MorphoPy: A python package for feature extraction of neural morphologies.

Sophie Laturnus[1,3,4], **Adam von Daranyi**[4], **Ziwei Huang**[1,3,4], **and Philipp Berens**[1,2,3,4]

1 Institute for Ophthalmic Research, University of Tübingen, Germany **2** Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany **3** Bernstein Center for Computational Neuroscience, University of Tübingen, Germany **4** Center for Integrative Neuroscience, University of Tübingen, Germany

DOI: 10.21105/joss.02339

Software

- Review
- Repository
- Archive

Editor: Olivia Guest

Reviewers:

- @Emptymalei
- @mstimberg

Submitted: 26 May 2020
Published: 03 August 2020

Summary

For a long time, the anatomy of a neuron has been considered a defining feature of neural cell types. However, computational analysis of neuronal morphologies persists to be a challenging problem. It usually begins with choosing a feature representation in order to make individual morphologies amenable to statistical data analysis or to processing by machine learning methods. Over the years, many different feature representations have been suggested in the literature, such as density maps (Jefferis et al., 2007), single valued summary statistics (morphometrics) (BBP, n.d.; Scorcioni, Polavaram, & Ascoli, 2008) or, more recently, persistence images (Kanari et al., 2018; Li, Wang, Ascoli, Mitra, & Wang, 2017). Unfortunately, current software packages for extracting them from morphological data are often focused on solely one such representation and implemented across various programming languages.

Our software package `MorphoPy` provides straightforward access to different feature representations from neural morphologies for downstream statistical analysis. It bundles common representations such as density maps, morphometrics, morphometric distributions and persistence images in one simple open source framework implemented in Python to make them accessible to a larger community. `MorphoPy` can be used either as a standalone command line tool or as a package within an interactive computing workflow.

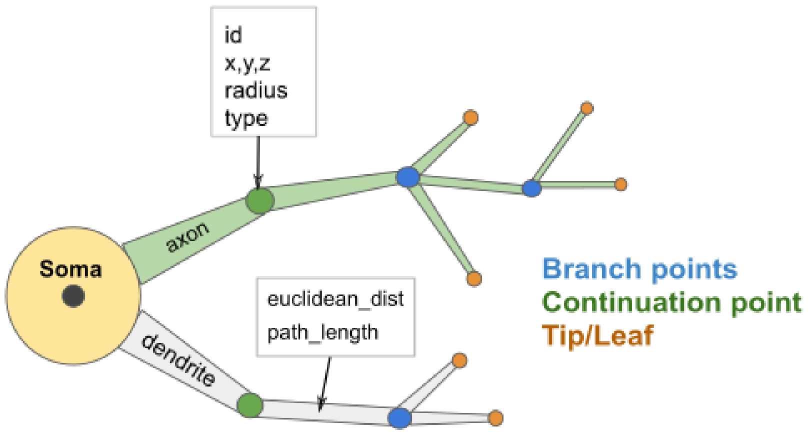

Figure 1: Neural reconstructions are represented as direct acyclic graphs with node and edge attributes.

MorphoPy builds on the functionality of the networkx package (Hagberg, Swart, & S Chult, 2008) and represents each neuron as a directed acyclic tree graph with node and edge attributes Figure 1. The package supports to read in files in the common swc-format, offers functions to compute various feature representations and provides 2D plotting routines for data exploration (Figure 2).

```
from morphopy.neurontree import NeuronTree as nt
from morphopy.computation import file_manager as fm

N = fm.load_swc_file("../data/EC3-80604.CNG.swc")
Dendrites = N.get_dendritic_tree()

from morphopy.neurontree.plotting import show_threeview
fig = plt.figure(figsize=(10,10))
show_threeview(N, fig)
```

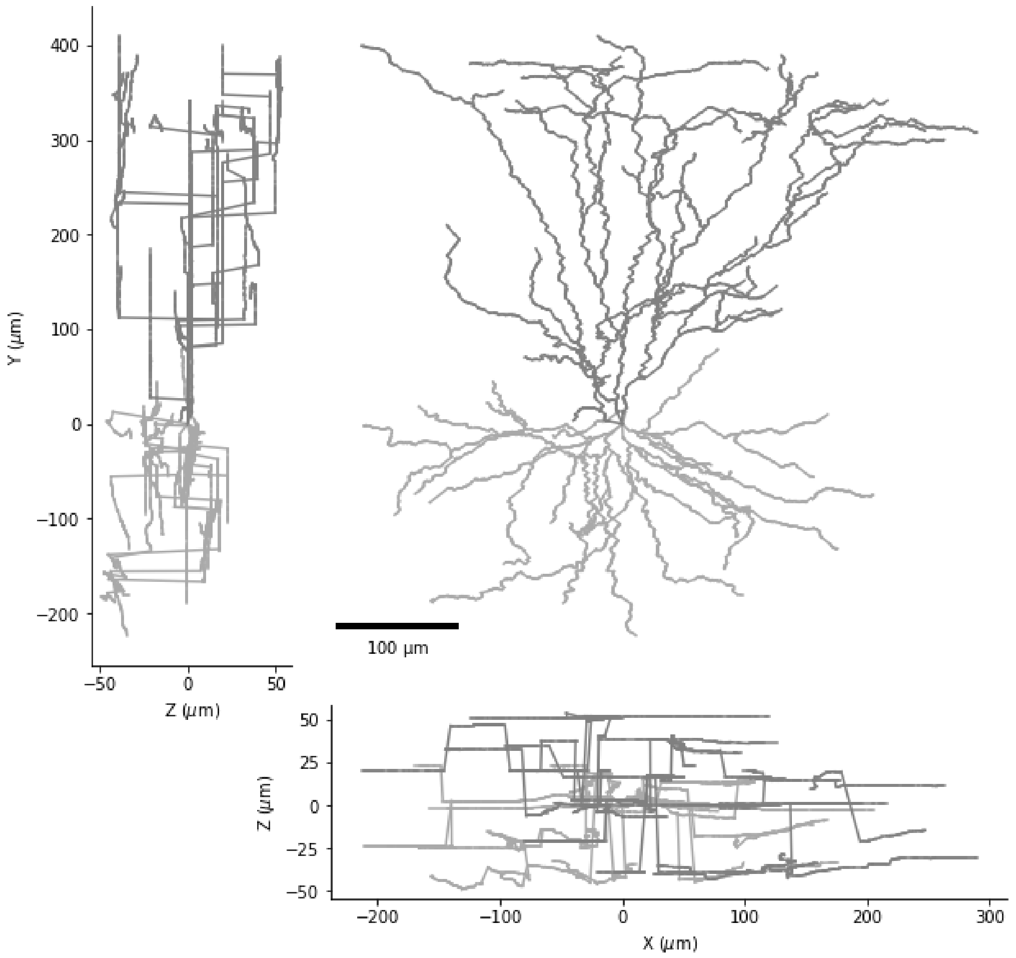

Figure 2: Plotting reconstructions in 2D.

As shown in the code snippet above, it is also possible to split the reconstruction into its different parts (axon or dendrites only) and operate on each neurite type separately.

Currently, MorphoPy supports the following feature representations:

Density maps are computed on the basis of a configuration file (or a dictionary) that controls parameters such as bin size and binning ranges. Additionally, users can specify whether and to which degree they want to smooth each density map (see Figure 3).

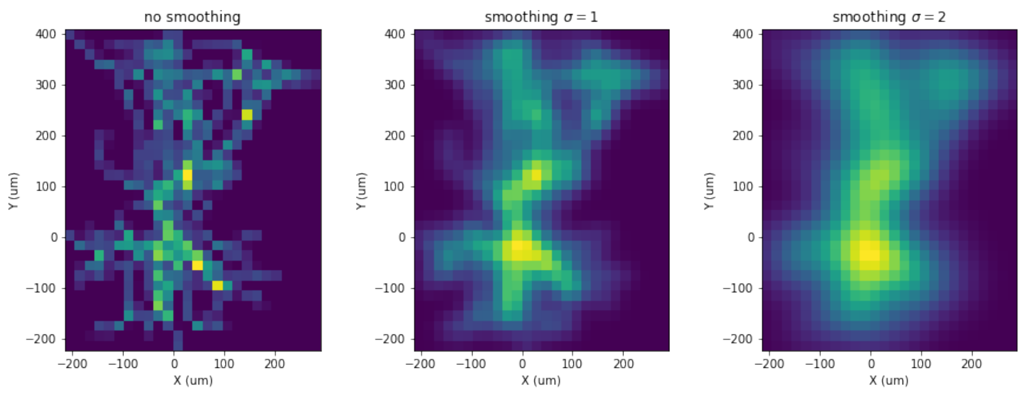

Figure 3: XY-density map of the dendrite plotted above with different degrees of Gaussian smoothing.

A variety of *morphometric statistics* can be computed on the nodes and edges of each reconstruction. The `get_morphometric_statistics()`-method offers a precompiled single valued selection of these statistics including e.g. min/max branch angles, maximal branch order, and maximal path length to the soma (see Figure 4), but in principle, they can be adjusted to the user's personal preference.

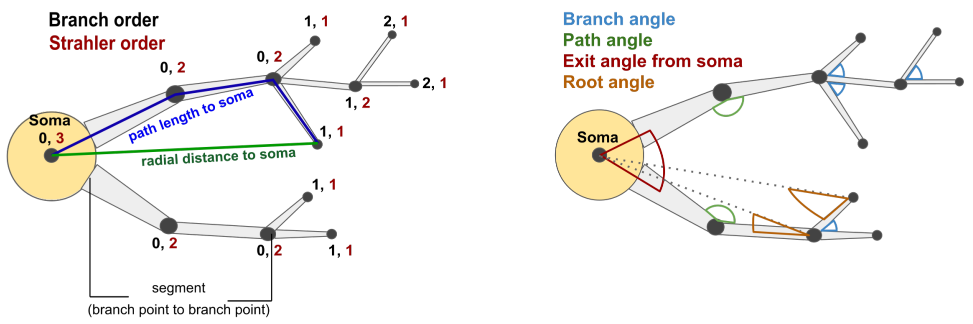

Figure 4: Node and edge related morphometric statistics.

Additionally, it is possible to query the entire *morphometric distribution* of each statistic either in form of a histogram or as a Gaussian kernel density estimate (kde). Figure 5, for example, shows the kde of radial distances, branch angles and their combination for the dendrites shown in Figure 2.

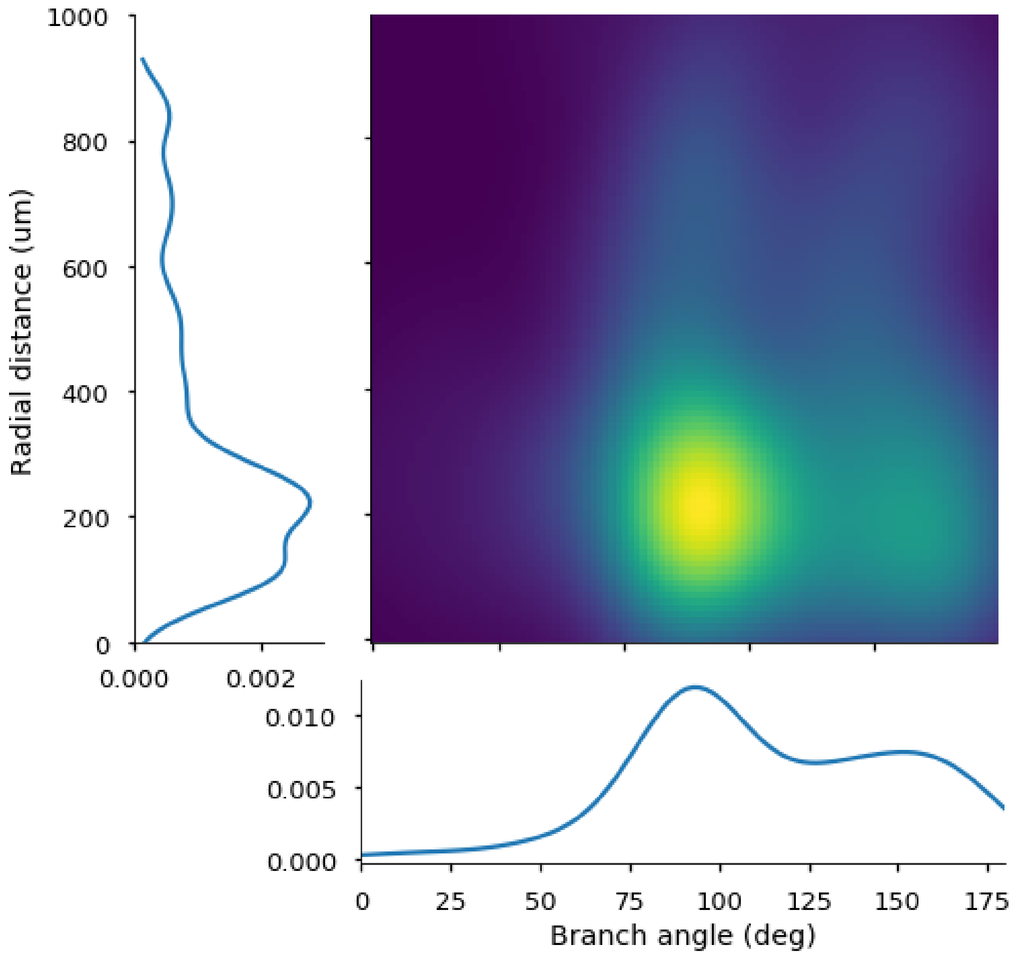

Figure 5: Kernel density estimate of branch angles as a function of the radial distance from the soma.

Furthermore, `MorphoPy` supports the generation of *2D persistence diagrams*. Persistence diagrams describe the branching of neural morphologies (Kanari et al., 2018; Li et al., 2017) with respect to a specified distance function. By default, `MorphoPy` computes a persistence diagram based on the radial distance from the soma, but users can choose from four different pre-implemented distance functions (radial distance, path length, height or branch order) or provide their own.

```
from morphopy.computation.feature_presentation import get_persistence

import numpy as np
def custom_distance(G, u, v):
    """
    Returns a distance between nodes u and v,
    which both are part of the graph given in G.
    """
    n = G.node[u]['pos']
    r = G.node[v]['pos']
    return np.dot(n, r)

df = get_persistence(Dendrites.get_topological_minor())
```

```
df_custom = get_persistence(Dendrites.get_topological_minor(), f=custom_distance)
```

In addition to working as a package in interactive computing environments, MorphoPy can be called from the command line to operate on single files or entire batches.

```
MorphoPy.py -c [density|persistence|stats]
            -i ['path_to_file'|'path_to_folder']
```

For a full documentation of MorphoPy's functionality please refer to our documentation and tutorial on our GitHub page.

MorphoPy has been developed in the context of a benchmarking study for cortical interneuron cell type classification based on their morphology (Laturnus, Kobak, & Berens, 2019). It has already been used in a series of scientific publications that relate transcriptome, electrophysiology and morphology of cortical interneurons in V1 and M1 (Scala et al., 2020, 2019).

Acknowledgements

This work was funded by the German Ministry of Education and Research(FKZ 01GQ1601), the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC2064/1 – 390727645; BE5601/4-1, SFB 1233 "Robust Vision", Project number 276693517), and the National Institute of Mental Health (U19MH114830).

References

BBP. (n.d.). NeuroM. *GitHub repository*. GitHub. Retrieved from https://github.com/BlueBrain/NeuroM

Hagberg, A., Swart, P., & S Chult, D. (2008). *Exploring network structure, dynamics, and function using NetworkX*. Los Alamos National Lab. (LANL), Los Alamos, NM, United States.

Jefferis, G. S., Potter, C. J., Chan, A. M., Marin, E. C., Rohlfing, T., Maurer Jr, C. R., & Luo, L. (2007). Comprehensive maps of drosophila higher olfactory centers: Spatially segregated fruit and pheromone representation. *Cell*, *128*(6), 1187–1203. doi:10.1016/j.cell.2007.01.040

Kanari, L., Dłotko, P., Scolamiero, M., Levi, R., Shillcock, J., Hess, K., & Markram, H. (2018). A topological representation of branching neuronal morphologies. *Neuroinformatics*, *16*(1), 3–13. doi:10.1007/s12021-017-9341-1

Laturnus, S., Kobak, D., & Berens, P. (2019). A systematic evaluation of neural morphology representations for cell type discrimination. *BioRxiv*, 591370.

Li, Y., Wang, D., Ascoli, G. A., Mitra, P., & Wang, Y. (2017). Metrics for comparing neuronal tree shapes based on persistent homology. *PLOS ONE*, *12*(8). doi:10.1371/journal.pone.0182184

Scala, F., Kobak, D., Bernabucci, M., Bernaerts, Y., Cadwell, C. R., Castro, J. R., Hartmanis, L., et al. (2020). Phenotypic variation within and across transcriptomic cell types in mouse motor cortex. *bioRxiv*. doi:10.1101/2020.02.03.929158

Scala, F., Kobak, D., Shan, S., Bernaerts, Y., Laturnus, S., Cadwell, C. R., Hartmanis, L., et al. (2019). Layer 4 of mouse neocortex differs in cell types and circuit organization between sensory areas. *Nature Communications*, *10*(1), 1–12. doi:10.1038/s41467-019-12058-z

Scorcioni, R., Polavaram, S., & Ascoli, G. A. (2008). L-measure: A web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature Protocols*, *3*(5), 866. doi:10.1038/nprot.2008.51