QuantMS nextflow workflow (nf-core compatible)

14th May 2025

Henry Webel

Does broccoli boost bad gut bacteria?

- One strain of *E. coli* was analyzed using MS-based proteomics

- Shoutout to Caroline Jachmann for creating a peptide atlas for *E. coli* and sharing information on small scale experiments

INFLAMMATION

NEWS | INFLAMMATION | MICROBIOME IN HEALTH AND DISEASE

Does broccoli boost bad gut bacteria?

22 August 2023

By Rob Clancy, staff writer. Reviewed by Dr Emily Gulliver

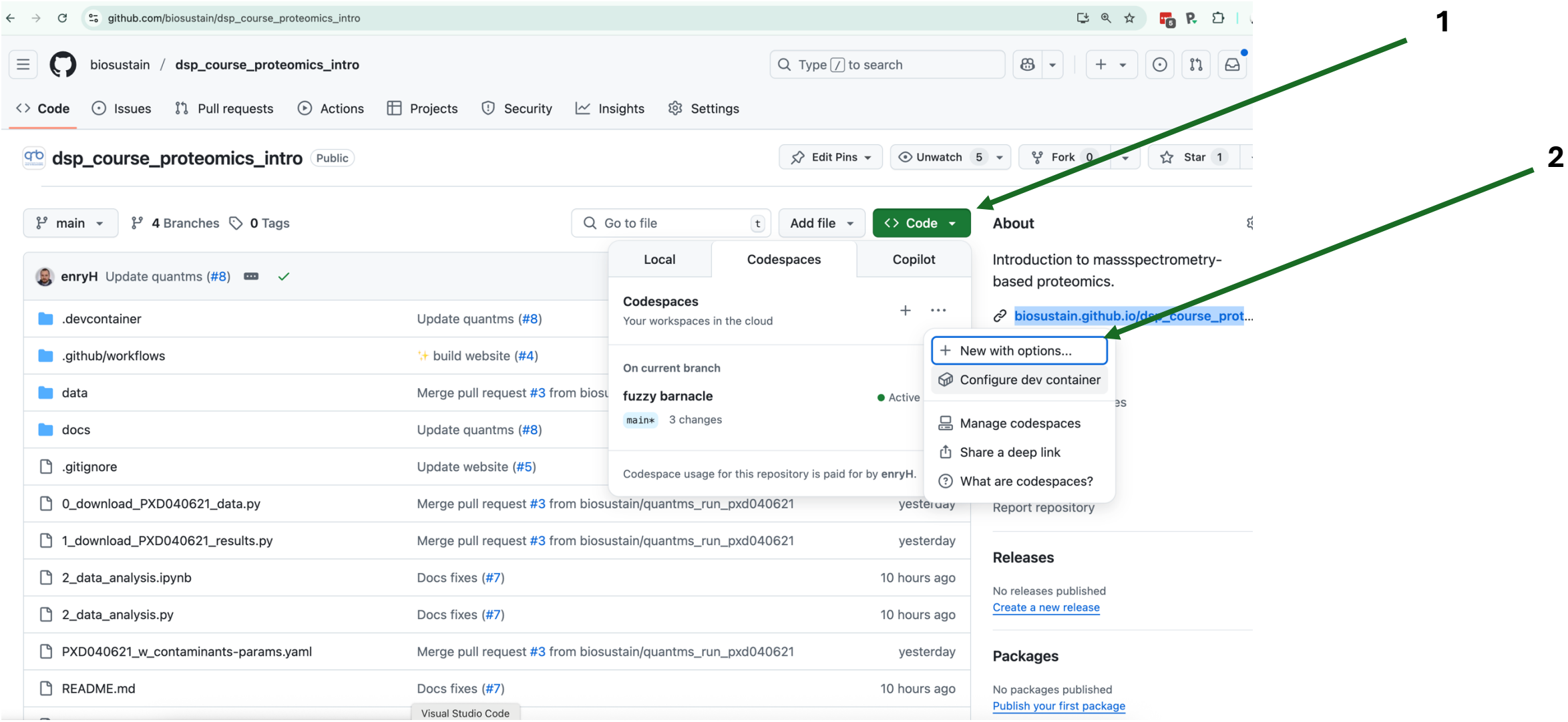
Dr Emily Gulliver

Latest research into the human microbiome begins to untangle how broccoli can alter healthy gut bacteria.

Cruciferous vegetables, including broccoli, cauliflower, brussels sprouts and kale, are often recommended due to the presence of the antioxidant sulforaphane, which is thought to be beneficial for general health and wellbeing and in treating diseases such as cancer.

https://www.hudson.org.au/news/does-broccoli-boost-bad-gut-bacteria/

Let’s start the workflow first



https://github.com/biosustain/dsp_course_proteomics_intro

Codespace with 4 cores (and 16GB of memory)

Create codespace for

biosustain/dsp_course_proteomics_intro

Branch
This branch will be checked out on creation
main

Dev container configuration
Your codespace will use this configuration
nextflow-training

Region
Your codespace will run in the selected region
Europe West

Machine type
Resources for your codespace
4-core

Create codespace

1

2

Sample Data Relationship Format (SDRF) file

Describes the design of an experiment (metadata): Defines in a standardized manner details about how the experiment was performed - sample information, organism, phenotype/grouping, etc.

PXD040621.sdrf

File Edit View Insert Format Data Tools Extensions Help

AC9 | Sulforaphane

	A	B	F	O	Q	AC
1	source name	characteristics[biological replicate]	characteristics[organism]	comment[cleavage agent details]	comment[data file]	factor value[phenotype]
2	sample 1	1	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_DMSO_rep1_EG-1.raw	Control
3	sample 2	2	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_DMSO_rep2_EG-2.raw	Control
4	sample 3	3	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_DMSO_rep3_EG-3.raw	Control
5	sample 4	4	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_DMSO_rep4_EG-4.raw	Control
6	sample 5	1	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_Suf_rep1_EG-5.raw	Sulforaphane
7	sample 6	2	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_Suf_rep2_EG-6.raw	Sulforaphane
8	sample 7	3	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_Suf_rep3_EG-7.raw	Sulforaphane
9	sample 8	4	Escherichia coli	NT=Trypsin; AC=1001251; CS=(?⇐[KR])(?!P)	20220830_JL-4884_Forster_Ecoli_Suf_rep4_EG-8.raw	Sulforaphane

LessSDRF tool helps to create these tables

View online

Thanks again to Caroline Jachmann. She created the first version of this SDRF file!

FASTA File with UNIPROT protein sequences

…

>sp|P0DSF9|EVGL_ECOLI Protein EvgL OS=Escherichia coli (strain K12) OX=83333 GN=evgL PE=1 SV=1

MLHCKGNNL

>sp|P0DSH1|YSAE_ECOLI Protein YsaE OS=Escherichia coli (strain K12) OX=83333 GN=ysaE PE=1 SV=1

MRNAVSKAGIISRRRLLLFQFAG

>sp|P0DV20|YTCB_ECOLI Protein YtcB OS=Escherichia coli (strain K12) OX=83333 GN=ytcB PE=1 SV=1

MHLQLIKDNIHSVVICYT

…

>CON_ENSEMBL:ENSBTAP00000038329 (Bos taurus) 9 kDa protein

LPENVTPEEQHKGTSVIHKAVLDVGEEGTEGAAVTAVVMATSSLLHTLTVSFNRPFLLSI

FCKETQSIIFLGKVTNPKEA

…

- 4413 known *E. coli* proteins
- 246 known contaminants sequences

Spectra files

- ThermoFisher instruments: raw files
- Open standard, text based: mzML files
- We use mzML files directly to skip the spectra extraction as quantms run on mzML files

One MS1 spectrum with two M2 spectra (DDA)

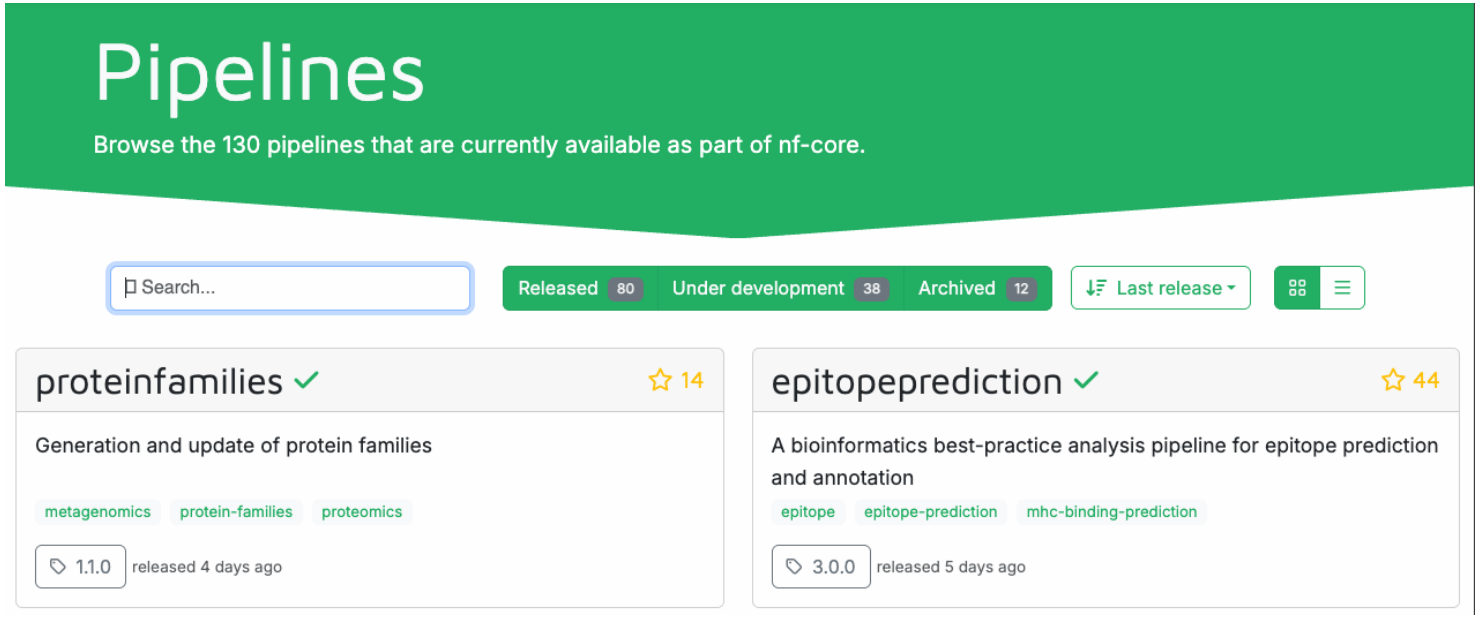

```
<spectrum id="controllerType=0 controllerNumber=1 scan=6" index="5" defaultArrayLength="736">
  <cvParam cvRef="MS" accession="MS:1000511" value="1" name="ms level" />
  <cvParam cvRef="MS" accession="MS:1000579" value="" name="MS1 spectrum" />
  <cvParam cvRef="MS" accession="MS:1000130" value="" name="positive scan" />
  <cvParam cvRef="MS" accession="MS:1000285" value="15248891" name="total ion current" />
  <cvParam cvRef="MS" accession="MS:1000127" value="" name="centroid spectrum" />
  <cvParam cvRef="MS" accession="MS:1000504" value="401.923461914063" name="base peak m/z" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
  <cvParam cvRef="MS" accession="MS:1000505" value="2704116.25" name="base peak intensity" unitAccession="MS:1000131" unitName="number of detector counts" unitCvRef="MS" />
  <cvParam cvRef="MS" accession="MS:1000528" value="375.884124755859" name="lowest observed m/z" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
  <cvParam cvRef="MS" accession="MS:1000527" value="1665.40612792969" name="highest observed m/z" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
  <scanList count="1">
    <cvParam cvRef="MS" accession="MS:1000795" value="" name="no combination" />
    <scan instrumentConfigurationRef="IC1">
      <cvParam cvRef="MS" accession="MS:1000016" value="6.0323342" name="scan start time" unitAccession="UO:0000031" unitName="minute" unitCvRef="UO" />
      <cvParam cvRef="MS" accession="MS:1000512" value="FTMS + p NSI Full ms [375.0000-1800.0000]" name="filter string" />
      <cvParam cvRef="MS" accession="MS:1000927" value="50" name="ion injection time" unitAccession="UO:0000028" unitName="millisecond" unitCvRef="UO" />
      <scanWindowList count="1">
        <scanWindow>
          <cvParam cvRef="MS" accession="MS:1000501" value="375" name="scan window lower limit" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
          <cvParam cvRef="MS" accession="MS:1000500" value="1800" name="scan window upper limit" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
        </scanWindow>
      </scanWindowList>
    </scan>
  </scanList>
  <binaryDataArrayList count="2">
    <binaryDataArray encodedLength="3152">
      <cvParam cvRef="MS" accession="MS:1000514" value="" name="m/z array" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
      <cvParam cvRef="MS" accession="MS:1000523" value="" name="64-bit float" />
      <cvParam cvRef="MS" accession="MS:1000574" value="" name="zlib compression" />
      <binary>eJwt03lYVXUawPHjVumjGA1lG5dTGtMi3MYUG32496hJOiNX3EvlcjLNsEDNckGWIyoGxaIwiU3AsUWcZ2TJEnEm5Wimg48C466j15M12ZRgU1paCTPP+/399Xne9fceFk3TzIeylxna//0Vjck5ovM26h1oDLdE
    </binaryDataArray>
    <binaryDataArray encodedLength="4356">
      <cvParam cvRef="MS" accession="MS:1000515" value="" name="intensity array" unitAccession="MS:1000131" unitName="number of counts" unitCvRef="MS" />
      <cvParam cvRef="MS" accession="MS:1000523" value="" name="64-bit float" />
      <cvParam cvRef="MS" accession="MS:1000574" value="" name="zlib compression" />
      <binary>eJwtWHlcT+kXviLFSI2MTJi5ZWIaTTGDsl+EkT3KVtypUVl+dhr5TG4qo7Jk8FPWm32pjC1kuxUlEu0JuSVDpozK0pdkPj3P/PV+3vu+73nPec45zznvFQRBuRdyQxIEQWzvf6ppFBz63sIYceZq06jafJHbNGr
    </binaryDataArray>
  </binaryDataArrayList>
</spectrum>
```

One MS1 spectrum with two M2 spectra (DDA)


```
<spectrum id="controllerType=0 controllerNumber=1 scan=6" index="5" defaultArrayLength="736">
<cvParam cvRef="MS" accession="MS:1000511" value="1" name="ms level" />
<cvParam cvRef="MS" accession="MS:1000579" value="" name="MS1 spectrum" />
<cvParam cvRef="MS" accession="MS:1000130" value="" name="positive scan" />
<cvParam cvRef="MS" accession="MS:1000285" value="15248891" name="total ion current" />
<cvParam cvRef="MS" accession="MS:1000127" value="" name="centroid spectrum" />
<cvParam cvRef="MS" accession="MS:1000504" value="401.923461914063" name="base peak m/z" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
<cvParam cvRef="MS" accession="MS:1000505" value="2704116.25" name="base peak intensity" unitAccession="MS:1000131" unitName="number of detector counts" unitCvRef="MS" />
<cvParam cvRef="MS" accession="MS:1000528" value="375.884124755859" name="lowest observed m/z" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
<cvParam cvRef="MS" accession="MS:1000527" value="1665.40612792969" name="highest observed m/z" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
<scanList count="1">
 <cvParam cvRef="MS" accession="MS:1000795" value="" name="no combination" />
 <scan instrumentConfigurationRef="IC1">
  <cvParam cvRef="MS" accession="MS:1000016" value="6.0323342" name="scan start time" unitAccession="UO:0000031" unitName="minute" unitCvRef="UO" />
  <cvParam cvRef="MS" accession="MS:1000512" value="FTMS + p NSI Full ms [375.0000-1800.0000]" name="filter string" />
  <cvParam cvRef="MS" accession="MS:1000927" value="50" name="ion injection time" unitAccession="UO:0000028" unitName="millisecond" unitCvRef="UO" />
  <scanWindowList count="1">
   <scanWindow>
    <cvParam cvRef="MS" accession="MS:1000501" value="375" name="scan window lower limit" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
    <cvParam cvRef="MS" accession="MS:1000500" value="1800" name="scan window upper limit" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
   </scanWindow>
  </scanWindowList>
 </scan>
</scanList>
<binaryDataArrayList count="2">
 <binaryDataArray encodedLength="3152">
  <cvParam cvRef="MS" accession="MS:1000514" value="" name="m/z array" unitAccession="MS:1000040" unitName="m/z" unitCvRef="MS" />
  <cvParam cvRef="MS" accession="MS:1000523" value="" name="64-bit float" />
  <cvParam cvRef="MS" accession="MS:1000574" value="" name="zlib compression" />
  <binary> ... 93jb+B7DdOsY=</binary>
 </binaryDataArray>
 <binaryDataArray encodedLength="4356">
  <cvParam cvRef="MS" accession="MS:1000515" value="" name="intensity array" unitAccession="MS:1000131" unitName="number of counts" unitCvRef="MS" />
  <cvParam cvRef="MS" accession="MS:1000523" value="" name="64-bit float" />
  <cvParam cvRef="MS" accession="MS:1000574" value="" name="zlib compression" />
  <binary>eJwtWHlcT+ ... yNfr3NnHXSynXsNq2F/69YyX/gWrVuzh</binary>
 </binaryDataArray>
</binaryDataArrayList>
</spectrum>
```

Nextflow and nf-core pipelines

- Nextflow is a workflow executor (analysis steps combined in an execution graph)
  - Reproducibile and scalable
- Nf-core is a collection of workflows maintained by nextflow and an open-source community



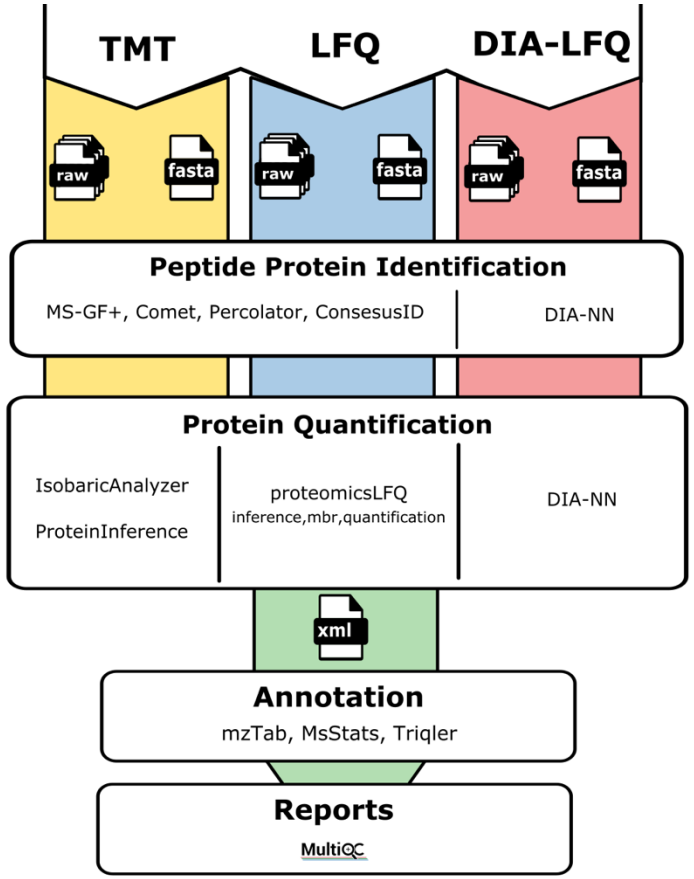
nextflow

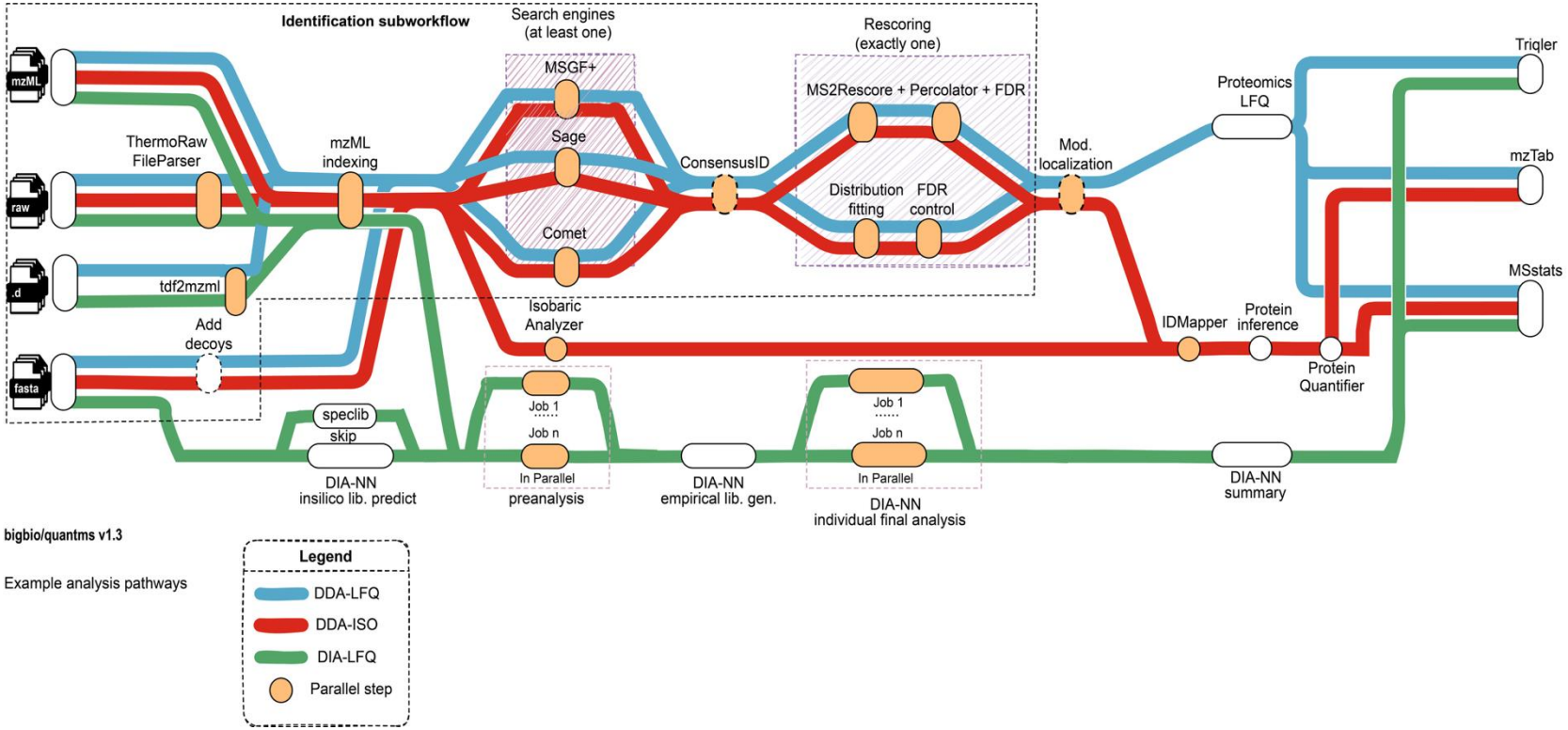
nf-core

A global community effort to collect a curated set of open-source analysis pipelines built using Nextflow.

QuantMS (nf-core compatible)

bigbio/quantms





Output Folder

```
# results/PXD040621
decoydatabase
extractpsmfeature
idfilter
idscoreswitcher
mzmlindexing
mzmlstatistics
percolator
pipeline_info.   # information to re-run pipeline (at least for 1.4.0)
Proteomicslfq  # most relevant
psmclean
sdrfparsing
searchenginecomet
summarypipeline
```

Some parameters

```
database: data/fasta/merged_ecoli_with_contaminants.fasta
input: data/PXD040621/PXD040621.sdrf.tsv
outdir: results/PXD040621

# only relevant for 1.3.0
max_memory: 15 GB
max_cpus: 4

# local_input_type: raw
local_input_type: mzML # default
# ! root_folder only applies to the mzML or raw spectrum files
root_folder: /workspaces/dsp_course_proteomics_intro/data/PXD040621/mzML/
publish_dir_mode: symlink
# running msstats: Only two pdfs which we do not use.
skip_post_msstats: true
```

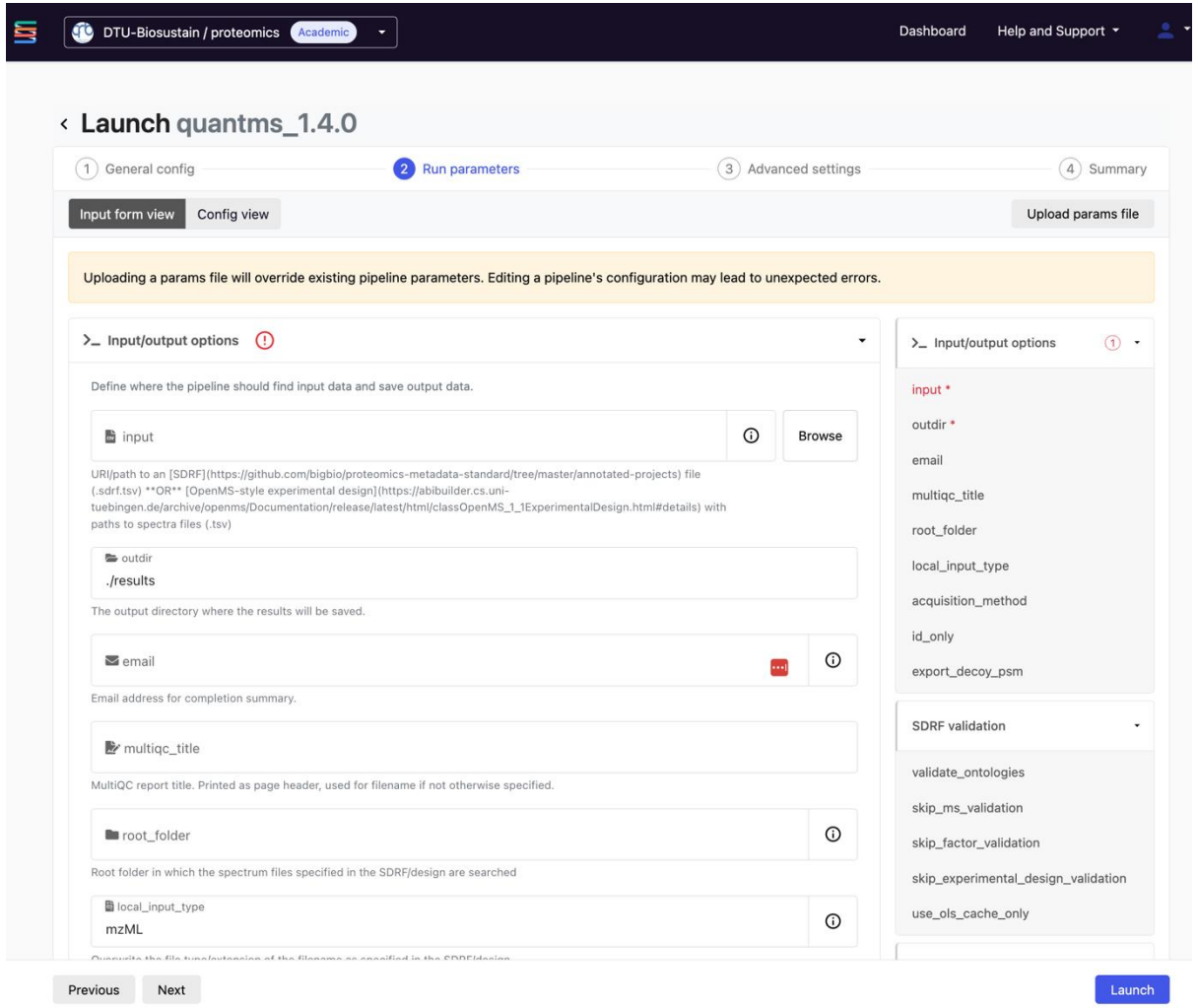
```
min_peptide_length: 6
max_peptide_length: 40


fdr_level: psm_level_fdrs
min_precursor_charge: 2
max_precursor_charge: 4

protein_quant: unique_peptides
min_peptides_per_protein: 1

precursor_mass_tolerance: 5
precursor_mass_tolerance_unit: ppm
fragment_mass_tolerance: 0.03
fragment_mass_tolerance_unit: Da
protein_level_fdr_cutoff: 0.01
```

Seqera interface for workflow



- Parameters grouped and in order
- Based on schema file
- For more information: Schema file documentation

Data Analysis hands-on using acore

14th May 2025

Henry Webel

PXD040621: Broccoli

Aim: identify the impact of sulforaphane on the human gut microbiome (in anaerobic conditions of the gut)

Scientific Story:

- Phylogeny of selected strains and screening of growth kinetics of selected strains under sulforaphane in anaerobic conditions (Figure 1)
- Growth profile of selected *E. coli* Strain E2348/69 in anaerobic and aerobic conditions with varying amounts of sulforaphane
- Proteomics analysis of most growing strain
- Metabolomics analysis of most growing strain

- Article and PXD040621 on Pride archive

Results

"Proteomics identified an increase in anaerobic respiration in *E. coli* grown in the presence of sulforaphane, indicating ~~suggesting~~ that sulforaphane may be acting as an additional carbon source in these bacteria. The metabolic profile following growth in sulforaphane, showed that sulfo-raphane increased the production of metabolites that are also known to interact with host tissues to decrease inflammation."

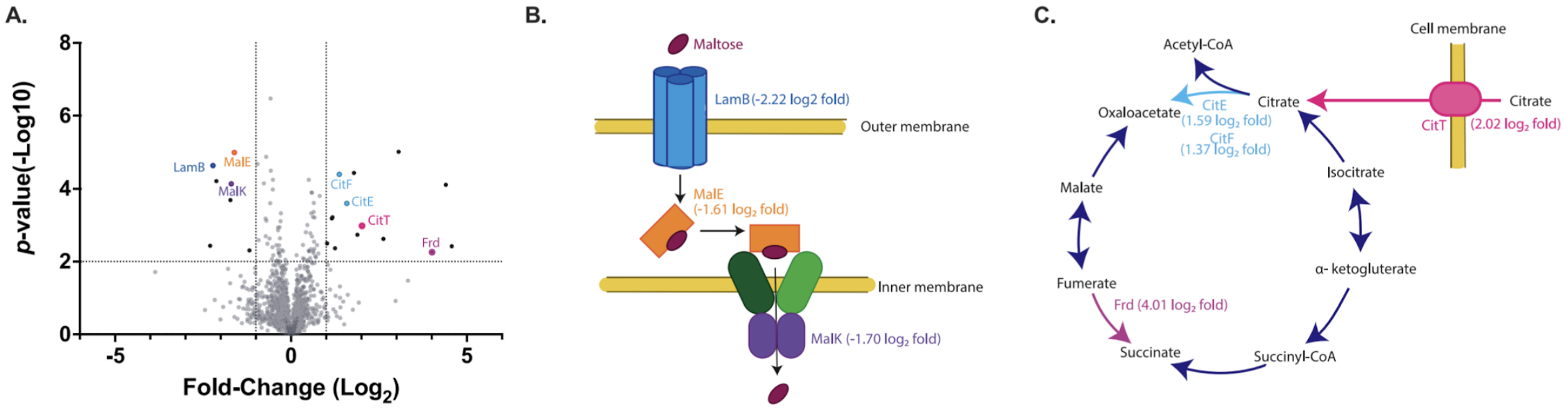
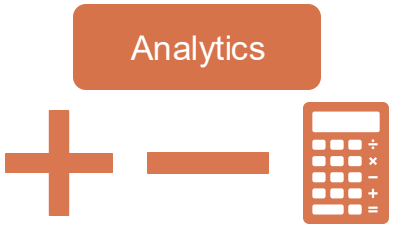


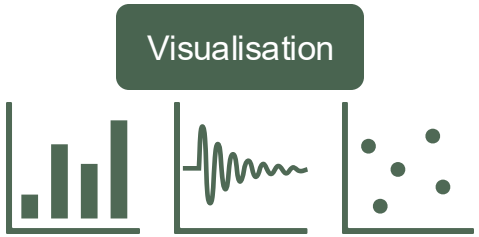
Fig. 3. Differentially produced proteins in *E. coli* EPEC ECE2348/69. A. Volcano plot of the change in protein production of individual proteins (dots) produced by. *E. coli* ECE2348/69 in response to the presence of sulforaphane. Significantly altered proteins (black or coloured dots) showed a $\log_2$ fold-change < -1 or > 1 (vertical dashed lines) with a p-value < 0.05 (horizontal dashed lines). **B.** Schematic representation of proteins with significantly decreased production involved in maltose uptake. **C.** Schematic representation of proteins with significantly increased production (coloured labels) involved in anaerobic respiration.

MoNA and DSP Open Source Libraries

Multi-omics Network Analytics

DTU



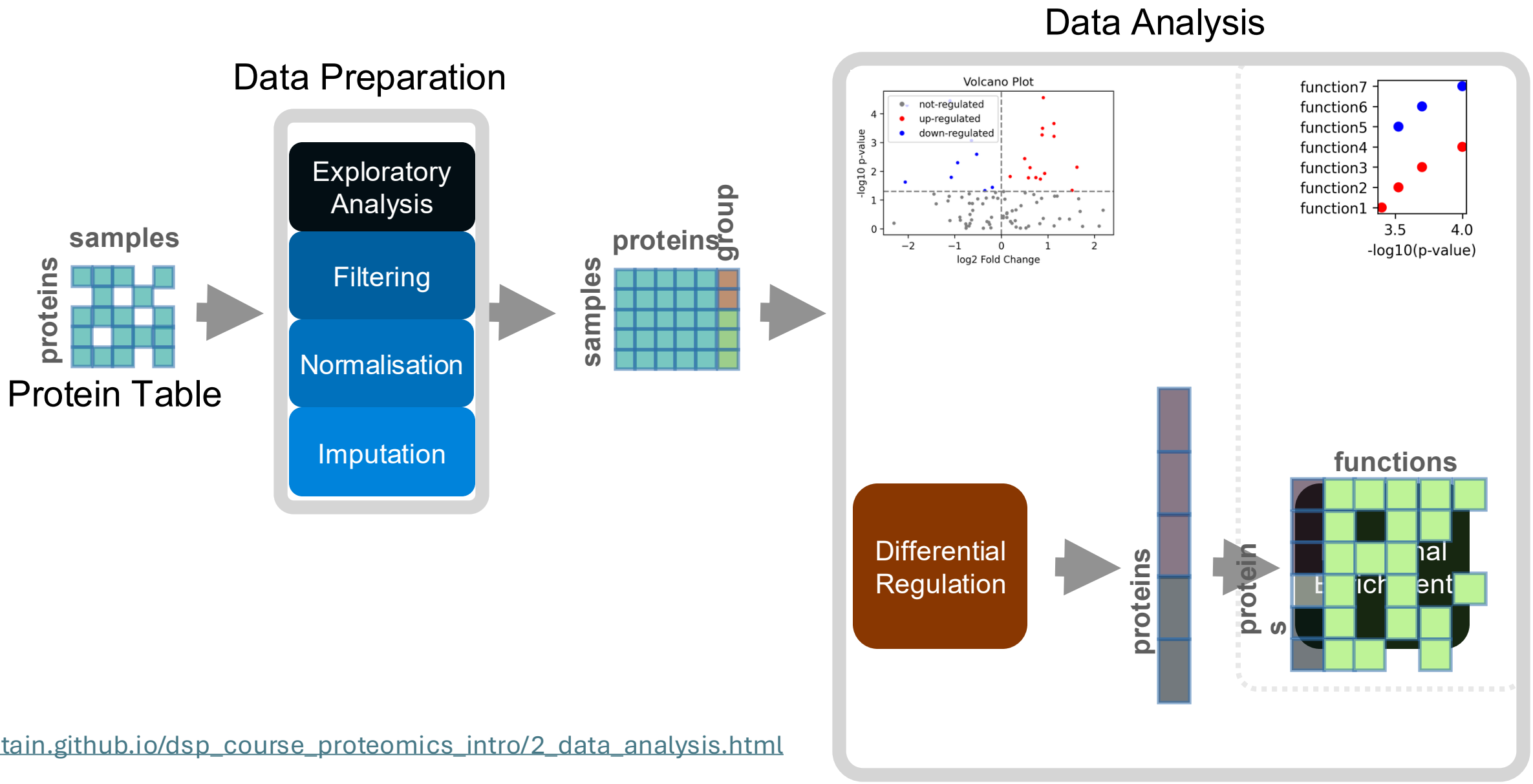


Analytics core library — analytics-core.readthedocs.io/

Visualization core library — github.com/Multiomics-Analytics-Group/vuecore

Automated Reporting library and cli — github.com/Multiomics-Analytics-Group/vuegen

Acore – Analytical core – workflow example



biosustain.github.io/dsp_course_proteomics_intro/2_data_analysis.html

Acore – Analytical core – workflow example

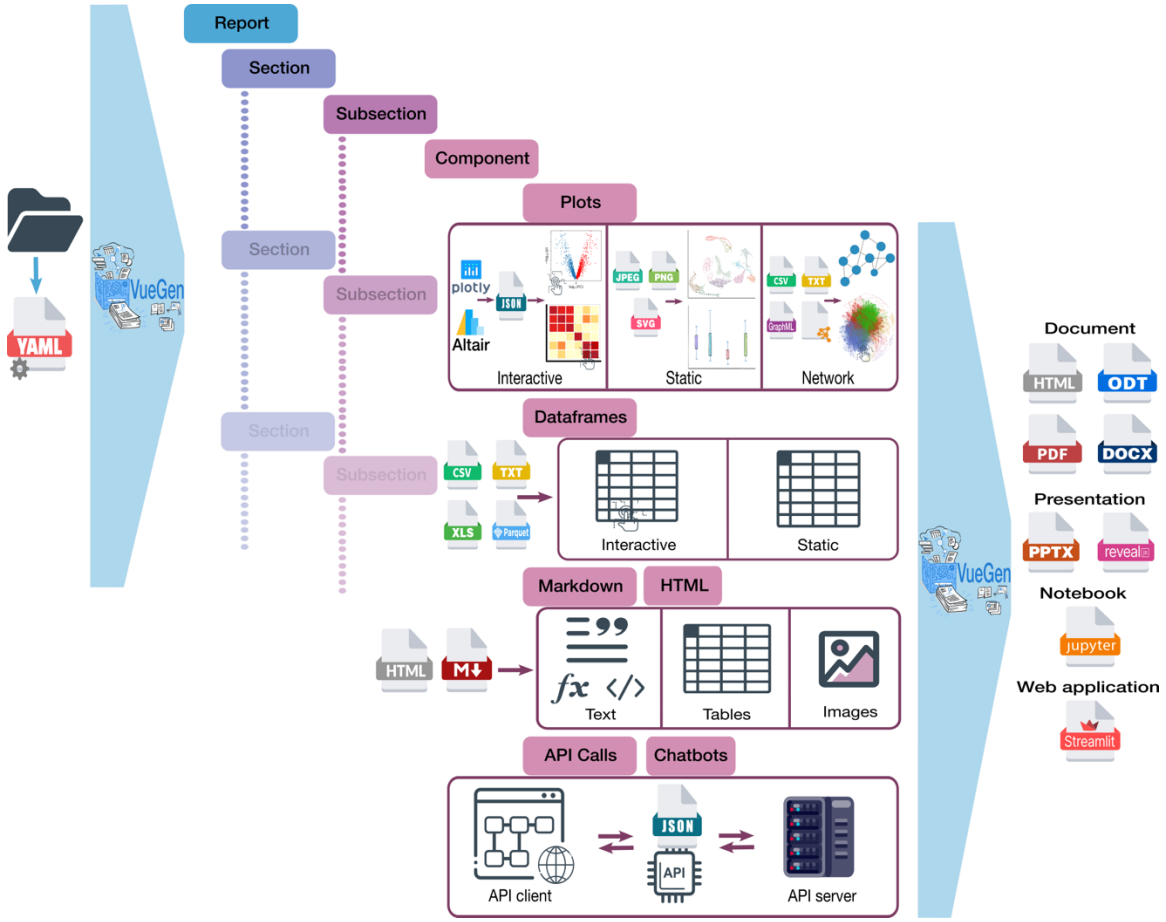
To Do

- Can be executed in an example notebook
- Which can be customized from the command line
- Which can operate on different datasets

Report Generation using Vuegen

VueGen in a nutshell

- Turn results folder into different reports
- We collect plots and dataframes from our analysis example
- The analysis notebooks will export the relevant files



VueGen implementation details