Introduction to Open & Reproducible Data Science (IORDS)

Michael Dayan, Foundation Campus Biotech Geneva (FCBG)

michael.dayan@fcbg.ch

2 ECTS

Open and reproducible science is an approach strongly emphasized to address the current reproducibility crisis revealed by the difficulty of reproducing published scientific experiments from fields ranging from machine learning to medicine. Initiatives have been conducted at all geographical scales to tackle this issue present globally: the European Commission published in December 2020 a report "Reproducibility of scientific results in the EU", the head of strategy of the Swiss National Science Foundation (SNSF) -- Katrin Milzow -- recently declared that "Open Science is a strategic priority in [their] multiannual program 2021-2024", the University of Geneva established in 2020 a roadmap until 2023 for Open Science and EPFL launched the Open Science Initiative in 2017 to promote open & reproducible research. The objective objective of **training on open science tools and practices** is emphasized in these initiatives and the two modules proposed below on data science and neuroimaging precisely aim at addressing it.

Two modules are proposed for participants as described below. **Module IORDS is independent of Module EDSAN, and, as it is self-contained, it can be attended without registering to Module EDSAN. Module EDSAN depends on knowledge of Module IORDS, but previous attendance to Module IORDS is not required.**

Module IORDS, "Introduction to Open & Reproducible Data Science", is aimed at students of all levels with a strong focus on computer science to train them in integrating the techniques that form the pillars of open & reproducible science. This module does not replace any computational course on any of the topic it features, and is meant instead as an overview of how the different computing pieces introduced fit together. As such, lectures on a given topic addressed during the course can be studied more in depth after the course to strive for mastery of that topic.

Module EDSAN, "Examples of Data Science Applications in Neuroimaging", illustrates the use of data science in neuroimaging which is a highly inter-disciplinary field pushing forward general computational tools: the well-known machine learning library scikit-learn in Python was developed largely by neuroimaging scientists, and so were Jupyter Notebooks and the most popular cloud solution to run them, Binder. As such, this module would benefit both students versed and not versed in neuroimaging and bioinformatics, as they would both discover applications in a highly dynamic field at the forefront of computational technologies, and learn how to collaborate within teams of varied expertise as they are typically found in research labs and the industry.

Detailed Curriculum in 2021 (provisional, subject to changes)

Module IORDS (Introduction to Open & Reproducible Data Science)

Lecture	Date & Time
Introduction	MON SEPT 20, 10AM-11AM
Linux Part 1	MON SEPT 27, 09AM-11 AM
Linux Part 2	MON OCT 04 , 09AM-11 AM
Linux Part 3	MON OCT 11 , 09AM-11 AM
GIT Part 1	MON OCT 18, 09AM-11 AM

GIT Part 2	MON OCT 25 , 09AM-11 AM
GIT Part 3	MON NOV 01 , 09AM-11 AM
Python Part 1	MON NOV 01 , 09AM-11 AM
Python Part 2	MON NOV 08 , 09AM-11 AM
Python Part 3	MON NOV 15 , 09AM-11 AM
Python Part 4	MON NOV 22 , 09AM-11 AM
Intro to Machine Learning Part 1	MON NOV 29 , 09AM-11 AM
Intro to Machine Learning Part 2	MON DEC 06 , 09AM-11 AM
Intro to Machine Learning Part 3	MON DEC 13 , 09AM-11 AM
Full example	MON DEC 20 , 09AM-11 AM

Syllabus

- Introduction
- Linux
    - Part 1
        - Linux filesystem
        - Bash terminal and commands
        - Commands help / manual
        - Navigating the filesystem
        - Parsing files
        - Piping commands
        - File permissions
        - Super user privileges
    - Part 2
        - Writing bash shell scripts
        - Using a dedicated developing environment
        - Executing scripts / permissions
        - Bash variables
        - Strings, arrays
        - Control flow statements (for loop, if-else statements)
    - Part 3
        - Control flow statements [continued] (while loop, case/switch)
        - Special script variables / exit codes
        - Arguments parsing with getopts
        - String/path manipulations
        - Hands on coding example of a full script

- Git
    - Git part 1: local repository
        - Git ecosystem
        - Git promotion model & typical workflow
        - Commit definition
        - Examining history
        - Hands-on: creating a history of commits on a local repo
    - Git part 2: branches & remote repositories
        - Concept of Branches (as a pointer to a commit)
        - Fast-forward merge
        - Introduction to Github
        - Remotes
        - Cloning, pushing and pulling
        - Tracking remote branches
        - Hands-on: creating a repo on Github and simulating team collaboration
    - Git part 3: advanced merge and contribution to Github projects

        - Three-way merge
        - Rebase
        - Cherry-picking
        - Tags
        - Merge conflicts
        - Hands-on: dealing with merge conflicts
        - Collaboration with fork / pull-request model
        - Hands-on: fork / PR model
- Python
    - Part 1:
        - Introduction to Jupyter notebooks
        - Markdown
        - Variables and datatypes
        - Mutable vs Immutable datatypes
        - Strings and string manipulation
        - Lists, Dictionaries and tuples
        - Control structures: for loop, if-else statements
        - Constant interactive hands-on in between each concept
    - Part 2:
        - Functions
        - None keyword
        - Function docstring
        - Type hinting
        - Development Environment (Visual Studio Code)
        - Modules and imports
        - Debugging
        - Exceptions and asserts statements
    - Part 3:
        - Object Oriented Programming Basics
        - Method attributes
        - Conda: creating, activating and configuring environments
        - Refactoring code
        - Creating modules and importing them in Jupyter notebooks
        - Numerical analysis and plotting with numpy: array creation, slicing and filtering
        - Plotting with matplotlib: line plot, scatter plot, bar plot, overlays, histograms, subplots, heatmaps
    - Part 4
        - Manipulating data with numpy and matplotlib
        - Reading files with python
        - Reading numerical files with numpy
        - File globbing and filtering
        - Example of dataset manipulation
        - Data analysis with pandas: dataset description, dealing with missing values, data filtering
        - Advanced visualization with seaborn
- Intro to Machine Learning (with Python)
    - Part 1:
        - General usefulness of ML in science
        - Example of ML applied to simple problem of predicting projectile range
        - Choice of features and labels
        - Assessing model performance, MSE
        - Basics of optimization, parameter estimation, model fitting
        - Model generalizability, training/testing
        - Hands-on example with sklearn using linear (OLS) model
        - Bias-variance trade-off
        - Cross-validation
        - Hands-on example with sklearn using polynomial features and Pipeline objects

        - Part 2:
            - Preventing overfitting with regularization
            - Hyper-parameter tuning
            - Nested cross-validation
            - Classification and classification metrics
            - SVM
            - Unbalanced classes
            - Dealing with regularization and class imbalance with SVM
            - Dealing with missing data / data imputation
            - Assessing model stability
            - Hyper parameter tuning with grid search
        - Part 3:
            - Unsupervised techniques
            - Dimensionality reduction with PCA
            - Clustering with K-means
            - Assessing clustering performance
    - Full project demonstration (including everything seen so far)
        - Setting up developing environment with conda
        - Exploring world happiness dataset with Jupyter Notebook
        - Create a model to predict happiness score from set of features
        - Create modular functions in an IDE, with docstring and type hints, and load them in the notebook
        - Create a package to install code developed so far
        - Implement continuous integration for automated code testing at each commit
        - Create Jupyter notebook demonstrating package use
    - Multiple choice question exam

Course location and additional information

The two modules are taught in English in the main amphitheater of Campus Biotech. Each module ends with a multiple-choice question exam.

The course will be available in hybrid mode, with attendance either physically in the auditorium of Campus Biotech in Geneva or remotely. Having/ bringing a computer/ laptop is required to attend/ connect to the interactive lectures!

Please keep informed of the directives of your university to know when physical attendance to courses becomes globally required again.

Registration

An institutional email address is preferred from course attendees; else please clearly state your affiliation in the application form! Registration is via the linked online form (https://tinyurl.com/iords2021**); registration will be done on a first come- first served basis, and closes on September 15.**