**checker.py**

**Main function:**

Checking file. If file contains rows with invalid data or inappropriate length (nativelly deduplicating data), number(s) of invalid rows are displayed and should be manually removed or rewritted to correct data type for further usage. Checking only: sufficient length of neuron (at least 3 points), items in row (14), correct data type of neuron (integer) and coordinates (float).

Otherwise displaying: number of lines, last neuron number, number of each neuron and it‘s number of deduplicated points

Dependent on yielder.py.

**User interface:**

Required arguments (**flag / alternative flag,** [argument type], (options)):

**-fcheck, --filename\_check** - [string] Name of checked file.

Example of usage:

>>> python checker.py **-fcheck** source\_file.csv

Expected (command line):

Document has 2897 lines and last neuron number is 53

Listing neurons and number of points...

Neuron number: 1 number of points (deduplicated): 55

Neuron number: 2 number of points (deduplicated): 44

Neuron number: 3 number of points (deduplicated): 62

Neuron number: 4 number of points (deduplicated): 37......

Failed:

Error(s) at line(s) 3 9 . Delete, or modify rows.

**yielder.py**

**Main function:**

Yielding and processing data for checker.py visualizer.py (only for single and burst!, not for range and group), comparer.py, and aligner.py. If used as main script normalizes data (for 0,0 graph root) and optionally filters point duplicities in every neuron data and write them into new csv file.

**User interface:**

Required arguments (**flag / alternative flag,** [argument type], (options)):

**-fin, / --filename\_input** - [string] (source\_file.csv) Data has to be presented in specified columns.

**-fout** **/ --filename\_output** ­ - [string] (output\_file.csv) Automatically created file with arbitrary name. Saves output data.

**-n / --count\_of\_neurons** -[integer] (Count of neuron) – Number of processed neurons.

Optional arguments (**flag / alternative flag,** [argument type], (options)):

**-dup, --allow\_duplicities** - [bool] (True/False) Allowing duplicite points in output file. If not set default is False – not allowed.

Examples of usage:

>>> python yielder.py **-fin** source\_file.csv **-fout** output\_file.csv **-n** 52 **-dup** True

>>> python yielder.py **-fs** source\_file.csv **-fo** output\_file.csv **-n** 40

**visualizer.py**

**Main function:**

Visualise data from provided csv file containing x and y coordinates in 2D space (second column – integer, fourth and fifth column - float) bounded to specific neuron characterised by number (second column – integer). Script is lazily yielding data from csv file, lazily filtering duplicite points (when neuron stopped growing and sequence of equal points was generated), harvesting all points of neuron and optionally normalising or semi-normalising data for 0,0 graph root if needed. There are several possibilities to plot processed data.

Dependent on yielder.py.

**User interface:**

**For plot data from single file**

Required arguments (**flag / alternative flag,** [argument type], (options)):

Example:

>>> python visualizer.py **-f** name\_of\_file.csv **-m** single | burst | range | group **-n** number(s) of neuron

**-f / --filename**  - [string] source\_file.csv - data has to be presented in specified columns.

**-n / --neuron\_first** -[integer] number(s) of neuron

**-m / --mode** - [string] ( single | burst | range | group )

– single – plot one neuron with base on 0,0 graph root, **-n**  has to be equal exactly one integer -> **-n** 8 plots neuron 8\*.

– burst – plot several neurons with 0,0 graph root sequentially, **-n** has to contain exactly two integers (first lower, second higher) -> **-n** 4 9 plots sequence of graphs of neuron 4, 5, 6, 7 and 8.

– range – plot range of choosen semi-normalized\*\* neurons into one graph -> **-n** 4 9 plots neurons 4, 5, 6, 7, 8 and 9 into single graph.

– group – plot random group of chosen semi-normalized neurons into single graph -> **-n** 7 8 2 1 25 – plots choosen neurons.

Optional arguments (**flag / alternative flag,** [argument type], (options)):

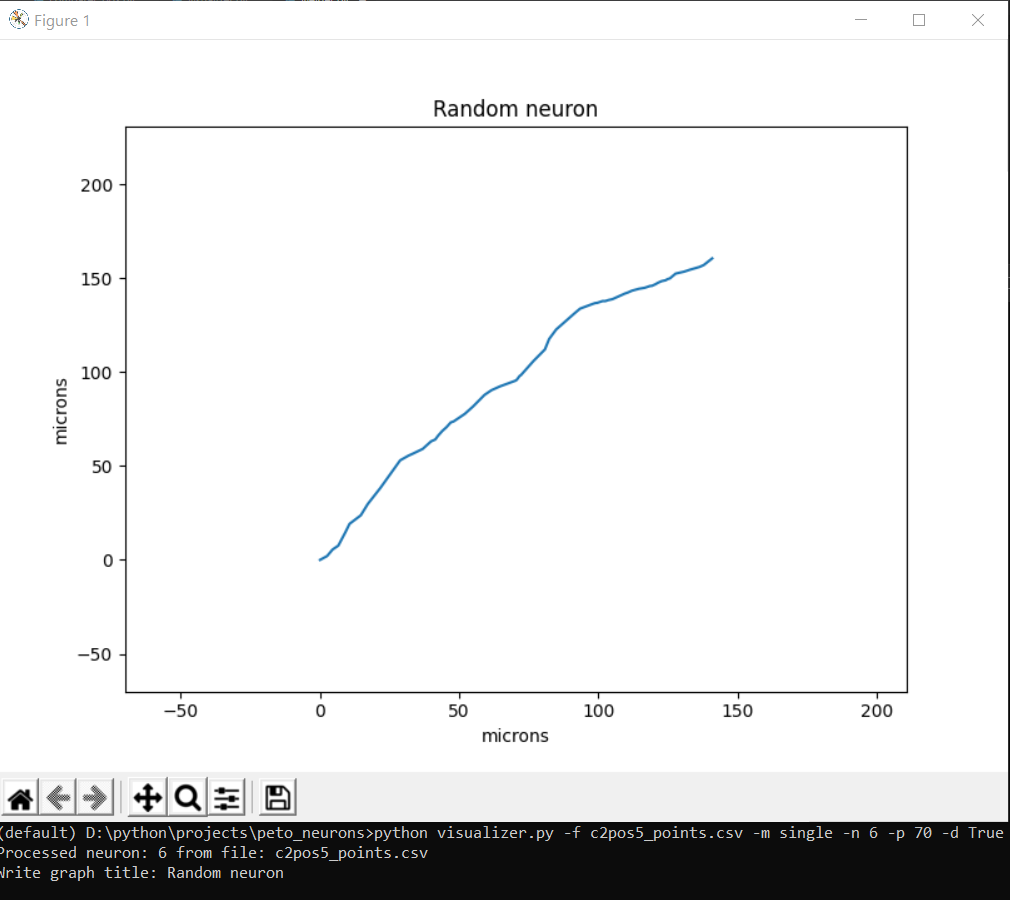
**-p / --padding** - [integer] padding around neuron, default = 20px.

**-d / --description** - [bool] Description of graph. If not specified in single and burst mode default = ‚Neuron + number of neuron‘. If custom description needed: **-d** True – description will be asked in command line. Modes range and group always requiring description by user input from command line (may be empty string by hitting enter) before graph plotting.

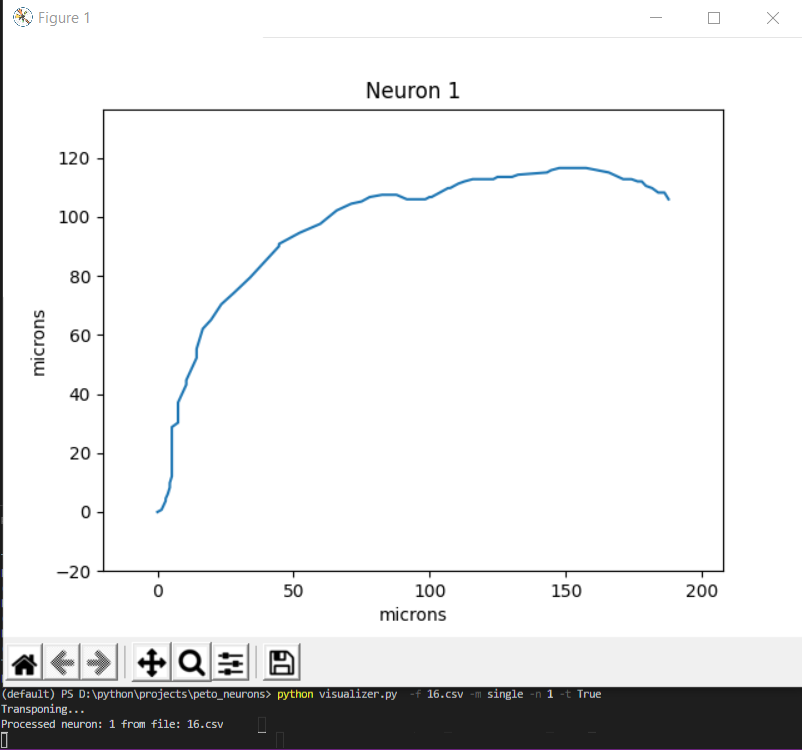
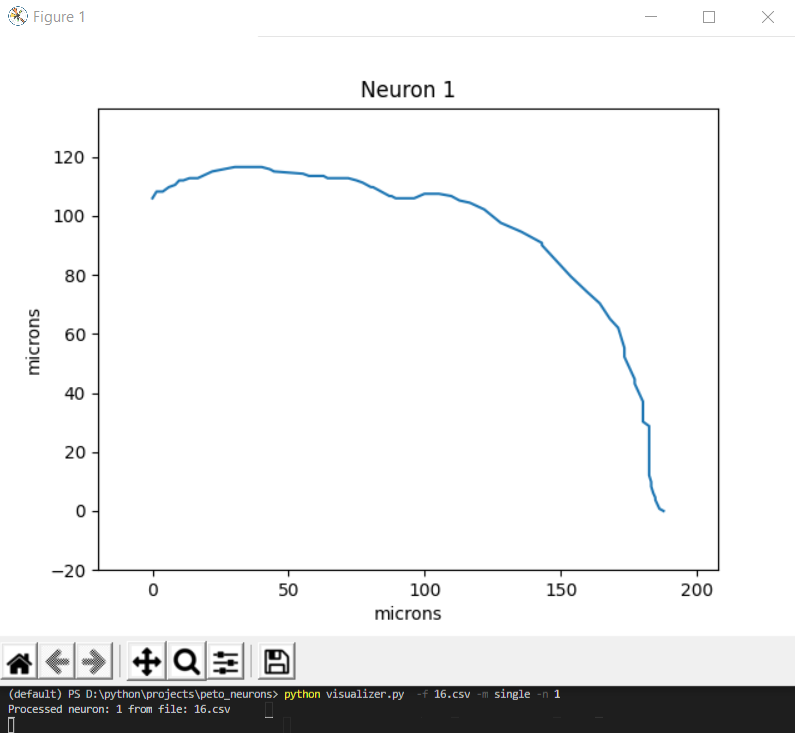
**-t / --transpone** - [bool] optionally transponing neuron for obtaining mirror horizontal orientation default = False.

Example:

>>> python visualizer.py **-f** name\_of\_file.csv **-m** single **-n** 6 **-p** 80 **-d** True



Example of transponation:



**For plot data from two files at once**

Required arguments (**flag,** [argument type]):

**-f / --filename**  - [string] source\_file.csv - data has to be presented in specified columns.

**-f2 / --filename2**  - [string] source\_file\_other.csv - data has to be presented in specified columns.

**-n / --neuron\_first** -[integer] number(s) of neuron

**-n2 / --neuron\_other** -[integer] number(s) of neuron for second file or group

**-m / --mode** - [string] ( range | group )

– range – plot range of choosen semi-normalized\* neurons -> **-n** 4 9 plots neurons 4, 5, 6, 7, 8 into first graph and **-n2** 2 3 plots 2 and 3 into secong graph. Both parameters has to be specified.

– group – plot group of chosen semi-normalized\* neurons into single graph -> **-n** 7 8 2 1 25 – plots choosen neurons into first graph and **-n2**4 5 9 8 2 1 5. Both parameters has to be specified.

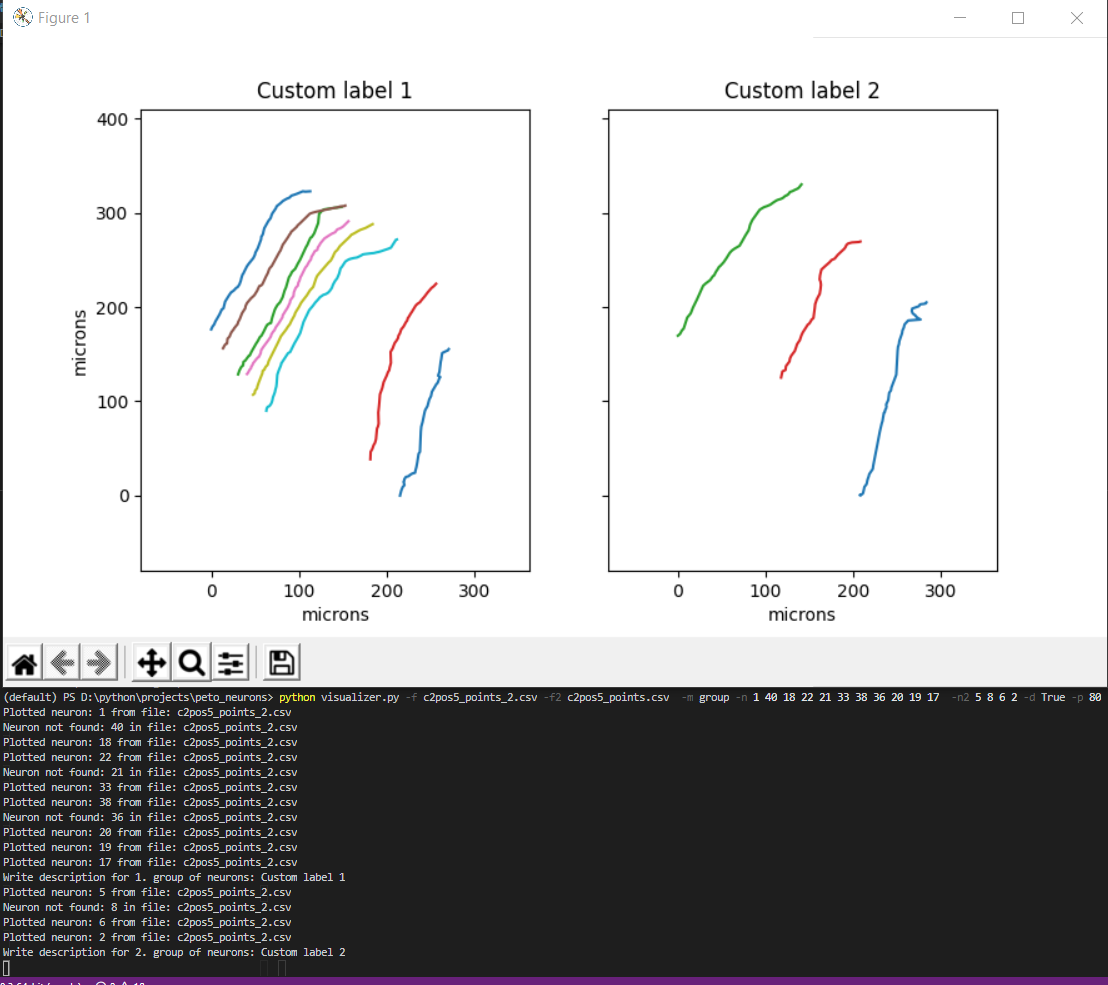
Optional arguments (**flag,** argument):

**-p / --padding** - [integer] padding around neuron(s), default = 20px.

**-d / --description** - [bool] Description of graph. If not specified default = ‚source\_file.csv‘. If needed: **-d** True. Description will be required by user input from command line (may be empty string) before graph plotting.

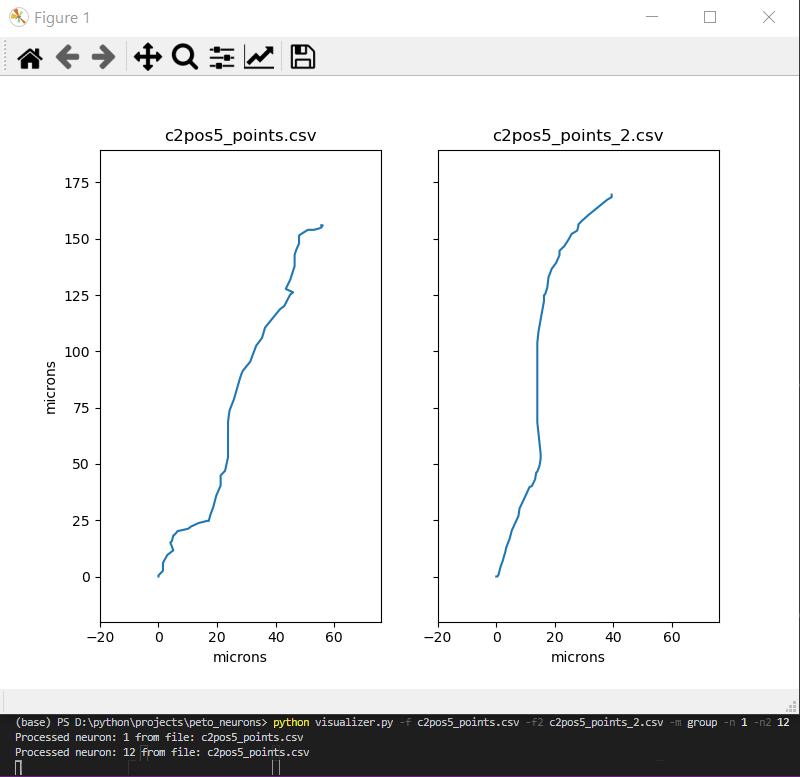
Example:

>>> python visualizer.py **-f** source\_file\_1.csv **-f2** source\_file\_2.csv **-m** group **-n** 1 40 18 22 21 33 38 36 20 13 17 **-p** 5 8 6 2 **-d** True **-p** 80



Example (single neuron comparation):

>>> python visualizer.py **-f** c2pos5\_points.csv **-f2** c2pos5\_points\_2.csv **-m** group **-n** 1 **-n2** 12



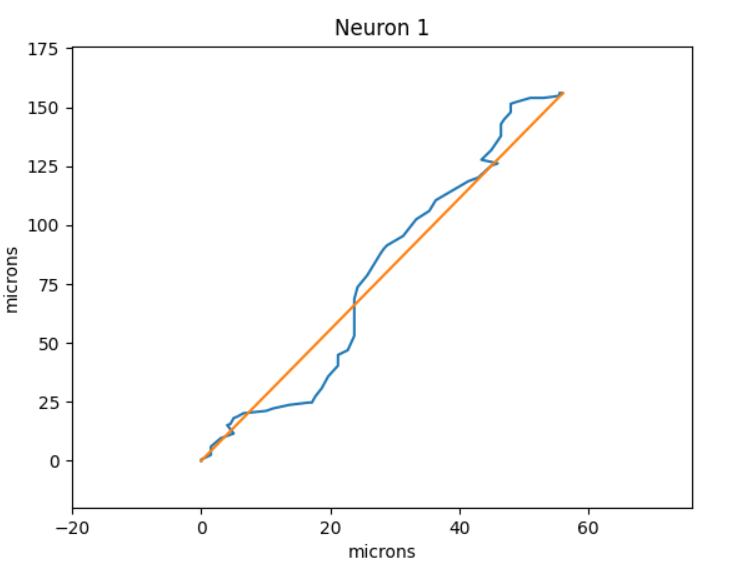
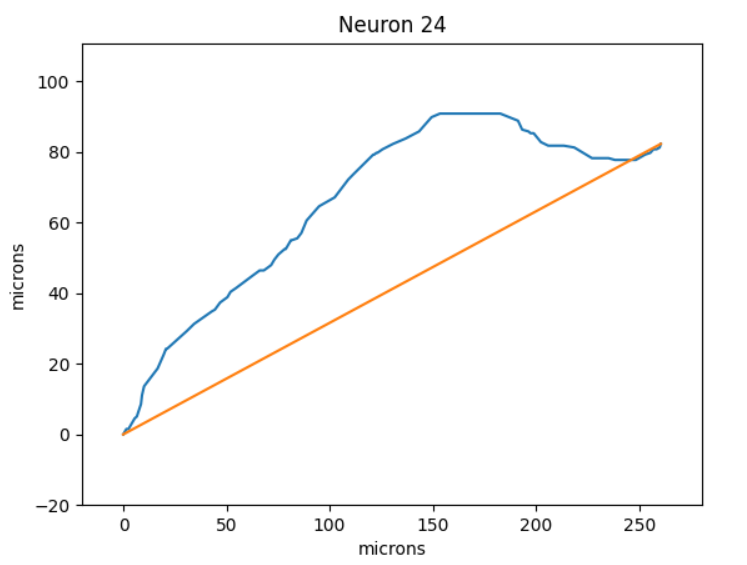
\* Script informs about found (Processed neuron: [integer]) or not found (Neuron not found: [integer]) neurons in file.

\*\* Whole group of neurons adjusted to 0,0 graph root

**comparer.py**

**Main function:**

Visualizing boxplots of deviations of neurons from their central axis (central axis is connecting lowest and highest point of neuron). Not applicable for shape assesment and comapration!

Required arguments (**flag,** [argument type]):

**-f / --filename**  - [string] source file. Data has to be presented in specified columns.

**-f2 / --filename2**  - [string] second source file. Data has to be presented in specified columns.

**-n / --neuron\_first** -[integer] number(s) of neuron

**-n2 / --neuron\_other** -[integer] number(s) of neuron

**-m / --mode** - [string] ( range | group )

– range – separate plot range of boxplot(s) from neuron deviation data -> **-n** 4 9 plots boxplot(s) 4, 5, 6, 7, 8 into first graph and **-n2** 2 3 plots 2 and 3 into second graph. Both parameters has to be specified.

– group – plot boxplot or group of boxplots from neuron deviation data -> **-n** 7 8 2 1 25 – plots choosen boxplots from neurons into first graph and **-n2**4 5 9 plots . Both parameters has to be specified.

**-o / --option** - [string] ( separate | pooled | averaged )

– separate – plot boxplot from each neuron list of deviations in provided group or range from mode

– pooled – plot one boxplot from pool of all deviations from neuron group or range

– averaged – plot one boxplot from each neuron average deviation from neuron group or range