Data Challenge Presentation

Teams: Fork Gladius + Issue Saber

EURO-LABS: Advanced Training School on Collaborative Software Development, GANIL, France, 02-06 February, 2026

Fork Gladius

- Konstantina Palli
- Yuma Ohsawa
- Francesco Esposito
- Francisco Geraldes Barba

Issue Saber

- Víctor Martínez Nouvilas
- Sandipan Dawn
- Francisca Filipa Pereira Barbosa Afonso
- Krzysztof Solak

PHYSICS CASE

Picture of AGATA detector taken in GANIL

98Zr
30.7 s

β^-=100%

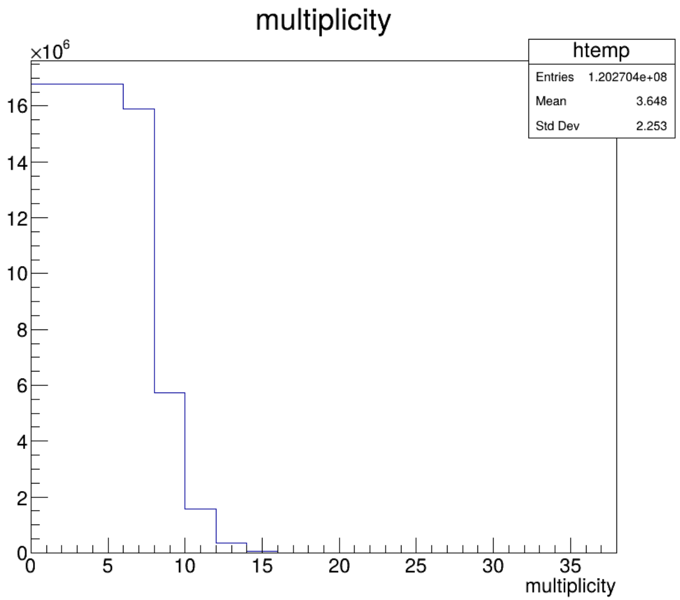
90Br
1.91 s

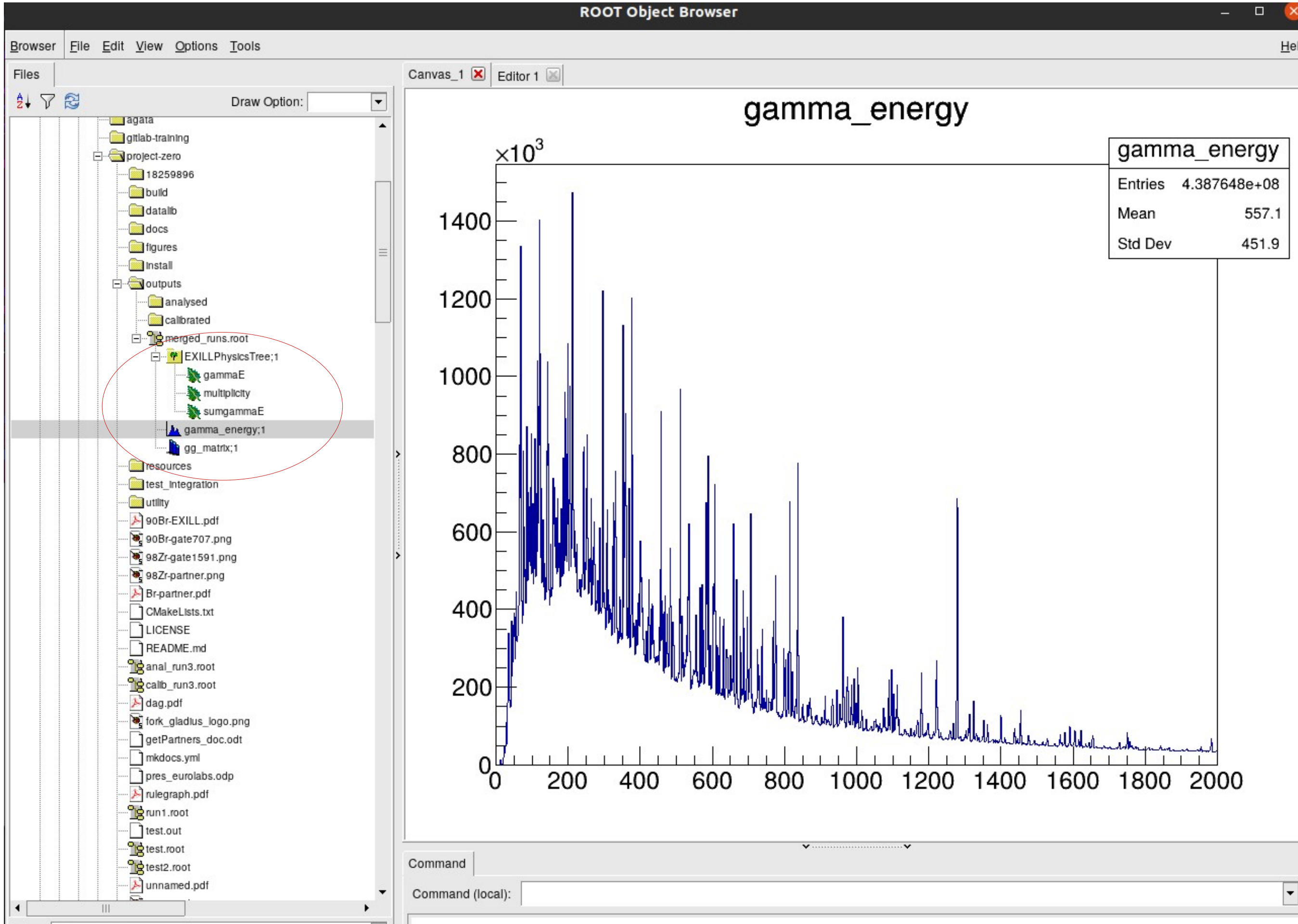
β^-=100%
β^-n=25.3%

Setup : VAMOS + AGATA
Gamma rays spectroscopy
Gammas emitted in flight (Doppler correction needed)

=> Only single gamma spectrum

EXILL: Data

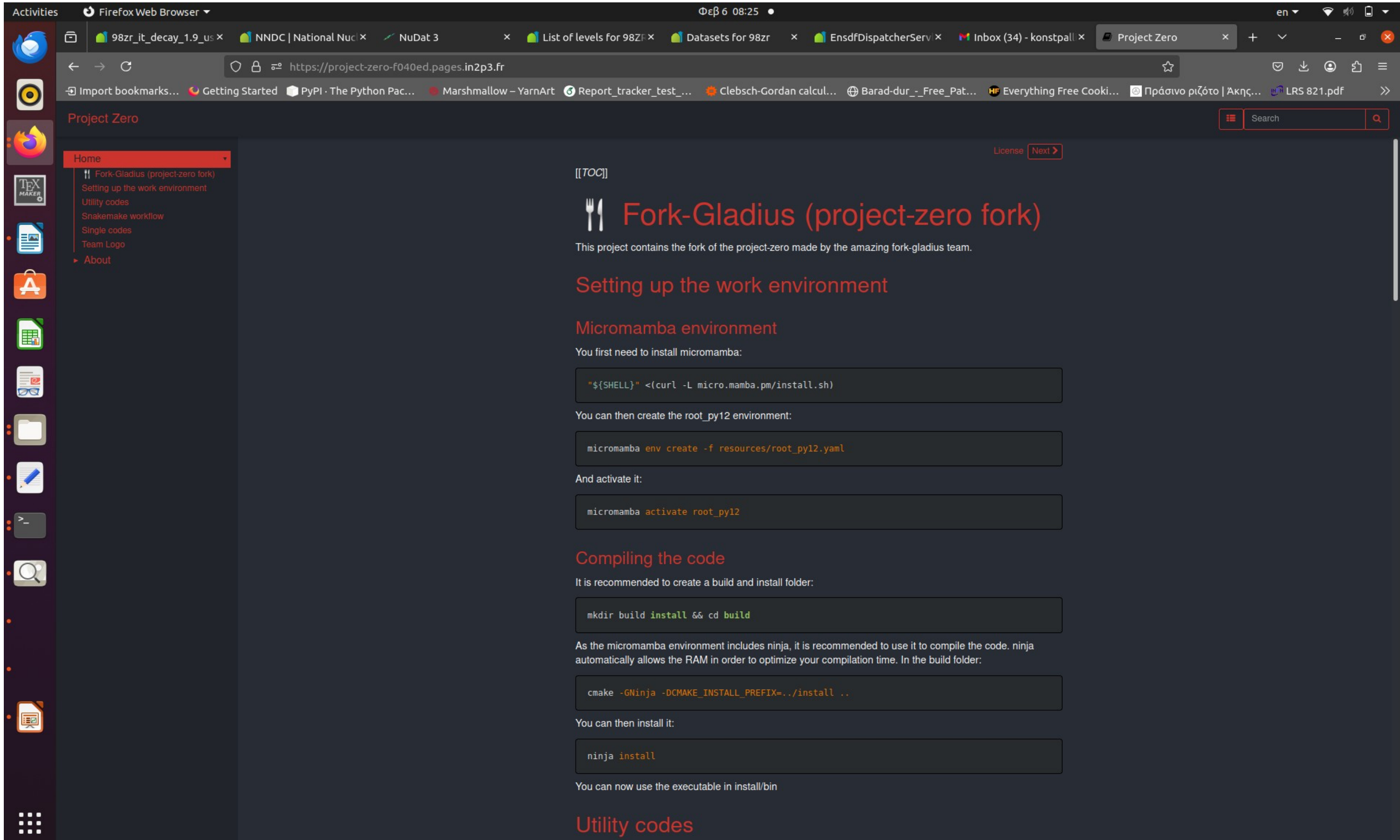




agata
gitlab-training
project-zero
  18259896
  build
  datalib
  docs
  figures
  install
  outputs
    analysed
    calibrated
    merged_runs.root
      EXILLPhysicsTree;1
        gammaE
        multiplicity
        sumgammaE
      gamma_energy;1
      gg_matrix;1
  resources
  test_integration
  utility
  90Br-EXILL.pdf
  90Br-gate707.png
  98Zr-gate1591.png
  98Zr-partner.png
  Br-partner.pdf
  CMakeLists.txt
  LICENSE
  README.md
  anal_run3.root
  calib_run3.root
  dag.pdf
  fork_gladius_logo.png
  getPartners_doc.odt
  mkdocs.yml
  pres_eurolabs.odp
  rulegraph.pdf
  run1.root
  test.out
  test.root
  test2.root
  unnamed.pdf

GitLab Page

https://project-zero-f040ed.pages.in2p3.fr/



[[TOC]]

🍴 Fork-Gladius (project-zero fork)

This project contains the fork of the project-zero made by the amazing fork-gladius team.

Setting up the work environment

Micromamba environment

You first need to install micromamba:

```
"${SHELL}" <(curl -L micro.mamba.pm/install.sh)
```

You can then create the root_py12 environment:

```
micromamba env create -f resources/root_py12.yaml
```

And activate it:

```
micromamba activate root_py12
```

Compiling the code

It is recommended to create a build and install folder:

```
mkdir build install && cd build
```

As the micromamba environment includes ninja, it is recommended to use it to compile the code. ninja automatically allows the RAM in order to optimize your compilation time. In the build folder:

```
cmake -GNinja -DCMAKE_INSTALL_PREFIX=../install ..
```

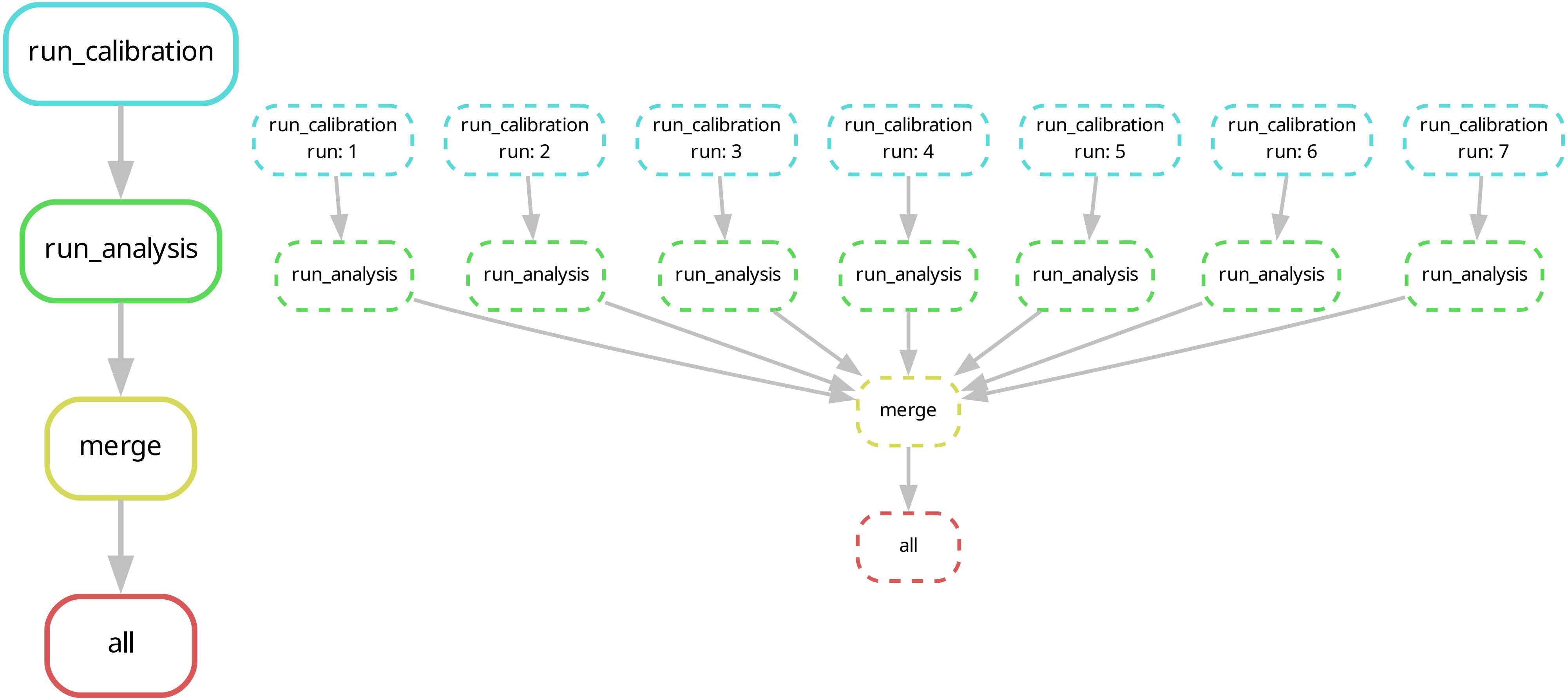
You can then install it:

```
ninja install
```

You can now use the executable in install/bin

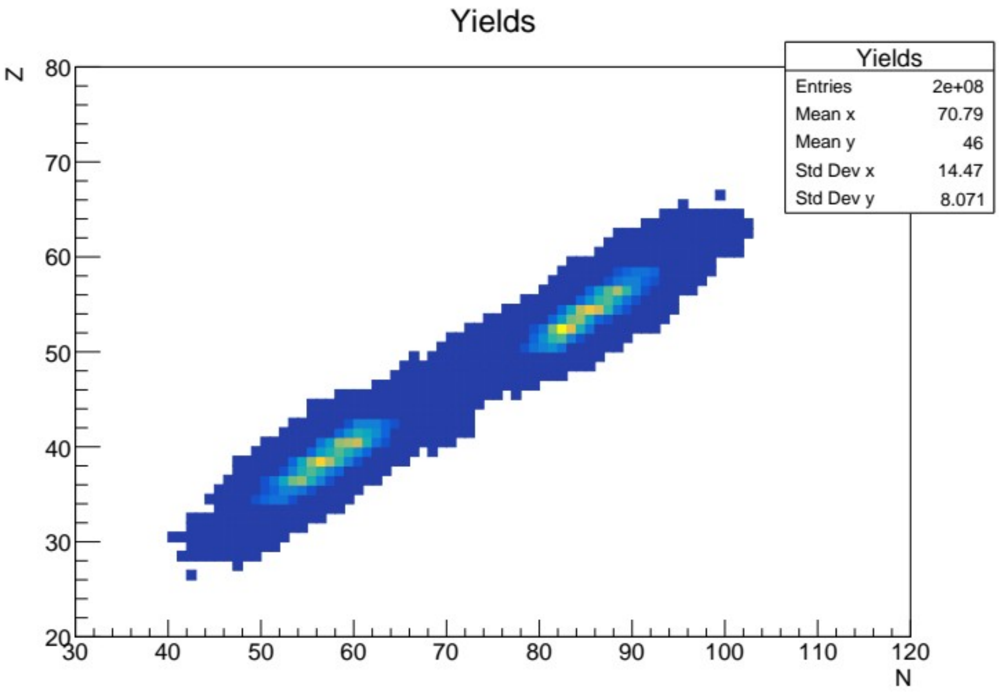
Utility codes

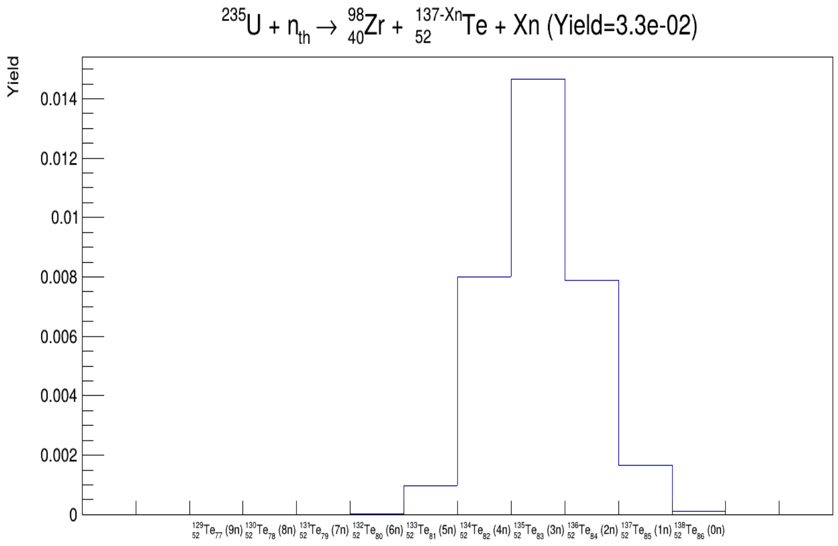
Snakemake structure



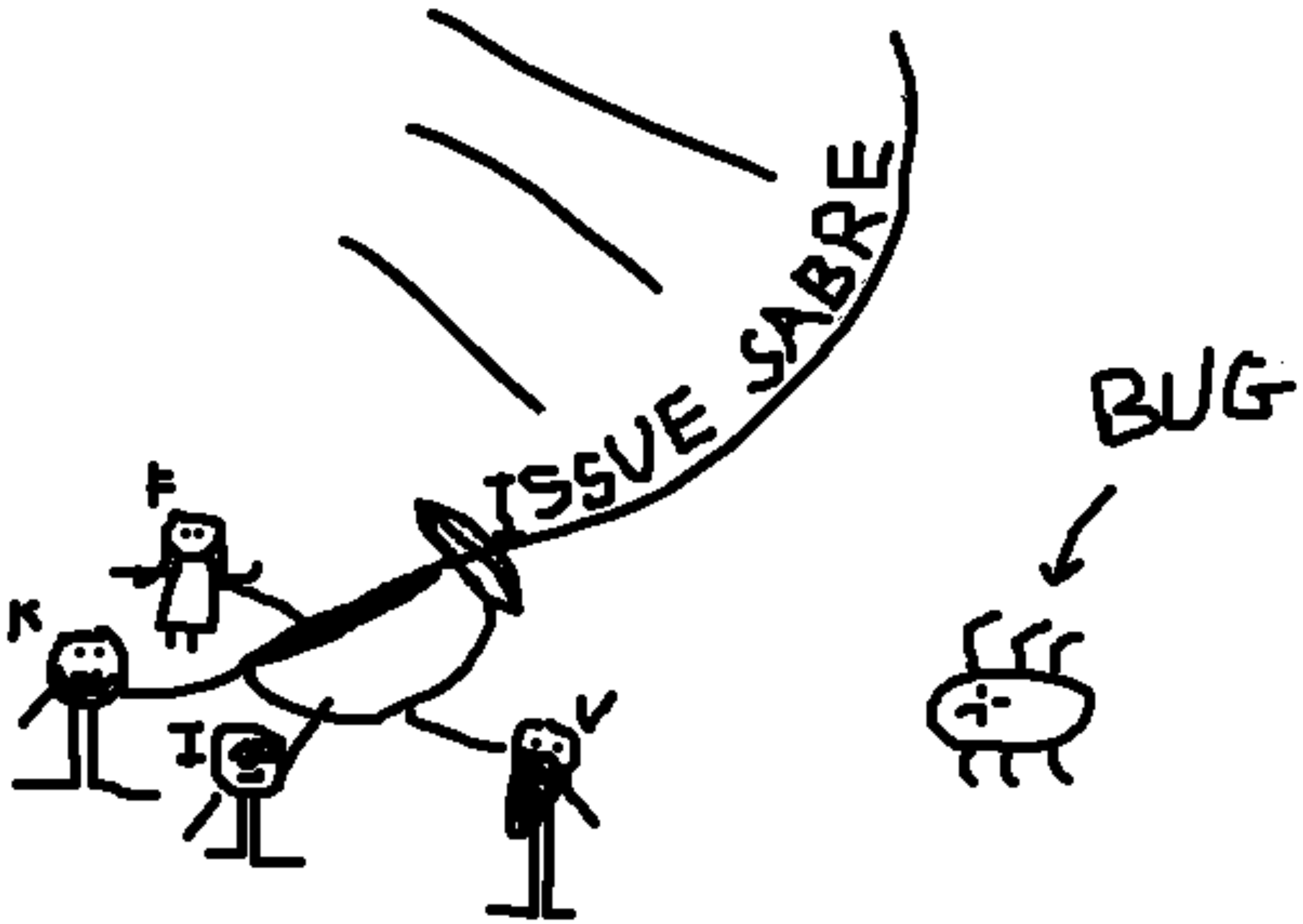
get_partner Code

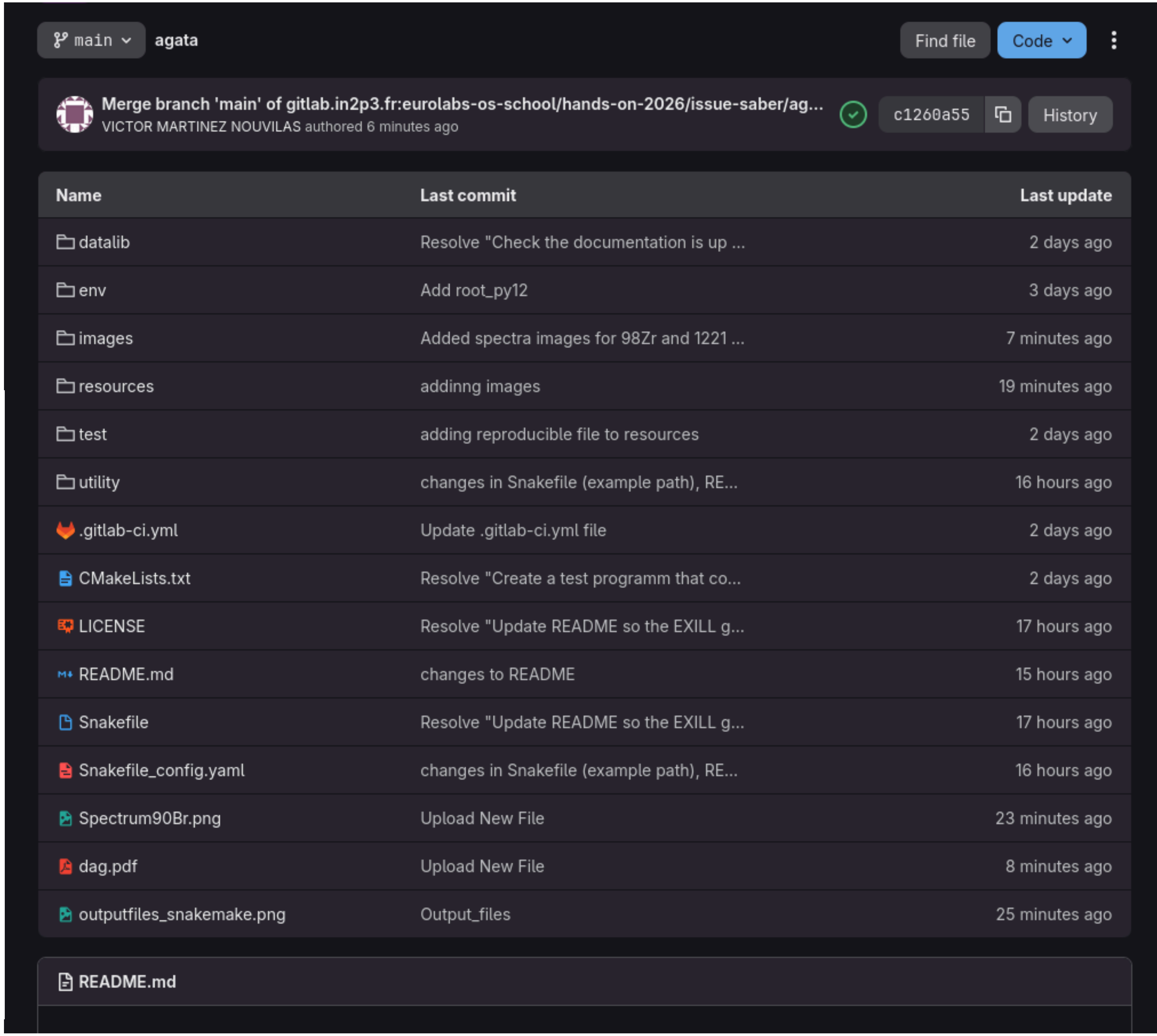
EXILl_Partners.root





Issue-saber





agata

Merge branch 'main' of gitlab.in2p3.fr:eurolabs-os-school/hands-on-2026/issue-saber/ag...
VICTOR MARTINEZ NOUVILAS authored 6 minutes ago

c1260a55

Name	Last commit	Last update
datalib	Resolve "Check the documentation is up ...	2 days ago
env	Add root_py12	3 days ago
images	Added spectra images for 98Zr and 1221 ...	7 minutes ago
resources	addinng images	19 minutes ago
test	adding reproducible file to resources	2 days ago
utility	changes in Snakefile (example path), RE...	16 hours ago
.gitlab-ci.yml	Update .gitlab-ci.yml file	2 days ago
CMakeLists.txt	Resolve "Create a test programm that co...	2 days ago
LICENSE	Resolve "Update README so the EXILL g...	17 hours ago
README.md	changes to README	15 hours ago
Snakefile	Resolve "Update README so the EXILL g...	17 hours ago
Snakefile_config.yaml	changes in Snakefile (example path), RE...	16 hours ago
Spectrum90Br.png	Upload New File	23 minutes ago
dag.pdf	Upload New File	8 minutes ago
outputfiles_snakemake.png	Output_files	25 minutes ago

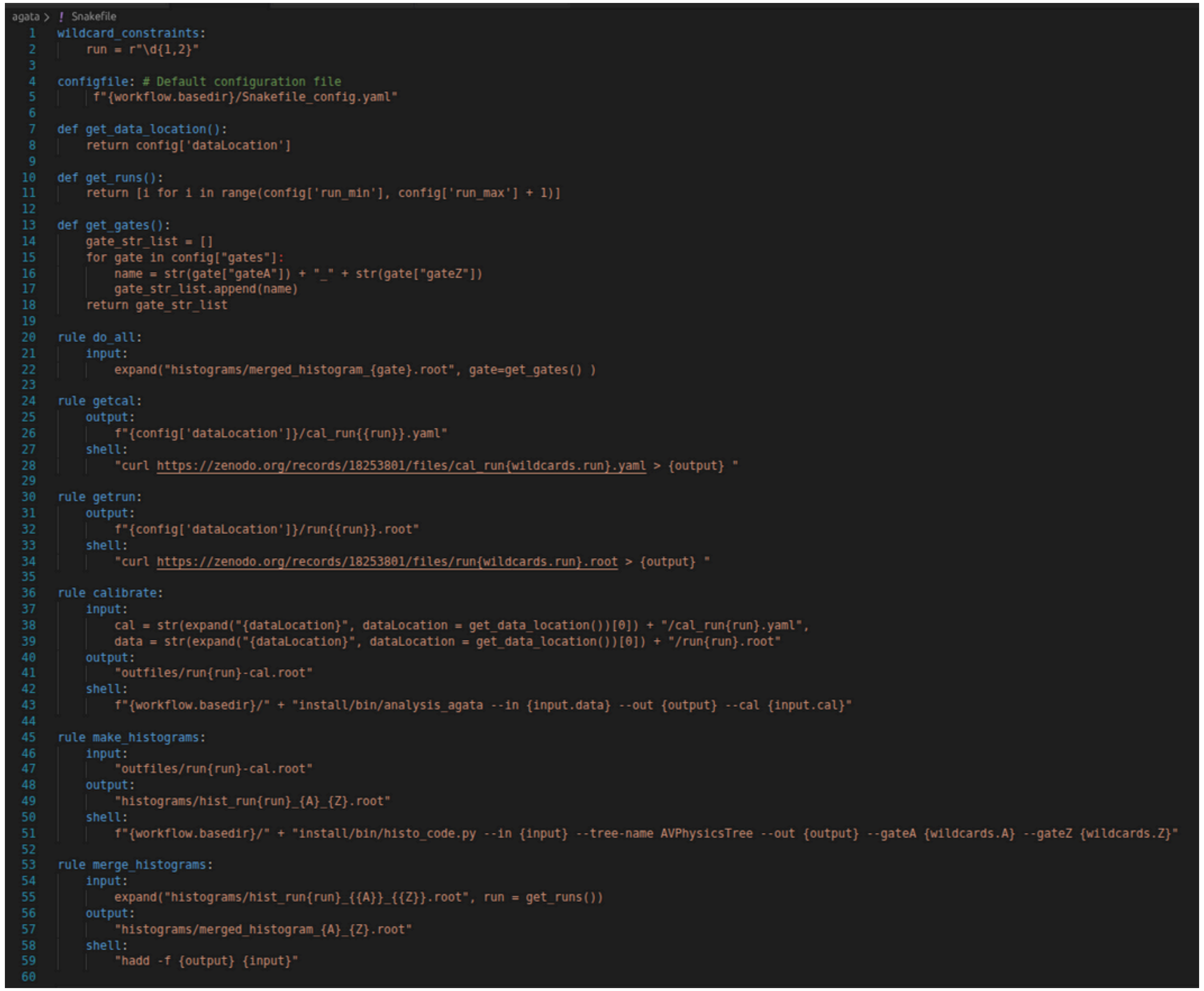
README.md

ANALYSIS WORKFLOW

+ Doppler correction

+ applying A/Z gates

CODES AND ISSUES



```
wildcard_constraints:
    run = r"\d{1,2}"

configfile: # Default configuration file
     f"{workflow.basedir}/Snakefile_config.yaml"

def get_data_location():
    return config['dataLocation']

def get_runs():
    return [i for i in range(config['run_min'], config['run_max'] + 1)]

def get_gates():
    gate_str_list = []
    for gate in config["gates"]:
        name = str(gate["gateA"]) + "_" + str(gate["gateZ"])
        gate_str_list.append(name)
    return gate_str_list

rule do_all:
    input:
        expand("histograms/merged_histogram_{gate}.root", gate=get_gates() )

rule getcal:
    output:
        f"{config['dataLocation']}/cal_run{{run}}.yaml"
    shell:
        "curl https://zenodo.org/records/18253801/files/cal_run{wildcards.run}.yaml > {output} "

rule getrun:
    output:
        f"{config['dataLocation']}/run{{run}}.root"
    shell:
        "curl https://zenodo.org/records/18253801/files/run{wildcards.run}.root > {output} "

rule calibrate:
    input:
        cal = str(expand("{dataLocation}", dataLocation = get_data_location())[0]) + "/cal_run{run}.yaml",
        data = str(expand("{dataLocation}", dataLocation = get_data_location())[0]) + "/run{run}.root"
    output:
        "outfiles/run{run}-cal.root"
    shell:
        f"{workflow.basedir}/" + "install/bin/analysis_agata --in {input.data} --out {output} --cal {input.cal}"

rule make_histograms:
    input:
        "outfiles/run{run}-cal.root"
    output:
        "histograms/hist_run{run}_{A}_{Z}.root"
    shell:
        f"{workflow.basedir}/" + "install/bin/histo_code.py --in {input} --tree-name AVPhysicsTree --out {output} --gateA {wildcards.A} --gateZ {wildcards.Z}"

rule merge_histograms:
    input:
        expand("histograms/hist_run{run}_{{A}}_{{Z}}.root", run = get_runs())
    output:
        "histograms/merged_histogram_{A}_{Z}.root"
    shell:
        "hadd -f {output} {input}"

```

Snakemake structure

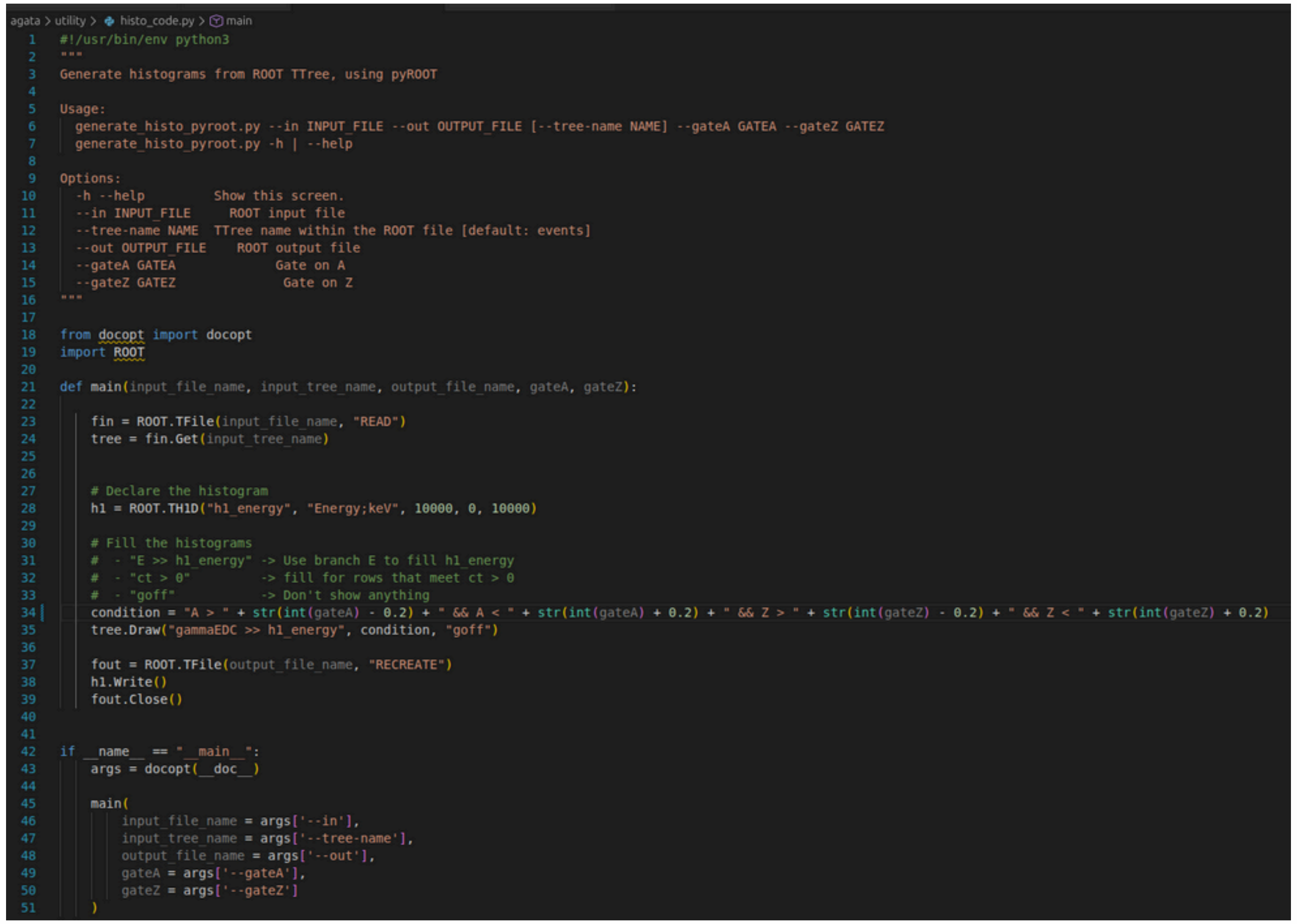
```yaml
# Default configuration
#  - note that yaml allow comments, so put some if needed
run_min: 1
run_max: 21
gates:
  - gate:
    gateA: 98
    gateZ: 40
  - gate:
    gateA: 90
    gateZ: 35

dataLocation: "/home/zeghni-l/Documents/18253801"

```

Configuration for snakemake

Some issues encountered:

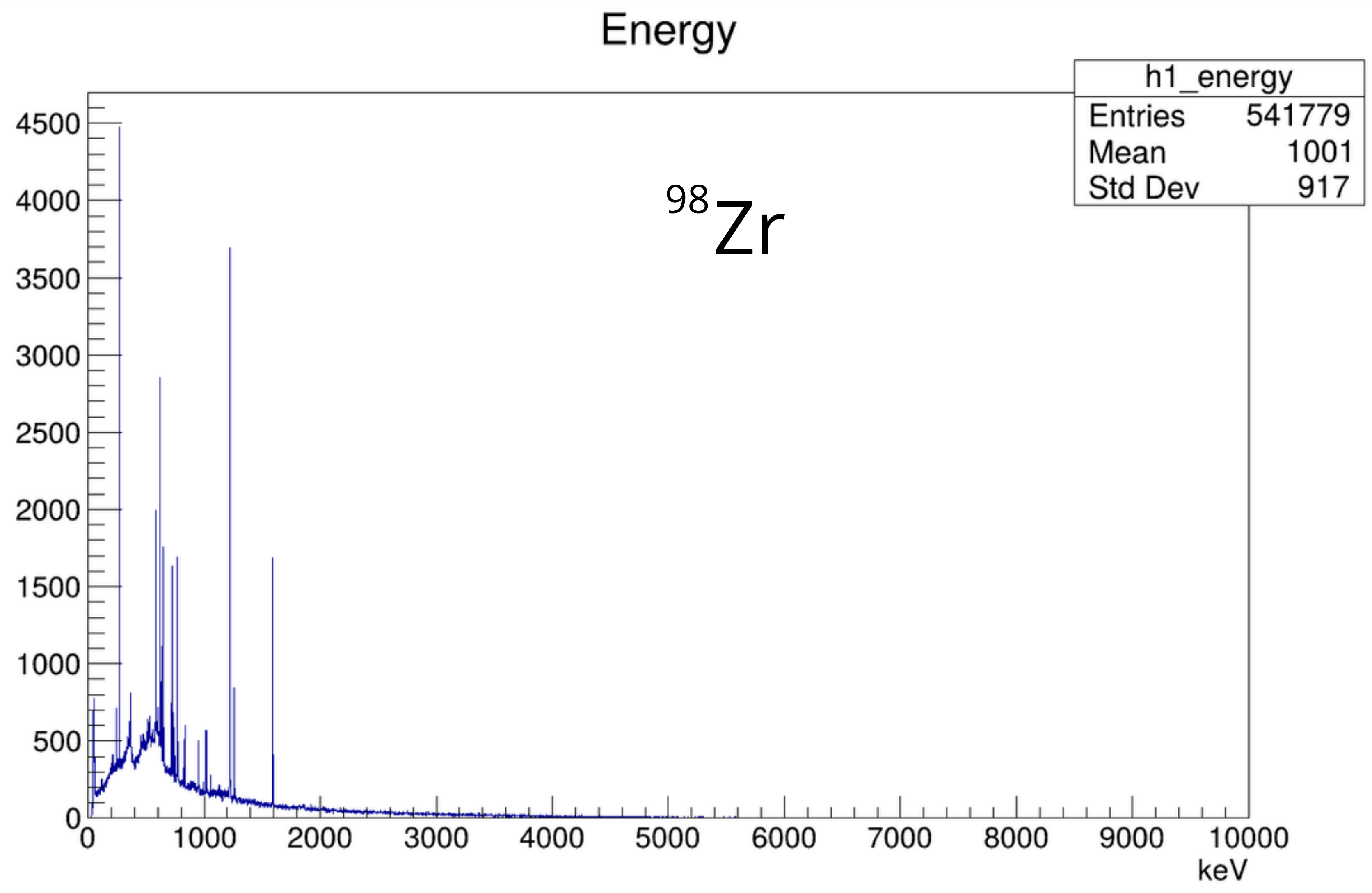
- Merging conflicts
- Path related issues
- README file
- MetaData missing

RESULTS



```python
#!/usr/bin/env python3
"""
Generate histograms from ROOT TTree, using pyROOT

Usage:
  generate_histo_pyroot.py --in INPUT_FILE --out OUTPUT_FILE [--tree-name NAME] --gateA GATEA --gateZ GATEZ
  generate_histo_pyroot.py -h | --help

Options:
  -h --help         Show this screen.
  --in INPUT_FILE     ROOT input file
  --tree-name NAME  TTree name within the ROOT file [default: events]
  --out OUTPUT_FILE    ROOT output file
  --gateA GATEA             Gate on A
  --gateZ GATEZ              Gate on Z
"""

from docopt import docopt
import ROOT

def main(input_file_name, input_tree_name, output_file_name, gateA, gateZ):

    fin = ROOT.TFile(input_file_name, "READ")
    tree = fin.Get(input_tree_name)


    # Declare the histogram
    h1 = ROOT.TH1D("h1_energy", "Energy;keV", 10000, 0, 10000)

    # Fill the histograms
    #  - "E >> h1_energy" -> Use branch E to fill h1_energy
    #  - "ct > 0"         -> fill for rows that meet ct > 0
    #  - "goff"           -> Don't show anything
    condition = "A > " + str(int(gateA) - 0.2) + " && A < " + str(int(gateA) + 0.2) + " && Z > " + str(int(gateZ) - 0.2) + " && Z < " + str(int(gateZ) + 0.2)
    tree.Draw("gammaEDC >> h1_energy", condition, "goff")

    fout = ROOT.TFile(output_file_name, "RECREATE")
    h1.Write()
    fout.Close()


if __name__ == "__main__":
    args = docopt(__doc__)

    main(
        input_file_name = args['--in'],
        input_tree_name = args['--tree-name'],
        output_file_name = args['--out'],
        gateA = args['--gateA'],
        gateZ = args['--gateZ']
    )
```

Python code to obtain histograms
Sets up the gates for A and Z

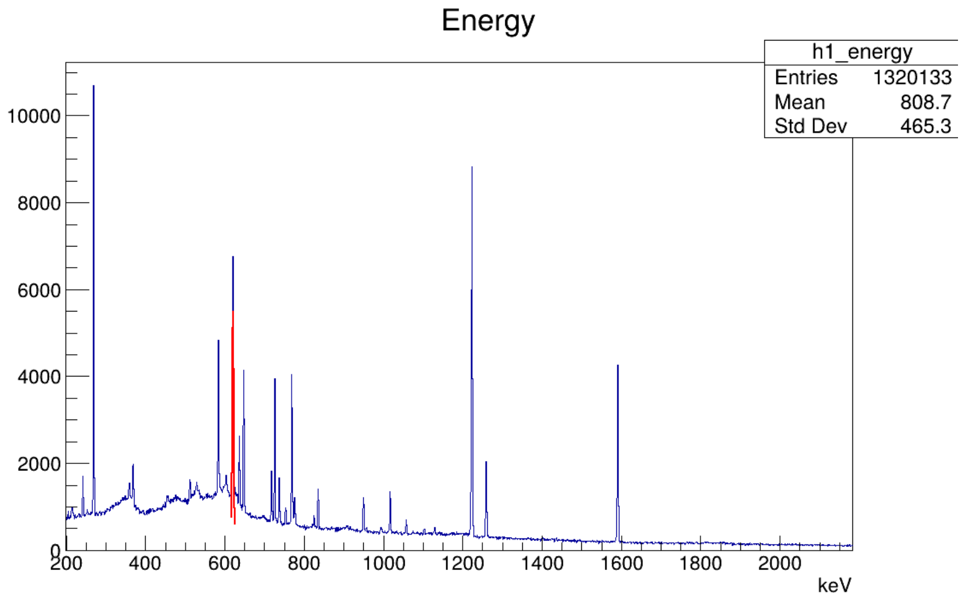


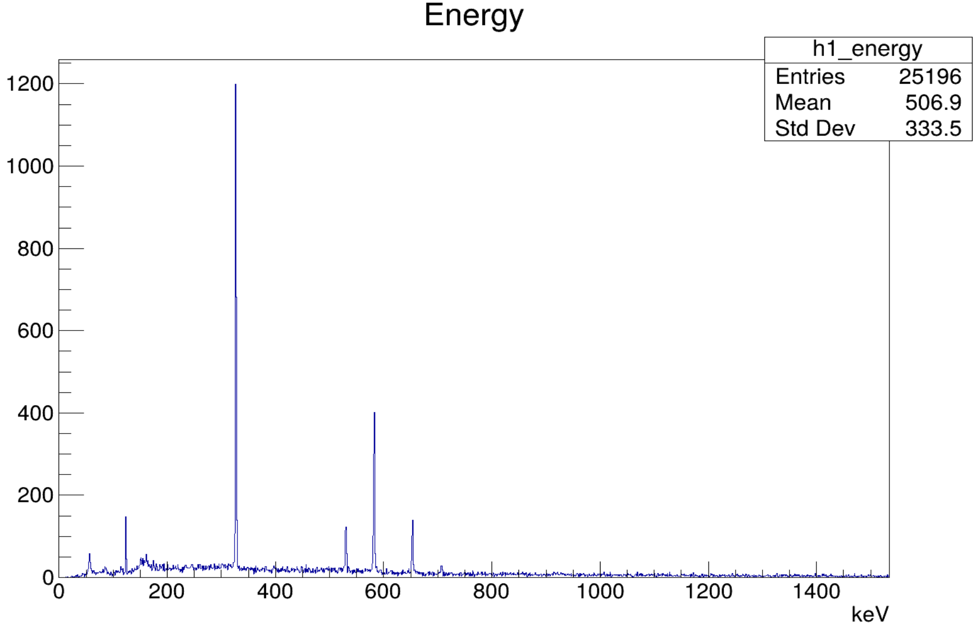
Gamma spectrum

AGATA Data

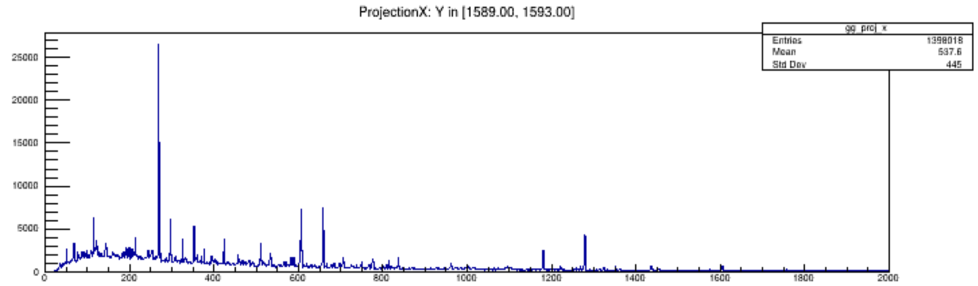
^{98}Zr Peak Energies:
268 keV
620 keV
1223 keV
1591 keV

^{90}Br Peak Energies:
123 keV
327 keV
530 keV
582 keV
653 keV

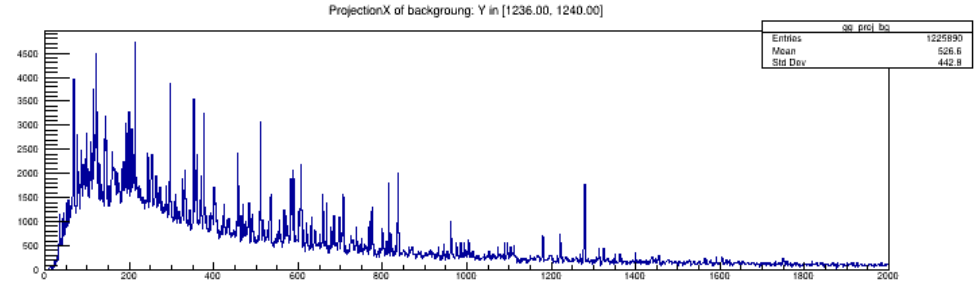




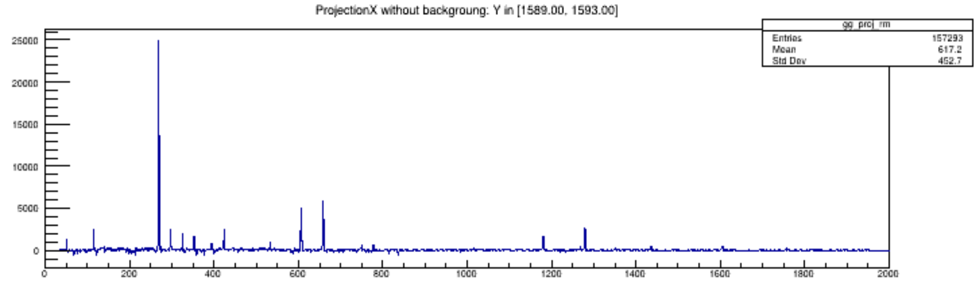
EXILL: ^{98}Zr γ spectrum in coincidence with the γ-ray at 1591 keV


ProjectionX: Y in [1589.00, 1593.00]

- Full spectrum

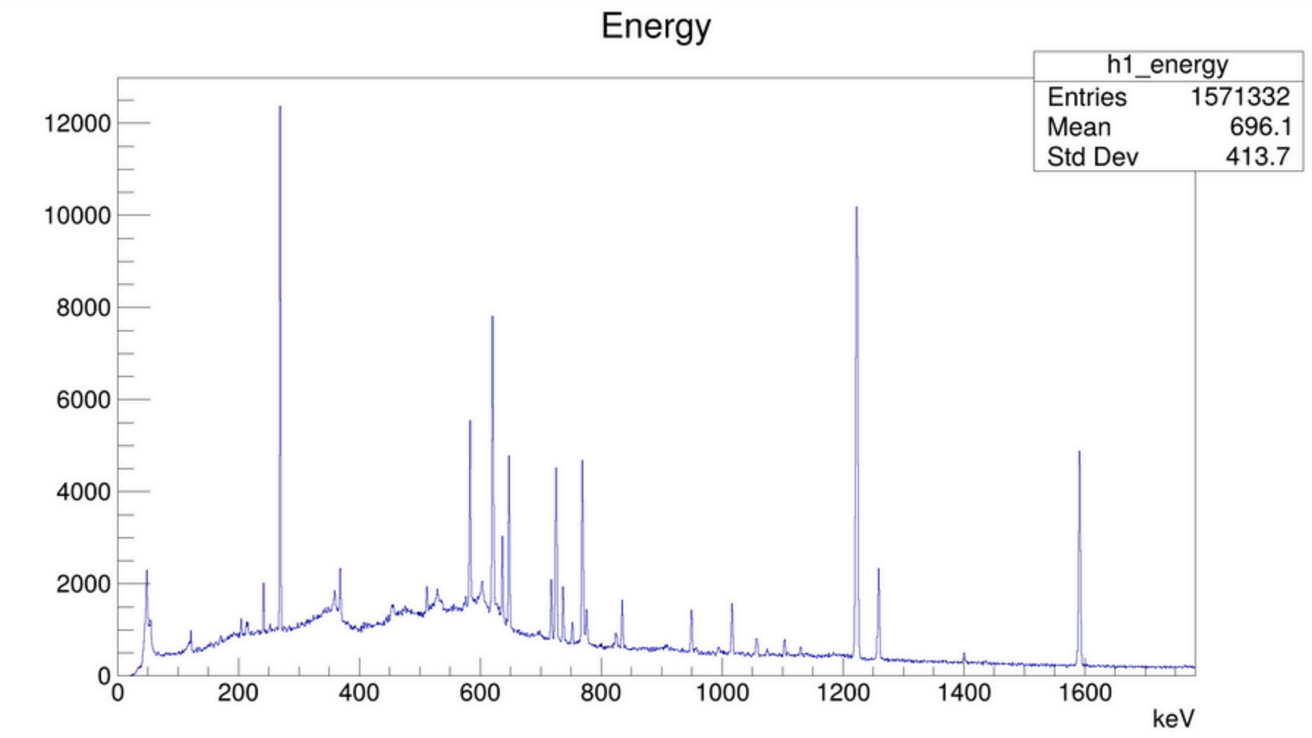

ProjectionX of backgroung: Y in [1236.00, 1240.00]

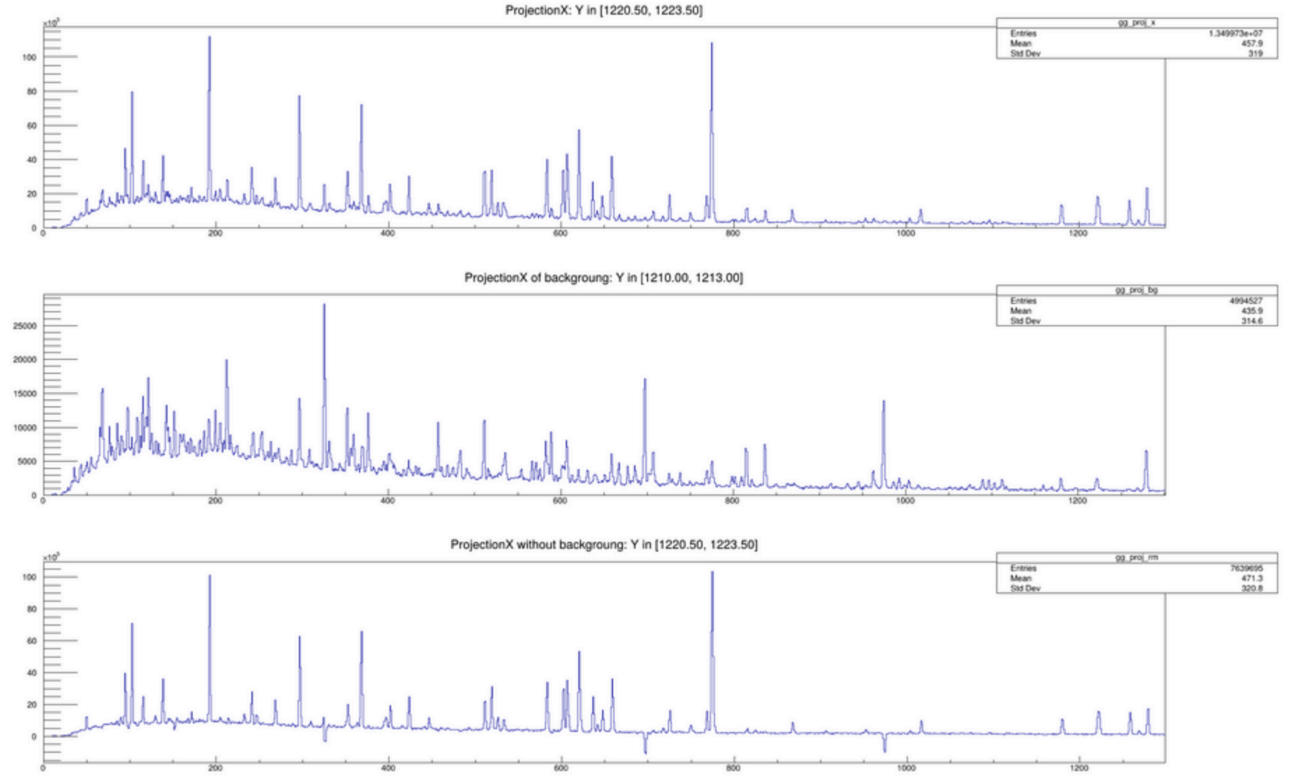
- Background

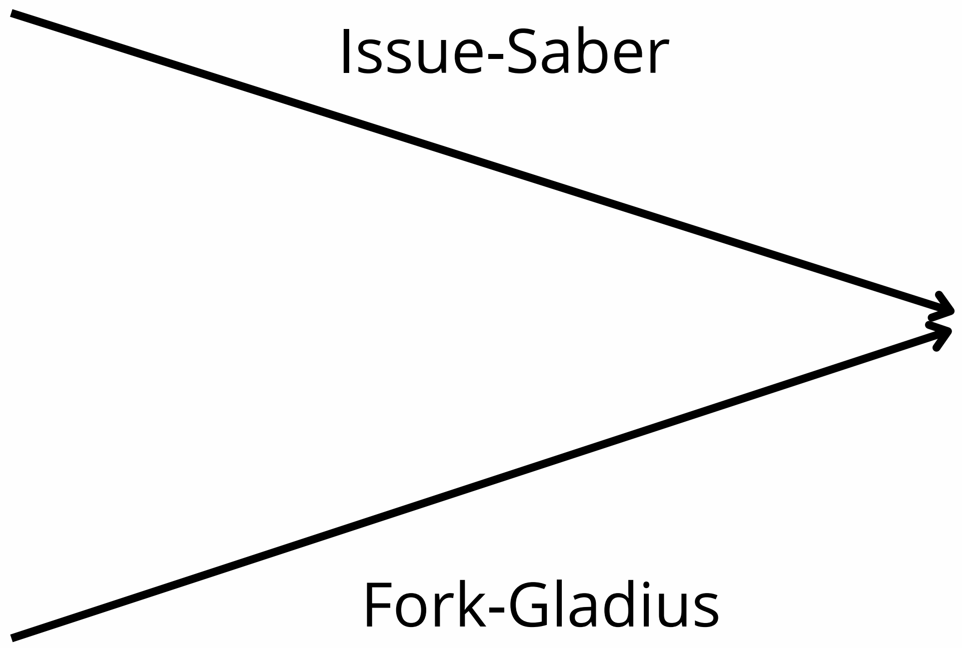

ProjectionX without backgroung: Y in [1589.00, 1593.00]

- Background-subtracted spectrum

FINAL RESULTS

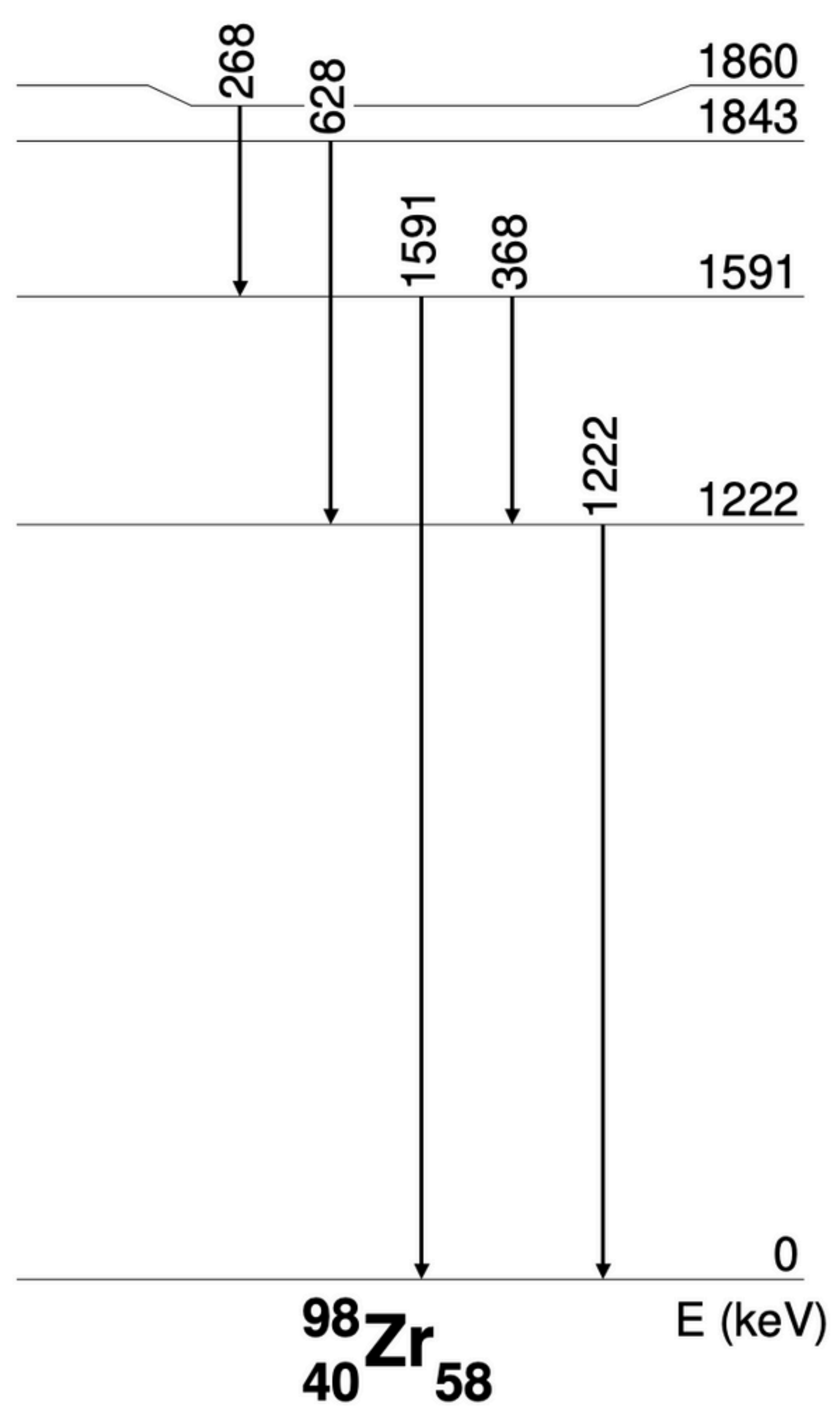






Issue-Saber

Fork-Gladius



Level scheme