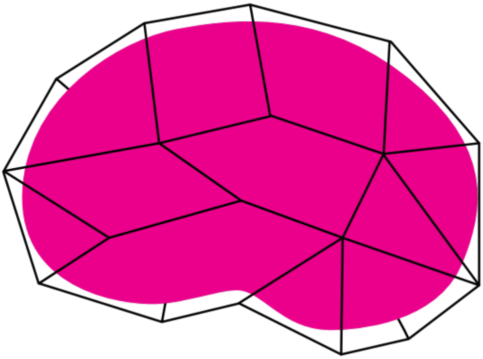
I  HACK

1st Brainhack Zurich - Open tools for reproducible neuroscience

2017, March 2&3

Neuroimaging · Neuroscience · Data Science · Visualization · Computing

Share ideas - Build Projects - Meet the community - Build tools

Room. We'll meet at the University Research Priority Program Dynamics of Healthy Aging at UZH, Andreasstrasse 15, 8050 Zurich, 4th floor.

Thursday: room AND4.57, start at 9:30
Friday: room AND4.06, start at 10:00

Laptop. If you'd like to participate in the tutorials, please bring a laptop. Regarding software to install for the tutorials, please consult the program which will be available at **dynage.github.io/brainhack-zh**.

As a quick preview: the tutorial on dynamic functional connectivity will require having a recent version of Matlab installed. The tutorial on the EEG toolbox requires an installation of Matlab and Automagic (**github.com/amirrezaw/automagic**). Other tutorials, like the Python tutorial or part of the visualization tutorial, will only require a modern web browser (must support web sockets; any recent version of Firefox or Google Chrome will do; Safari is not guaranteed to work).

For non-UZH members, WIFI will be available via *eduroam*.

Food. Tea and coffee will be provided. Meals are self-paid. We will organize meals, but we will also be happy to point you towards other dine-in or take-out options close by.

- We will order pizza for lunch on Thursday. If you'd like to join you can just sign up on Thursday morning.
- Thursday dinner (reservation required) at Giesserei (**diegiesserei.ch**).
- Friday lunch (reservation required) at Tibits (**tibits.ch**).

Community hour. On Friday, we will have a *Community hour*. That means we ask for your input in this *unconference session*. If you would like to give a brief *ignite talk* (a 5-minute talk, with or without slides, about a topic that excites you and that not enough people know about; or about a question you have for the community; or plans for an upcoming event you would like to share and get input). If you are interested in contributing to that, you will have the opportunity to sign up on location, prior to the session.

Thursday, March 2, 2017

9:30 - 10:15

Meet-and-greet and introduction to Brainhack

10:15 - 12:00

Tutorial

Riccardo Murri

An Introduction to Python for Scientific Programming

- recap of Python syntax for those coming from other programming languages
- scientific programming in Python: numpy, scipy, pandas
- worked out examples (if time allows)

I am going to assume that participants already know how to write a computer program (e.g. in MATLAB or R), though they might not know Python specifically.

We will work with IPython/Jupyter notebooks on a server that I will set up on Science Cloud.

So the only technical requirements are a modern web browser (must support web sockets; any recent version of Firefox or Google Chrome will do) and a working internet connection.

It would be great if I had them ready before the workshop -- but you know how these things go... But I promise I can make everything available for download immediately after :-)

12:00 - 12:30

Tutorial

Tyanko Aleksiev

Riccardo Murri

Science Cloud Introduction

13:30 - 14:30

Tutorial

Jessica Oschwald

Franz Liem

Software virtualization

Part 1: NeuroDebian - a computing platform for neuro- (and open) science

Learn about the community-driven computing platform for neuroscience: NeuroDebian. After this talk, you will know what NeuroDebian is and how it benefits you as a researcher. For example, you will hear how this tool can save your time, and offers a platform to contribute to open science and reproducibility. Finally, the talk will include a short demonstration of potential practical applications of NeuroDebian. The demonstration does not require you to follow along, but if you wish to do so, make sure that you have a working installation of NeuroDebian ready. Installation instructions can be found online **http://neuro.debian.net/index.html** (section *Get NeuroDebian*).

Part 2: BIDS Apps and BIDS Apps on the Science Cloud

This talk will demonstrate a new way to analyze neuroimaging data via the recently introduced BIDS Apps (bids-apps.neuroimaging.io). BIDS Apps provide software as container images (docker.com), and thereby enable reproducible analysis and eliminate the need to install a complex set of software tools. Standardizing the input data organization (Brain Imaging Data Structure BIDS, **http://bids.neuroimaging.io**) allows scientists to apply a wide variety of analysis pipelines on their data.

These Apps can easily be run on your local computer. Furthermore, they also can straightforwardly be run on UZH's Science Cloud (**http://www.s3it.uzh.ch/en/scienceit/infrastructure/sciencecloud.html**) via the bidswrapps tool (**http://github.com/fliem/bidswrapps)**.

Slides will be available at **https://github.com/DynAge/bhg17-zh-tutorials/tree/master/BIDS**

Target audience. This talk is aimed at neuroscientists at all levels of experience.

15:00 - 17:00

Tutorial

Nicolas Langer

Andreas Pedroni

Automagic EEG toolbox

In this session we will introduce two newly developed EEG toolboxes from our lab: *Automagic* and the *Microstates toolbox for EEGLAB*. First, we will present the theoretical background for EEG preprocessing and Microstates analysis. Afterwards, a step-by-step tutorial will guide you through both toolboxes (please have MATLAB installed). Both are freely available on github, can be used stand-alone, but also will be integrated as EEGLAB plugins.

Automagic is a toolbox for preprocessing of EEG-datasets. In a first step, it automagically removes artifacts (e.g. eye movements, noisy electrodes, and other dark magic) from raw EEG-data. In a second step, *Automagic* enables visual inspection and an efficient manual removal of remaining artifacts. Furthermore, you can rate the quality of individual EEG-files. *Automagic* supports various EEG data formats. It's just magic!

With the *Microstates toolbox for EEGLAB* (some call it *Micromagic*) the user can segment EEG data into Microstates with state of the art clustering methods. It furthermore offers functions for backfitting and calculates common Microstates parameters that can be submitted to statistical analyses. It's magic!

Friday, March 3, 2017

10:00 - 11:30

Tutorial

Ronald Sladky

Franz Liem

Visualization

Part 1: sweetView: a simple, quick, and powerful viewer for MRI images and SPM results

We all love SPM. However, creating really *nice* figures for your talk or publication typically involves quite a lot of manual labor and post processing. And then, just when you are done and send the figures to your co-authors, you are informed that you need to exclude one subject from group analysis because their drug screening was positive (or negative – depending on the study).

Things like this happen and this was my motivation to create sweetView a simple and powerful viewer for MRI images and SPM results that allows you to quickly create triplanar or mosaic overlays of your SPM results. The core features include fast selection of images, easily customizable overlays for masks or SPMs, adding (anatomical) labels and saving the slice selection and multiple cursor positions, so you can easily reproduce your original figure design.

The software has been designed for people who use Matlab and SPM12. After my brief introduction demo, I would like to know if you find it useful and what features you would like me to implement in a future version of sweetView that I have not yet thought about.

Part 2: Plotting brain images with Nilearn

Creating result images for an MRI study often is done manually. Automating this process (for instance when displaying images for each subject of your sample, or after re-running analyses in the publication process) can save time and trouble and makes your analysis reproducible. Nilearn (**http://nilearn.github.io**) is a python package for running machine learning algorithms on neuroimaging data. Here we will take a brief look at Nilearn's abilities to visualize MRI images to create publication-ready images of brains.

Material. The tutorial will be provided as IPython notebooks which can be found here: **http://github.com/fliem/nilearn_tutorials**. They can be run locally (which requires installing IPython and Nilearn - I recommend doing this via the anaconda distribution **http://www.continuum.io/downloads**) or without installing anything in your browser (**http://github.com/fliem/nilearn_tutorials**).

Target audience. This demonstration is aimed at people interested in creating publication-ready images of brains in an automated fashion. Knowledge of python is helpful but not required needed.

11:30 - 12:00

Ignitie talk

Claudio Tessone

Networks

13:00 - 15:00

Tutorial

Michael Notter

Nipype Tutorial – How to analyze your MRI data in an easy and flexible way

This tutorial is for people who want to learn about Nipype, an open-source Python based neuroimaging software that provides an uniform interface to existing neuroimaging toolboxes, such as SPM, FSL, FreeSurfer, AFNI and many more. Nipype facilitates the interaction between these softwares and allows the user to create workflows tailored to the needs of the analysis at hand. This tutorial will cover the basics of Nipype and hopefully convince you to use Nipype for all of your future MRI projects.

My tutorial will also be hands-on, but participants don't need to install anything for it.

15:30 - 17:30

Tutorial

Thomas Bolton

Youngs Farouk

Anjali Tarun

Dynamic functional connectivity for dummies

Brief description: When analysing fMRI data, moment-to-moment fluctuations in the interaction between brain regions are relevant pieces of information that can be captured by various analytical approaches. Here, we will first provide a broad theoretical introduction on the dynamic functional connectivity methodological landscape. We will then focus on two analytical pipelines: (1) the generation of eigenconnectivity building blocks from a sliding window analysis, and (2) the generation of innovation-driven co-activation patterns following total activation. Those two methodologies will be exemplified by hands-on work applied, with different parameter settings, on resting-state data from the Human Connectome Project.

To download or install beforehand: The only requirement for the attendants will be a working version of Matlab. Scripts will be tested on Matlab 2014b and on more recent versions, but should in principle also work for earlier versions.

Material. The material required by the attendants for the tutorial will include:

- fMRI data from a subset of subjects sampled from the Human Connectome Project initiative
- Matlab scripts enabling to run dynamic functional connectivity routines on this data
- Slides presented in the introduction part of the tutorial
- Attendants are encouraged, but not obliged, to take a look at this material beforehand.

For reference, links to scientific articles describing the covered techniques are:

- Dynamic functional connectivity review (on which the introduction is based):

http://www.sciencedirect.com/science/article/pii/S1053811916307881

- Eigenconnectivities: **http://www.sciencedirect.com/science/article/pii/S105381191300774X**
- Total activation: **http://www.sciencedirect.com/science/article/pii/S1053811913001146**
- Innovation-driven co-activation patterns: **http://www.nature.com/article-assets/npg/ncomms/2015/150716/ncomms8751/extref/ncomms8751-s1.pdf**
- For more details about the introduced techniques, attendants are also pointed at our website: **http://miplab.epfl.ch/index.php/software**

BRAIN HACK ZURICH 2017

organized by

Amelie Haugg, Franz Liem, Jessica Oschwald,
Frank Scharnowski, Ronald Sladky, and Vivian Steiger