Mon, 2020-03-30

to do:

- ☒ start a daily log
- ☒ download git repository on Uppmax
- ☒ do some organizatorial stuff:
  - ☒ where are the data?
  - ☒ how it looks like?
  - ☒ create directory structure for the project
  - ☒ link raw data into my "raw_ext" folder
- ☒ make the project plan
- ☒ save daily log and project plan into "notebooks"
- ☒ make the diagram for work flow
- ☒ push changes to git

- log in to Uppmax: `ssh -AX username@rackham.uppmax.uu.se`
- project number: **g2020008**
- where to run analyses: **/home/username/**
- where is the raw data: **/proj/g2020008/nobackup/private/1__Zhang__2017/**

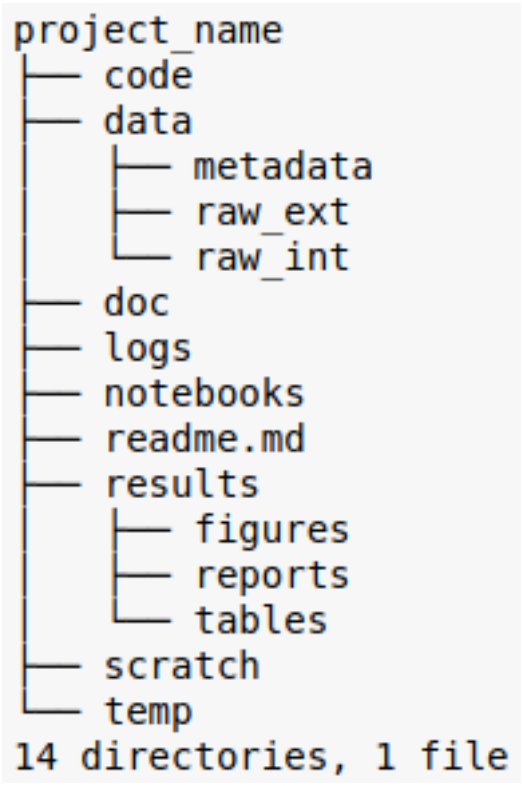
```
sudo apt-get install tree
```
the directory tree: tree

```
.
├── genomics_data
│   ├── Illumina
│   │   ├── E745-1.L500_SZAXPI015146-56_1_clean.fq.gz
│   │   └── E745-1.L500_SZAXPI015146-56_2_clean.fq.gz
│   ├── Nanopore
│   │   └── E745_all.fasta.gz
│   └── PacBio
│   ├── m131023_233432_42174_c100519312550000001823081209281335_s1_X0.1.subreads.fastq.gz
│   ├── m131023_233432_42174_c100519312550000001823081209281335_s1_X0.2.subreads.fastq.gz
│   ├── m131023_233432_42174_c100519312550000001823081209281335_s1_X0.3.subreads.fastq.gz
│   ├── m131024_200535_42174_c100563672550000001823084212221342_s1_p0.1.subreads.fastq.gz
│   ├── m131024_200535_42174_c100563672550000001823084212221342_s1_p0.2.subreads.fastq.gz
│   └── m131024_200535_42174_c100563672550000001823084212221342_s1_p0.3.subreads.fastq.gz
└── transcriptomics_data
    ├── RNA-Seq_BH
    │   ├── trim_paired_ERR1797972_pass_1.fastq.gz
    │   ├── trim_paired_ERR1797972_pass_2.fastq.gz
    │   ├── trim_paired_ERR1797973_pass_1.fastq.gz
    │   ├── trim_paired_ERR1797973_pass_2.fastq.gz
    │   ├── trim_paired_ERR1797974_pass_1.fastq.gz
    │   ├── trim_paired_ERR1797974_pass_2.fastq.gz
    │   ├── trim_single_ERR1797972_pass_1.fastq.gz
    │   ├── trim_single_ERR1797972_pass_2.fastq.gz
    │   ├── trim_single_ERR1797973_pass_1.fastq.gz
    │   ├── trim_single_ERR1797973_pass_2.fastq.gz
    │   ├── trim_single_ERR1797974_pass_1.fastq.gz
    │   └── trim_single_ERR1797974_pass_2.fastq.gz
    ├── RNA-Seq_Serum
```

```
├── RNA-Seq_Serum
│   ├── trim_paired_ERR1797969_pass_1.fastq.gz
│   ├── trim_paired_ERR1797969_pass_2.fastq.gz
│   ├── trim_paired_ERR1797970_pass_1.fastq.gz
│   ├── trim_paired_ERR1797970_pass_2.fastq.gz
│   ├── trim_paired_ERR1797971_pass_1.fastq.gz
│   ├── trim_paired_ERR1797971_pass_2.fastq.gz
│   ├── trim_single_ERR1797969_pass_1.fastq.gz
│   ├── trim_single_ERR1797969_pass_2.fastq.gz
│   ├── trim_single_ERR1797970_pass_1.fastq.gz
│   ├── trim_single_ERR1797970_pass_2.fastq.gz
│   ├── trim_single_ERR1797971_pass_1.fastq.gz
│   ├── trim_single_ERR1797971_pass_2.fastq.gz
│   └── untrimmed
│   ├── ERR1797969_pass_1.fastq.gz
│   ├── ERR1797969_pass_2.fastq.gz
│   ├── ERR1797970_pass_1.fastq.gz
│   ├── ERR1797970_pass_2.fastq.gz
│   ├── ERR1797971_pass_1.fastq.gz
│   └── ERR1797971_pass_2.fastq.gz
├── Tn-Seq_BHI
│   ├── trim_ERR1801012_pass.fastq.gz
│   ├── trim_ERR1801013_pass.fastq.gz
│   └── trim_ERR1801014_pass.fastq.gz
├── Tn-Seq_HSerum
│   ├── trim_ERR1801009_pass.fastq.gz
│   ├── trim_ERR1801010_pass.fastq.gz
│   └── trim_ERR1801011_pass.fastq.gz
└── Tn-Seq_Serum
├── trim_ERR1801006_pass.fastq.gz
├── trim_ERR1801007_pass.fastq.gz
└── trim_ERR1801008_pass.fastq.gz
11 directories, 48 files
```

- connect to github to save the **code**, **results** and **notebook** folders.
  - copy repository to local machine: `git clone URL_of_your_repo`

- create a directory tree in my "/home/username/git_repo/" folder using a custom script:
  - write custom script: **0directory_tree.sh**
  - copy script to uppmax (you need to be *not logged in* Uppmax): `scp address/local_file user@rackham.uppmax.uu.se:/home/usernameRackham/`
  - make script executable: `chmod u+x 0directory_tree.sh`
  - run script like this: `./0directory_tree.sh path_to_where_you_want_the_new_folder project_name`



- copy script into the correct folder:
  - make a new folder in **code**, named **0_organization**: `mkdir 0_organization`
  - paste file there and add a README
- create a soft link from the **raw_ext** folder in my directory tree to the raw data in the project:
  ln -s {source-filename} {symbolic-filename}
  `ln -s /proj/g2020008/nobackup/private/1_Zhang_2017/ link_to_raw_data`

Wed, 2020-04-01

to do:

☒ figure out how to correctly export jupyter to pdf (the images are not showing and the font is wrong)

☒ correct README.md showing directory tree

☒ change the diagram for the project plan to include received comments:
  ☒ "You don't need to run QC on the PacBio reads, neither trimmomatic or FastQC are very good at handle long reads so you can skip that step."

- ☒ "For the Spades assembly you can input both long and short reads if you would like."
- ☒ "For the annotation part, it is enough to run Prokka, Maker2 is designed to run on eukaryotic genomes."
- ☒ "Also, please add a section on how you are going to organise scripts, data from analysis and so forth."

☒ push changes to git

Task 1

Tried and failed:

- export from jupyter notebook depends on 'nbconvert' settings. It needs to have installed:
  - 'pandoc' (`sudo apt-get install pandoc`)
  - 'XeLaTex' (`sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-generic-recommended`)
  - the output is bad, no image included
- export as html then convert to pdf with wkhtmltopdf (`sudo apt-get install wkhtmltopdf`)
  - didn't get it to work
- same but with pandoc
  - `pandoc test.html -t latex -o test.pdf`- works better (images are there and structure is preserved) but the layout need improving.
  - but directly from .ipynb to .pdf it doesn't work.
  - "Pandoc first converts the Markdown file to LaTeX which then gets compiled to PDF"-so the issue might be with LaTex.
- use 'nbconvert' directly: **jupyter nbconvert --to output_format input_notebook**
  `jupyter nbconvert --to markdown wiki_daily_log.ipynb`
- installed pandoc2. nothing works

Will just export .ipybn to .md and upload it like that to git.

- `pandoc wiki_daily_log.ipynb -o wiki_daily_log_pandoc.md`
  As pdf:

- `pandoc wiki_daily_log.ipynb -t latex -o test_pandoc.pdf`

Pandoc options:
`--resource-path` defines the path where Pandoc will look for resources that are linked in the notebook. This allows us to discover images etc that are in a different folder from where we are invocing pandoc.
`--extract-media` is a path where images and other media will be extracted at conversion time. Any links to images etc should point to files at this path in the output format.
`-s` (or `--standalone`) tells Pandoc that the output should be a "standalone" format. This does different things depending on the output, such as adding a header if converting to HTML.
`-o` the output file, and implicitly the output file type (e.g., markdown)
`-t` the type of output file if we want to override the default (e.g., GitHub-flavored markdown vs. Pandoc markdown)