Introduction to the course Big Data analysis in Biomedical Research

2022_09_21

Prof. E. Araldi - University of Mainz

Learning outcomes

By the end of this course, you should be able to formulate an appropriate analysis plan for a research question in the biomedical field, select, gather and prepare data for analysis, and choose and apply appropriate statistical methods to the data.

You will learn:

- **Analytical methods** for processing, describing and visualizing complex biomedical data (graded midterm examination);
- **Terminology** for statistical tools and machine learning in biomedical discovery;
- **Project examples** where data analysis is used for meaningful results;
- Independently execute a **biomedical analysis project** (graded endterm exam);
- Understand the link between data and biomedical sciences: how do we **interpret** the findings (graded endterm exam)?

Tentative Syllabus

1) Introduction to Python:
    a. Intro to Python, Jupyter Lab;
    b. Python syntax, objects, models and methods, functions

2) Data preprocessing with Python:
    a. NumPy arrays and operations
    b. Intro to Pandas, dataframe manipulation, datetime

3) Introduction to statistics:
    a. Random variables and central limit theorem, distributions, t-test
    b. Inference, Confidence intervals, power calculations, association tests

4) Exploratory data analysis and visualization:
    a. Intro to Matplotlib and Seaborn
    b. Histograms, QQ-Plot, scatterplot, Correlations, Mann-Whitney-Wilcoxon Test

5) What can big data do for biomedical scientists:
    a. How to deal with big data, high performance computing
    b. How to define the problem, find datasets and plan the analysis

Homework 1x/week (if all done and turned in each week, +0.5 on final grade)

Tentative Syllabus

6) Linear models:
    a. Matrices, matrix algebra, linear models;
    b. Fitting linear models to data and testing.

7) Linear and logistic regression:
    a. Least square estimation, multiple linear regression;
    b. logistic regression.

8) Principal component analysis:
    a. Distance;
    b. Dimensionality reduction

9) Classification:
    a. Tree-based methods;
    b. K-nearest neighbors.

Midterm
(between the end of October and mid November)

Tentative Syllabus

10) Clustering:
    a. Hierarchical clustering;
    b. K-means
    c. Heatmaps

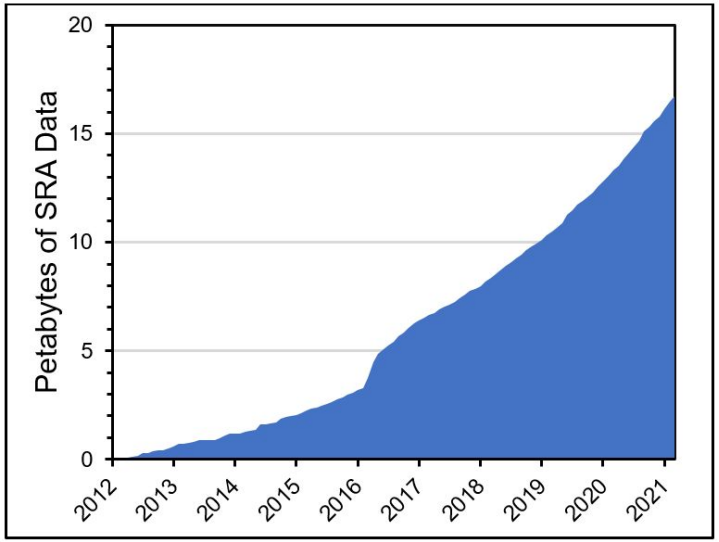
11) Predictive models:
    a. Support vector machine
    b. Random forest
    c. XGBoost model

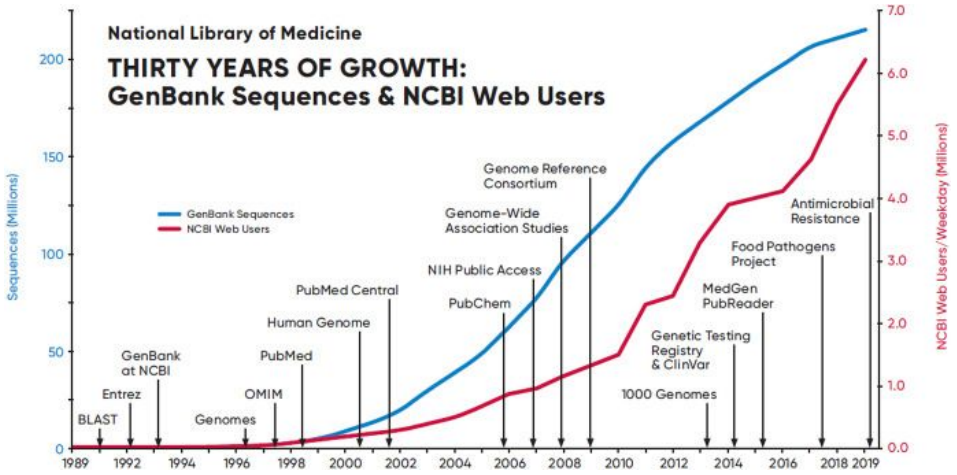
12) Concluding remarks

Endterm exam / final project
(turned in by the end of December)

From data analysis to meaningful biological and clinical insights

Technological advances = more data = more insights

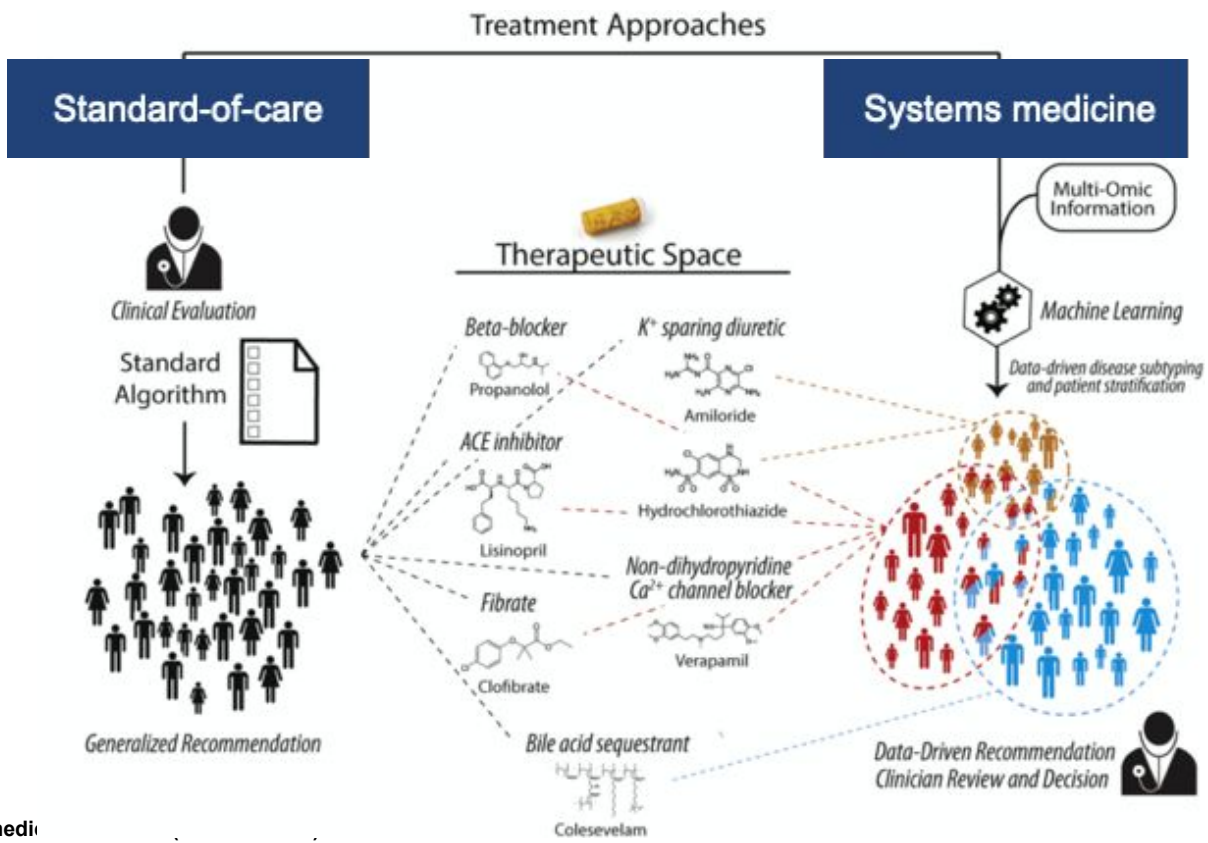




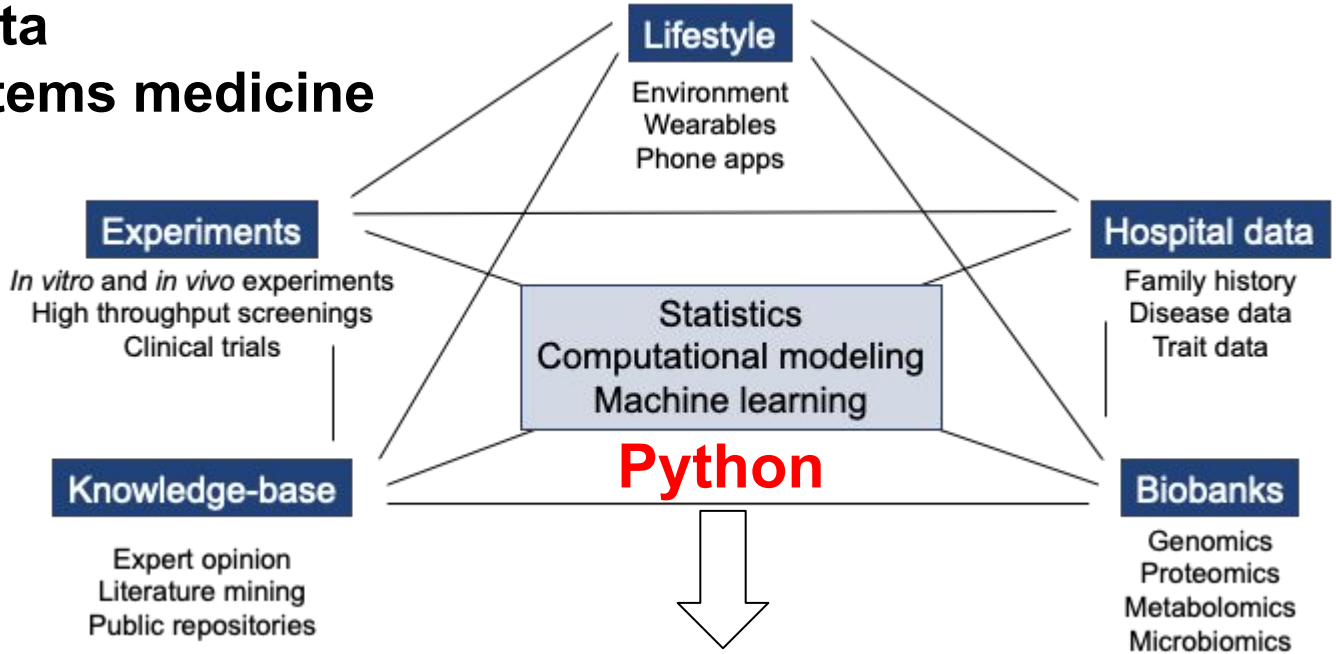
Problem: how to deal with all this data? **Data science (and this course!)**

https://ncbiinsights.ncbi.nlm.nih.gov/2021/08/09/espsss-workshop/

Why big data analysis: standard-of-care vs systems medicine



Big data in systems medicine



Greater understanding of onset, progression of disease

Understanding of molecular mechanisms

Disease prevention

Early diagnostic

Personalized treatments (more effective treatments, less side effects)

Introduction to Python

2022_09_21

Prof. E. Araldi - University of Mainz

What is Python?

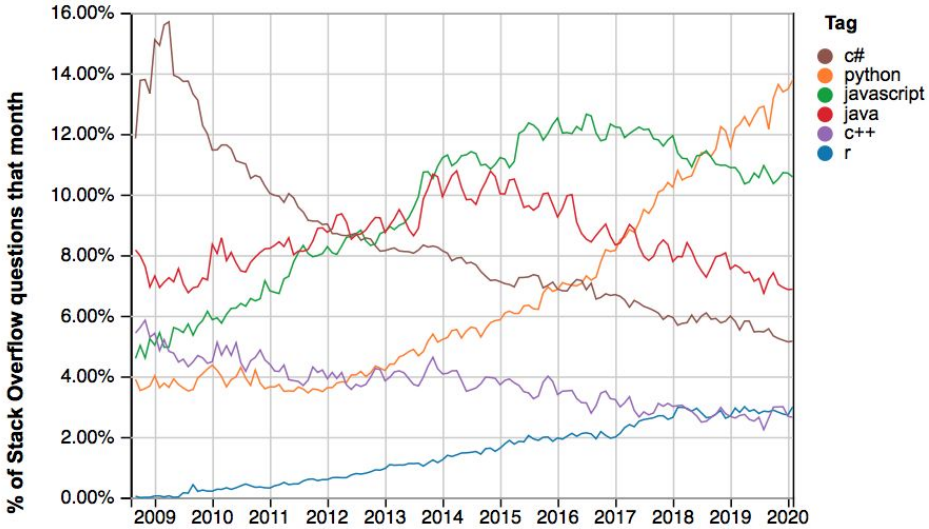
- Python is a high-level, interpreted, interactive and object-oriented programming language used for general-purpose software engineering.
- It was designed initially by Guido Van Rossum in 1991 and mainly developed for emphasis on code readability and easy implementation → Easy to learn!
- It is an open-source and one of the most popular programming language for data science (with a strong community of users in machine learning, artificial intelligence, data modeling, etc).

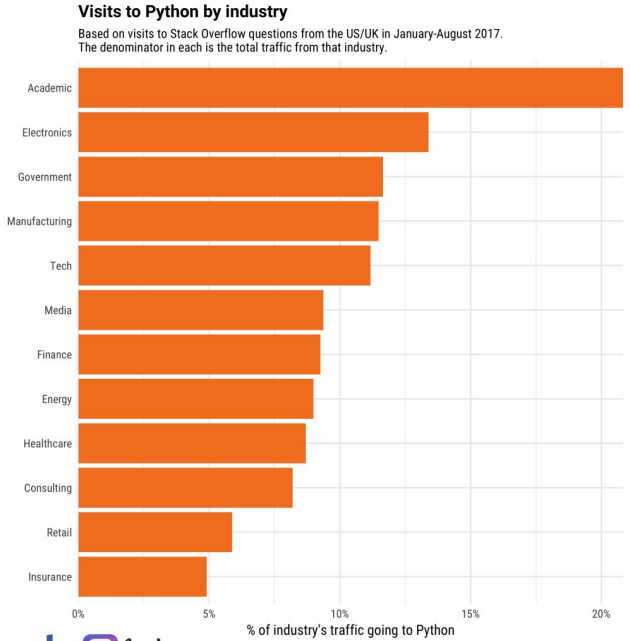
Shall you code with the strength of many men, Sir Knight!

Why is Python?

- **Versatile:**
  - Python is used in Data Mining, Data Science, AI, Machine Learning, Web Development, Web Frameworks, Embedded Systems, Graphic Design applications, Gaming, Network development, Product development, Rapid Application Development, Testing, Automation Scripting…
  - It offers great flexibility and has extensive open-source libraries for multiple applications: Pandas, numpy, scikit-learn, tensorflow, etc.
  - Useful in every industry (healthcare, financial services, marketing, education, etc)
- **Easy:** High readability, simple syntax. Easier to maintain and more efficiently-written alternative to languages that perform similar functionalities like C, R, and Java.
- **Community-based Support:** StackOverflow has answers to most of your questions already.
- **Scalable:** great for quick prototyping, and major projects
- **Fastest growing programming language:** it can keep up with technologies and needs

Python popularity vs other programming languages and by industry





Introduction to Jupyter Notebook

What is a Notebook

The Jupyter notebook is a web based interactive computational environment which provides a unique combination of code, shell environment and text. Three components:

The notebook web application: An interactive web application for writing and running code interactively and authoring notebook documents.

Kernels: Separate processes started by the notebook web application that runs user's code in a given language and returns output back to the notebook web application.

Notebook documents: Self-contained documents that contain a representation of all the contents visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.

Introduction to Jupyter Notebook

Different Cell Types?
Notebooks consist of a linear sequence of cells. There are four basic cell types:

Code cells: Input and output of live code that is run in the kernel

Markdown cells: Narrative text with embedded LaTeX equations

Heading cells: 6 levels of hierarchical organization and formatting

Raw cells: Unformatted text that is included, without modification, when notebooks are converted to different formats using nbconvert These cell types can be viewed by clicking Cell -> Cell Type in menu bar

When you need help:

https://stackoverflow.com/

Join the Slack Team of the class:
https://join.slack.com/t/slack-ymo7428/shared_invite/zt-1gg4u3nx2-Fc~4vupr9uzrosNMyk2O4A

Homework

Install Python 3.x and jupyter lab on your computer

Install Python 3.x and Jupyter Lab -- for Microsoft Windows

Python for Microsoft Windows

Download and install python 3.x for your version of Windows: www.python.org/download

Add Python to PATH (important!)

Check on Command Prompt if python is installed.

```
python
exit()
```

Check if PIP is installed (it should be).

```
pip help
```

Install Python 3.x and Jupyter Lab -- for MacOS X

Install python 3.x in MacOsX

In Terminal:

```
python --version
```

(it should be python 2.x.x)

www.python.org/download

Download and follow instructions to install python 3.x for your version of MacOs X

Check if installation was successful. In terminal:

```
python3
exit()
```

Check if PIP is installed (it should be).

```
pip help
```

Install Jupyter Lab

Install Jupyter Lab with pip

https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html

From Command Prompt or Terminal:

```
pip install jupyterlab
```

Install PIP, the Python Package Manager

Microsoft Windows
https://phoenixnap.com/kb/install-pip-windows

In Command Prompt, go to a directory where you can download/install things (e.g. create a python folder in desktop)

```
cd C:\Users\yourname\Desktop\python
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

Check if PIP is installed

```
pip help
```

MacOS X
https://pip.pypa.io/en/stable/installation/

In Terminal go to a directory where you can download/install things (e.g. create a folder in desktop)

```
cd ~/Desktop/folder_create
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

Check if PIP is installed

```
pip help
```

A few more housekeeping details

- Due to other not reschedulable commitments, class will be skipped some days (example, October 5th), but you will be informed in advance.

- Regardless, classes will end well before the end of the semester (probably beginning of December). In that time, we will be available to answer questions both during normal scheduled class days/times (upon appointment) or at other agreed upon times. This should give you enough time to work on the final project.

Evaluation

- I cannot forecast right now when the **midterm exam** will be, because we could be slower/faster this semester in covering some topics, but I will let you know well in advance (and choose between a couple of option)
- You will have also from Fri to Wed to complete it. In case you cannot do it in the assigned week, there is some flexibility to do it in the following weeks. It will be a guided biomedical analysis (coding) project with everything learnt until that point.
- The **final evaluation** is either a coding guided (= provided my me) biomedical analysis project OR your own data analysis project.
- The extra points for homework are for homework assigments, which you will receive on Fri and due on Wed (that does not include the exercises done in class). Also there there can be some flexibility (example, cutoff date is the week after if you do not have time that week, but that should not happen often). You can turn in the homework by submitting it on moodle.
- The midterm will be 40% of the grade, the endterm 60%, with an extra 0.5 points for homework (so theoretically you can total 6.5).