Practical Computing for Bioinformatics
Assignment #2:
Data analysis with Python

KU Leuven — November 24, 2022

Introduction

The roundworm *Caenorhabditis elegans* is widely used as a model-organism for studying embryo development. An attractive feature from the standpoint of the researcher, is that this animal has an invariant cell lineage. This implies that every cell can be uniquely identified, which allows researchers to compare individual cells across replicates. In this assignment you will use python to parse and analyse data from the *C. elegans* atlas (Cao *et al.*, 2020). The *C. elegans* atlas includes data across seventeen developing embryos, starting from the 4-cell stage, up to about 350 cells. For every embryo, the cell lineage tree was reconstructed, and cell volumes together with cell surface areas were tracked over time.

Open `PythonAssignment2022.py` in your favorite editor, and start by updating the function `my_name()` so that it returns your full name as a string.

> **Important:** As part of the evaluation, your code will be subjected to automatic tests. Please double check that your functions accept and return the correct number and type of variables. Do not change the names of the methods, or alter the input arguments.
>
> This is an individual assignment. Do not copy code or answers from other students.
>
> All the input data you need is included in the zip-file.
>
> Do not include any other packages than the ones already stated in the import section. If an import fails, you will need to install the package first. Use `conda install <package name>` if you are using Anaconda/Miniconda (recommended). Otherwise use `pip install <package name>`.

1 Querying the lineage file

Open the file `lineage_named_Sample04` with your favourite text/spreadsheet editor. This file contains the cell positions over time for a given embryo (sample04), and the tree-structure is encoded by the following fields:

- `self` : A unique ID for every cell
- `parent` : Links a cell to a cell from the previous time point
- `child1` : Links a cell to a cell in the next time point
- `child2` : Links a cell to a cell in the next time point (only used if a cell divides)

Note the cell naming convention: A new cell inherits the name of its parent with a suffix added. This suffix can either be a/p (anterior/posterior), or l/r (left/right). However, a few exceptions to this rule do exist (e.g. cell EMS has daughter cells E and MS).

If you understand the file structure, you are now ready to start with the first task. What we want to do is easily retrieve the parent cells for any given query cell.

a) Complete the method `build_datastructure_lineage_query`. This method reads in the lineage file and composes a python datastructure you can use to query the lineage tree. Think about the datastructure that is most appropriate for the task at hand (list ?, dictionary ?, set ?...).

b) Complete the method `get_parents_for_cell`. This method will return the parent cells for a given cell as a list, using the datastructure you just build. Mind the list order: it should go back in time, and not include the query cell itself.

Example output

```
>> lineage_file = Path(".data/lineage_named_Sample04")
>> ds_lineage_query = build_datastructure_lineage_query(lineage_file)
>> parents = get_parents_for_cell('ABalapaaa', ds_lineage_query)
>> parents
['ABalapaa', 'ABalapa', 'ABalap', 'ABala', 'ABal', 'ABa']
```

Note: This can be easily done using only native Python commands, but if you prefer already using the `pandas` package, that's also fine.

2 Volume and surface area

We will now work with the file `volumeAndSurface.csv` . Each row represents a volume and surface area measurement for a given cell at a given time point (expressed in minutes) for a given embryo (sample) (Figure 1). We will first explore the surface area to volume ratio ($SurfaceArea/Volume$) of the cells by making a simple plot. You can use the standard `matplotlib` visualization package, or you can turn to `seaborn`, which is built on top of `matplotlib`, and which will provide you with a bunch of nice graphs out-of-the-box.

a) Complete the method `plot_surf_vol_ratio_over_time`. This method plots the average surface area to volume ratio for every cell (Y-axis), with respect to the average time point at which a cell exists during development (X-axis). To illustrate, if cell EMS exists on average at t=2.25 minutes and has an average surface to volume ratio of 0.09, this should be visualized with one datapoint at position (2.25, 0.09). A regression line should be overlayed to highlight the trend over time.

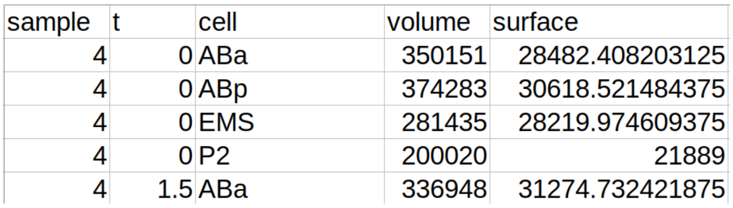
b) Save this figure to the `./output` folder as `surfToVolRatio.png`. Also include this figure in your report.

Question 1

What is the trend you see ? What explains this trend ?

3 Volume ratios

Now we will look at volume ratios. More specifically, the volume ratio of sister cells (daughter cells from the same parent cell). These are noteworthy because an asymmetric cell division can be an indicator of an



sample	t	cell	volume	surface
4	0	ABa	350151	28482.408203125
4	0	ABp	374283	30618.521484375
4	0	EMS	281435	28219.974609375
4	0	P2	200020	21889
4	1.5	ABa	336948	31274.732421875

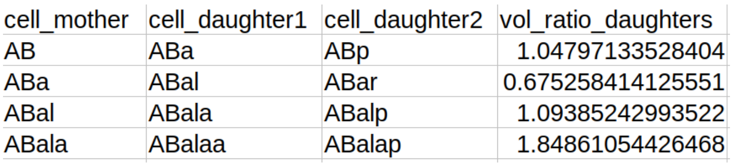
Figure 1: Layout of `VolumeAndSurface.csv`.

underlying polarization event in the mother cell. For example, the asymmetric distribution of PAR-proteins in the *C. elegans* zygote, will eventually lead to two daughter cells (AB and P0) with differing volumes.

Starting from the same dataset (`VolumeAndSurface.csv`), we need to do some parsing in order to detect all the cell divisions that occur across the samples. As noted above, the cell name of a given cell is composed of the parent cell name, plus a suffix. More specifically, two daughter cells can be labeled as 'anterior' and 'posterior', as in ABpl**a** and ABpl**p**, or they can be labeled as 'left' and 'right', as in Ea**l** and Ea**r**. Using this information we can pair up daughter cells. A few exceptions to this naming rule do exist (e.g. EMS with daughter cells E and MS), but you can ignore these cell divisions.

a) Complete the method `get_volume_ratios`. This method composes a dataframe containing the average volume ratios of sister cells (see Figure 2). Make sure that daughter1 always ends with 'a' or 'l'. Conversely, daughter2 should end with 'p' or 'r'. The volume ratio should be expressed as $Volume_{Daughter1}/Volume_{Daughter2}$.

b) Write the data as a csv-file to the `./output` folder as `VolumeRatios.csv`.



cell_mother	cell_daughter1	cell_daughter2	vol_ratio_daughters
AB	ABa	ABp	1.04797133528404
ABa	ABal	ABar	0.675258414125551
ABal	ABala	ABalp	1.09385242993522
ABala	ABalaa	ABalap	1.84861054426468

Figure 2: Layout of `VolumeRatios.csv`. Note: The ratios in this screenshot are not correct.

Note: Using a dedicated package for working with tabulated data (a.k.a dataframes) can make our lives a lot easier. The `pandas` package is a popular choice and supports all parsing functionalities you will probably ever need.

Tip: Getting comfortable with the more advanced data manipulation concepts such as **table joining** and **groupby** operations will save you lots of parsing headaches and time, not only for this assignment, but throughout your future career! If these concepts are not familiar to you, be sure to look online for some more background. There are numerous tutorials/videos out there. Afterwards, look in the pandas API for the equivalent operations and be on the lookout for other pandas methods that might come in handy.

Tip: Regular expressions (regex) are your friends.

Question 2

Plotting out the volume ratios, you should see something similar to Figure 3. What do you think causes the skew in distribution ?

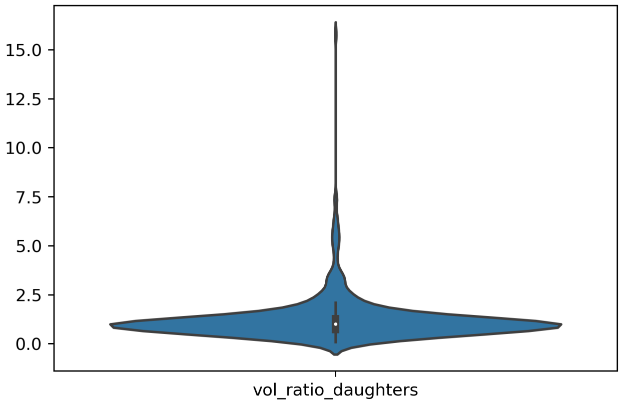


Figure 3: Violinplot of volume ratios of daughter cells

That's it. Happy Python programming !

> **Important:** Your final submission should be a `.zip` file containing two files:
>
> 1. Your edited `PythonAssignment2022.py`.
> 2. A text document containing the answers to the questions. (`.pdf` or `.docx`)

References

Cao, J., Guan, G., Ho, V. W. S., Wong, M. K., Chan, L. Y., Tang, C., Zhao, Z., and Yan, H. (2020). Establishment of a morphological atlas of the Caenorhabditis elegans embryo using deep-learning-based 4D segmentation. *Nature Communications*, **11**(1).