Belle II Software Training

Data analysis training in HEP experiments meeting

Kilian Lieret
for the Belle II *Documentation, Training & Software outreach WG*

Sep 21, 2021

PRINCETON UNIVERSITY

Let’s take a look!

training.belle.org

current release:
chapter 21

basf2

e78d2af5

Search docs

» 3. Beginners' tutorials View page source

3. Beginners' tutorials

This online textbook aims to help new Belle II members to get started with the software by following through a series of hands-on lessons.

We want YOU to contribute to this book!

Tip

Just as there are many versions of the Belle 2 software, there are many versions of this documentation to match it. After all, if a new feature is added in our software, we also want to have the documentation for it.

The current version of this documentation is shown on the top left of this page, just below the logo. You can also change your version by clicking on other versions.

If you are a new to all of this, we recommend you to select the *recommended release version* (`release-xx-xx-xx (recommended)` in the above list).

You can also take a sneak peek at the most recent version of the documentation by selecting the development version. However not all of the code examples might work for you yet.

The earliest release version which contains this online book is `release-05-01-12`.

Warning

If you change the version to an earlier version than the current one, some pages (also this page!) might not exist.

Some stats

Everything you need to get started

244 code snippets

```
# ROE variables
roe_kinematics = ["roeE()", "roeM()", "roeP()", "roeMbc()", "roeDeltae()"]
roe_multiplicities = [
    "nROE_Charged()",
    "nROE_Photons()",
    "nROE_NeutralHadrons()",
]
b_vars += roe_kinematics + roe_multiplicities
```

45 figures

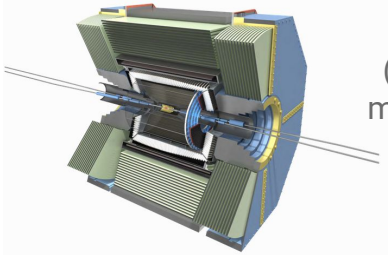
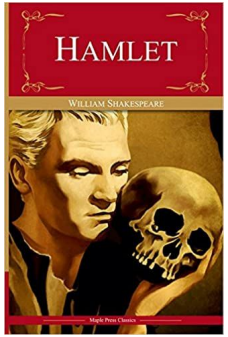


Fig. 21.3 The Belle II detector.

2.5x Hamlets (excluding most code)



57x keeping an overview

Key points

- The ROE of a selection is build with `buildRestOfEvent`
- ROE masks are added with `appendROEMask` or `appendRO`
  beam-induced or other background particles.
- For many analyses ROE is used as middleware to get
  or flavor tag.
- Usage of ROE without a mask is **not** recommended.

Overview

Teaching: 10-20 min

Exercises: 10-20 min

Prerequisites:

- The previous lesson

Questions:

- I combined several particles into `x`. How do I select everything that is not "in" `x`?
- How to exclude some particles from this Rest of Event / what is an ROE mask?

meson

Foldable additional information:

Strange variable names ▼

212 exercises 159 hints, 191 solutions

Task

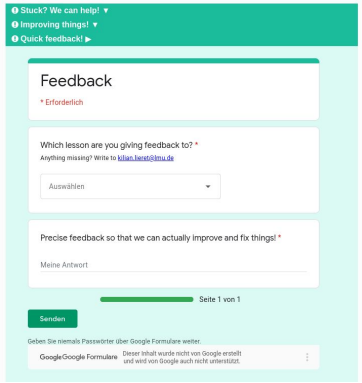
Add PID and track variables for all charged final state particles and the invariant mass of the intermediate resonances to the ntuple. Also add the standard variables from before for all particles in the decay chain, the kinematics both in the lab and the CMS frame.

- Hint: Where to look ▼
- Hint: Partial solution for final state particles ▼
- Hint: CMS variables ▼
- Hint: Partial solution for the CMS variables ▼
- Solution ▼

Challenges

Challenge	Solution	
COVID, timezones, different schedules, make it hard to offer events at the right time & level	Focus on **self-study** ready training material. Event events focus on self-study time.	✔
Many scattered resources; lack of red thread	Beginner's tutorials on sphinx as central entry point	✔
New versions of software break examples	• everything is plain text and bundled together with software • code is included from source files • source files are executed as unit tests	✔
Avoid duplication of documentation	Easy to link to existing API documentation as both lives on Sphinx	✔
Let's keep it fun for students	• Lot's of exercises • full solutions • many partial hints	✔

Challenges

Challenge	Solution	
Let's keep track of issues	Everything on JIRA	✔
Need to get feedback from students	• Minimal feedback form at the bottom of each page • Can’t think of anything simpler • Still barely any reports • 👉 Encourage your students! 	✔ 🥺
Grow number of contributors	• Provide more recognition (authorship on talks & papers) • Clearer step by step instructions for contributing • Working on documentation and training counts as service task!	🚧 🥺

Let’s get technical: rst

- The online book is built with Sphinx
- We have documentation on how to documentation 😉
- Everything is a plain-text file in rst format
- rst takes only a couple of minutes to learn

```
.. admonition:: Question
   :class: exercise stacked

   What's the object-oriented way to get rich?

.. admonition:: Hint
   :class: toggle xhint stacked

   Think about relationships between classes!

.. admonition:: Solution
   :class: toggle solution

   Inheritance.
```

Question

What’s the object-oriented way to get rich?

Hint ▼

Solution ▼

```
.. admonition:: Key points
    :class: key-points

    * There are 10 kinds of people in this world:
      Those who understand binary, those who don't,
      and those who weren't expecting a base 3 joke.
```

Key points

- There are 10 kinds of people in this world: Those who understand binary, those who don’t, and those who weren’t expecting a base 3 joke.

Let’s get technical: including code

Want to always

- Give full code as solution in the end
- Explain code with small snippet
- Include partial solution in hints

But duplicating parts of code is bad!

- Put code in separate code files (ideally run as unit test)
- Use partial code inclusion in hints or explanations
- Previously: Used line number based partial code inclusion (but terrible for maintaining)
- Now: Using markers in code

Exercise

Look up how to create a Rest Of Event for your particle list (it's a single line of code). Add the `fillWithMostLikely=True` option.

Hint ▼

Solution ▶

```
# build the rest of the event [S10|S30|S40]
ma.buildRestOfEvent("B0", fillWithMostLikely=True, path=main)  # [E10]
```

```
.. admonition:: Solution
    :class: toggle solution

    .. literalinclude:: steering_files/0
        :start-at: S10
        :end-at: E10
```

Summary

- Our training (and our software) is **open source**: https://training.belle.org and https://github.com/belle2/basf2

- More information on our ACAT poster, proceedings “A new software training model at Belle II” soon

Advertisement: HSF Training group

- Is there any part in your training that is **experiment independent**? (bash, python, git, docker, ML, …)
- **Wouldn't it be more efficient to teach/maintain/build that as a large cross-experiment community?**
- The HSF Training WG
  - coordinates shared training efforts
  - organizes regular trainings
  - fosters the development of new training modules that can be taught by *anyone*
- Everything is open source, most of it on our GitHub organization
- Check out the available modules in the HSF Training Center
- Recent development: monthly hackathons, for example on our C++ course.