , max, which])

Returns minimal, maximal or average fragmentation. This is defined as compartments per segment.

(min|max|avg)SegmentEuclidianLength(neuron [, min , max, which])

Returns minimal, maximal or average euclidian segment length.

(min|max|avg)SegmentPathLength(neuron [, min , max, which])

Returns minimal, maximal or average segment path length.

(min|max|avg)SegmentTaper(neuron [, min , max, which])

Returns minimal, maximal or average segment taper. This is defined as segment end diameter divided by segment start diameter.

Neuron::Morphometrics::Tools

initSegments(neuron)

Initialises segments; puts a list of all segments into the exported variable `@SEGLIST`; puts a list of all bifurcations into the exported variable `@FORKLIST`; automatically called from Neuron::SWC::Read[6].

numCompartments(neuron)

Returns the number of compartments in the neuron.

(min|max|avg)BranchOrder(neuron)

Returns minimal, maximal or average branch order in the neuron.

Neuron::Statistics::StdDev

stdDev(neuron, parameter [, min, max, which])

Returns the standard deviation for *parameter*.

---

[6] See below for information on `@SEGLIST` and `@FORKLIST`.

Neuron::SWC::Read

readHeader(file)

Parses the file's informational header and returns a hash containing it.

printHeader(header)

Prints information from header hash in a nicely formatted way.

readData(file)

Parses actual neuron data and returns it in an array of hashes[7].

Neuron::SWC::Write

writeHeader(header, file)

Write header hash into SWC file.

writeData(neuron, file)

Writes data from array of hashes into SWC file.

Neuron::XML::Write

writeXMLHeader(header, file)

writes data from array of hashes into XML file linearly. WriteXMLDataTree must also be called in order to get a correct XML file.

writeXMLData(neuron, file)

writes data from array of hashes into XML file linearly.

writeXMLDataTree(neuron, file)

writes data from array of hashes into XML file, preserving tree structure.

---

[7] See `@NEURON`, below, for information on this data structure.

Data Structures

`@NEURON`

This is the main data structure used by virtually all functions. It is comprised of an array of hashes, one array element representing one compartment. Each hash stores the following information about the compartment:

- area
- branch order
- Burke's taper rate
- compartment length
- compartment number
- diameter
- Euclidian distance from soma
- location flag
- parent
- path distance from soma
- segment number
- volume
- x coordinate
- y coordinate
- z coordinate
- an array containing pointers to the compartment's children, if it is a bifurcation

`@SEGLIST`

This is an array of arrays, each one containing a list of the compartments that make up the respective segment. The zeroth element of each list array contains a hash with additional information about the segment. When certain segment morphometric functions are called, more elements are added to the hash.

- branch order
- Euclidian distance from soma
- Euclidian length[*]
- final diameter
- initial diameter
- number of compartments[*]
- path distance from soma
- path length[*]
- segment contraction[*]
- segment taper

---

[*] added by calling the respective function

`@FORKLIST`

This is an array of hashes, each one containing one bifurcation's details. After initialisation, this contains the compartment number, branch order, and Euclidian and path distance from the soma; running morphometrics on bifurcations adds more elements to the hash.

- bifurcation torque angle
- compartment amplitude angle
- parent/child ratio
- daughter ratio
- power correction
- Rall's value
- segment amplitude angle
- tilt

(all these are added by calling the respective functions)