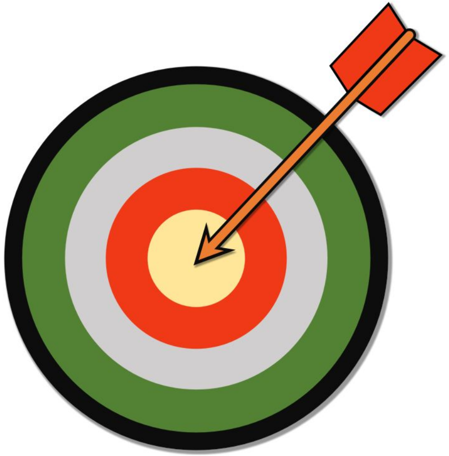
Analysis Skims

General Overview

Trevor Shillington,
on behalf of the Data Production group

2023 Belle II Summer Workshop @ Duke University

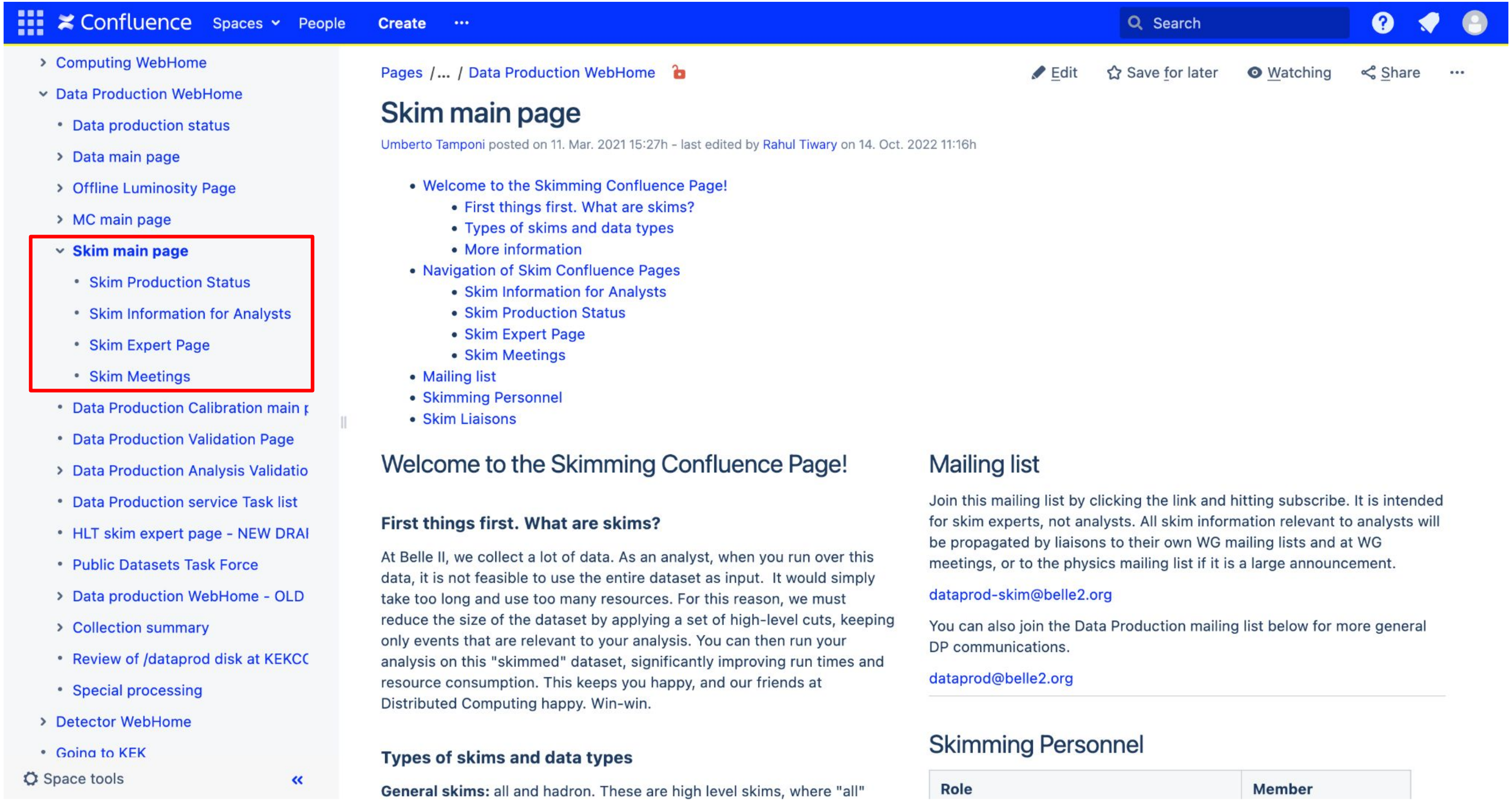
Skimming



Primary goal:

Centrally produce analyst friendly data and MC

Skim Confluence page


Confluence
Spaces
People
Create
Search
Computing WebHome
Data Production WebHome
Data production status
Data main page
Offline Luminosity Page
MC main page
Skim main page
Skim Production Status
Skim Information for Analysts
Skim Expert Page
Skim Meetings
Data Production Calibration main p
Data Production Validation Page
Data Production Analysis Validatio
Data Production service Task list
HLT skim expert page - NEW DRAI
Public Datasets Task Force
Data production WebHome - OLD
Collection summary
Review of /dataprod disk at KEKCC
Special processing
Detector WebHome
Going to KEK
Space tools
Pages / ... / Data Production WebHome
Edit
Save for later
Watching
Share

Skim main page

Umberto Tamponi posted on 11. Mar. 2021 15:27h - last edited by Rahul Tiwary on 14. Oct. 2022 11:16h

- Welcome to the Skimming Confluence Page!
  - First things first. What are skims?
  - Types of skims and data types
  - More information
- Navigation of Skim Confluence Pages
  - Skim Information for Analysts
  - Skim Production Status
  - Skim Expert Page
  - Skim Meetings
- Mailing list
- Skimming Personnel
- Skim Liaisons

Welcome to the Skimming Confluence Page!

First things first. What are skims?

At Belle II, we collect a lot of data. As an analyst, when you run over this data, it is not feasible to use the entire dataset as input. It would simply take too long and use too many resources. For this reason, we must reduce the size of the dataset by applying a set of high-level cuts, keeping only events that are relevant to your analysis. You can then run your analysis on this "skimmed" dataset, significantly improving run times and resource consumption. This keeps you happy, and our friends at Distributed Computing happy. Win-win.

Types of skims and data types

General skims: all and hadron. These are high level skims, where "all"

Mailing list

Join this mailing list by clicking the link and hitting subscribe. It is intended for skim experts, not analysts. All skim information relevant to analysts will be propagated by liaisons to their own WG mailing lists and at WG meetings, or to the physics mailing list if it is a large announcement.

dataprod-skim@belle2.org

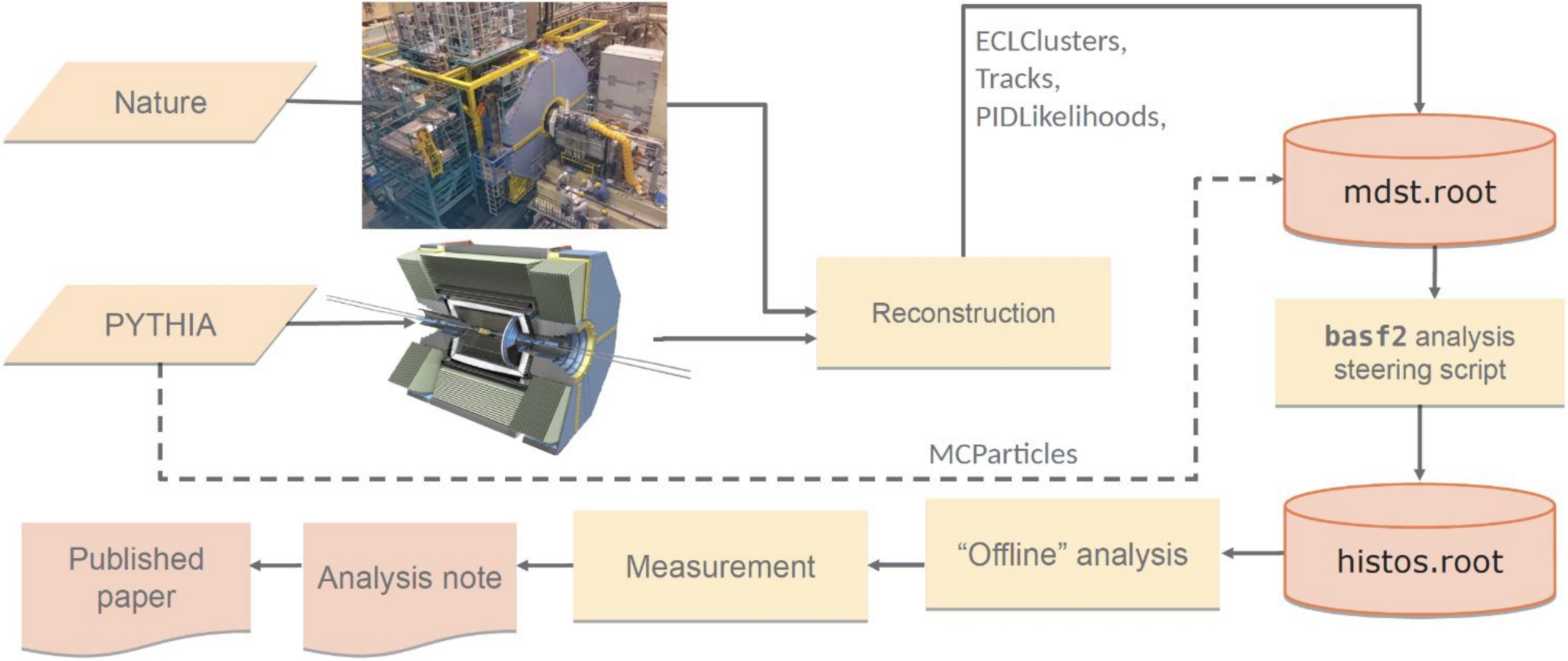
You can also join the Data Production mailing list below for more general DP communications.

dataprod@belle2.org

Skimming Personnel

Role	Member

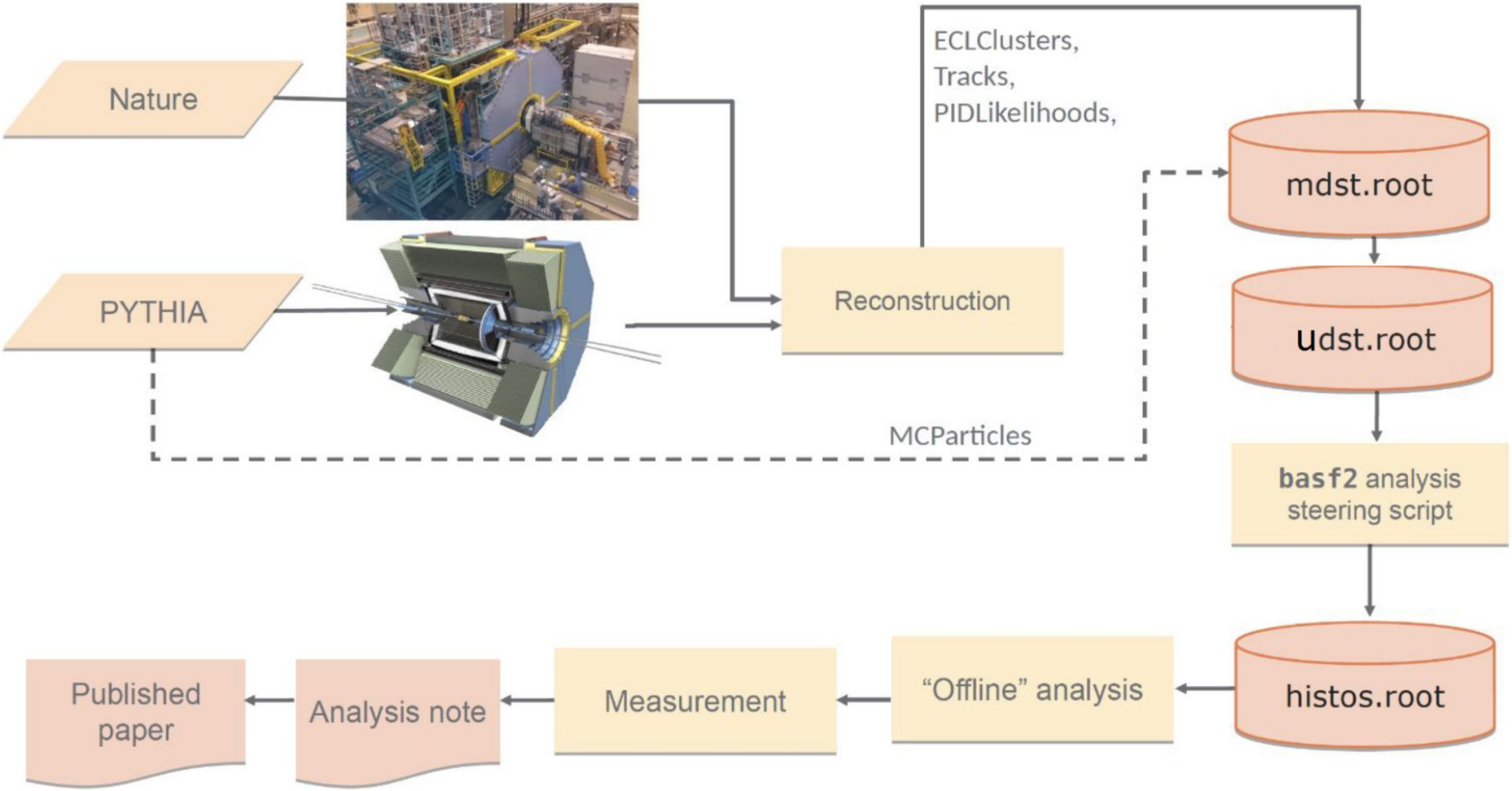
The big picture



*~~shamelessly stolen~~ borrowed from Sam Cunliffe's talk"Introduction to the analysis package" - Belle II SKW, 15.06.2018

*~~shamelessly stolen again~~ borrowed from Jake Bennett's DP talk at 2022 Belle II Summer Workshop

The big picture



Nature

PYTHIA

Reconstruction

ECLClusters,
Tracks,
PIDLikelihoods,

MCParticles

mdst.root

udst.root

basf2 analysis steering script

histos.root

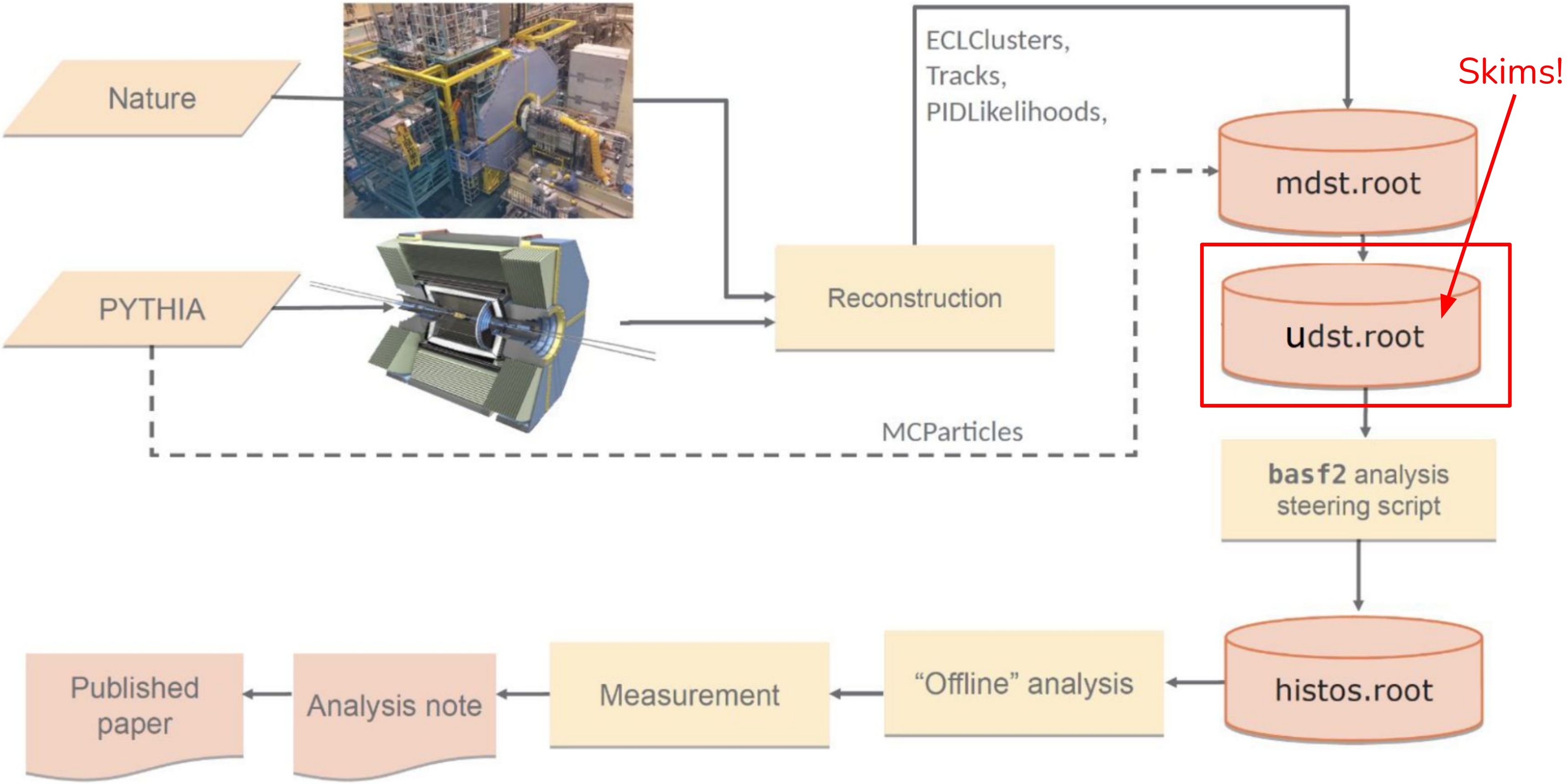
“Offline” analysis

Measurement

Analysis note

Published paper

The big picture



Nature

PYTHIA

Reconstruction

ECLClusters,
Tracks,
PIDLikelihoods,

MCParticles

mdst.root

udst.root

Skims!

basf2 analysis steering script

histos.root

"Offline" analysis

Measurement

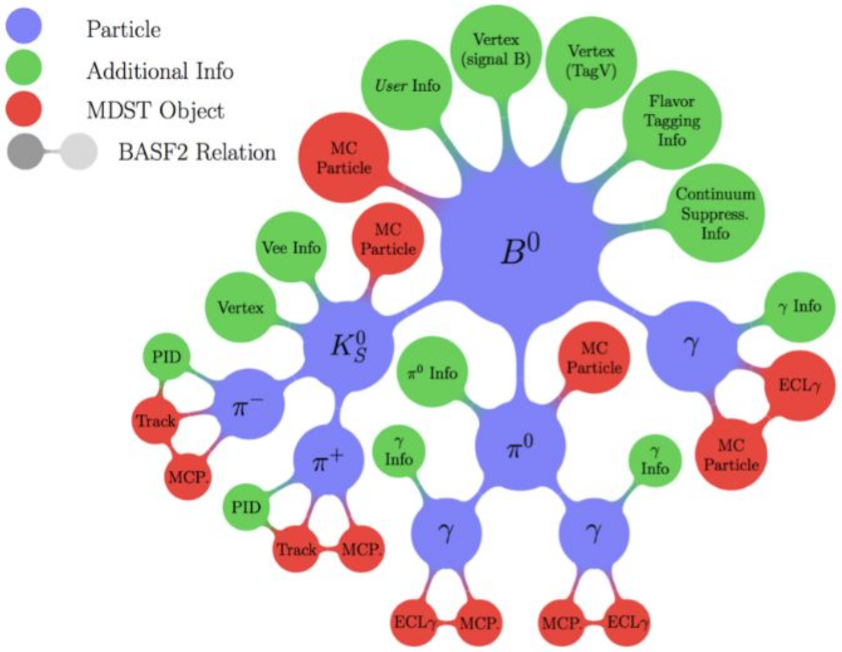
Analysis note

Published paper

Analysis Skims

Skims are meant to provide analysis-oriented MC and data in reduced sizes

- Produced as **udst**, using mdst as input
  - udst = mdst information + analysis-level information
- Analysis-level information:
  - Particle objects
  - Vertex fit results (covariance matrices)
  - Information from full B and D reconstruction
  - Continuum suppression
  - & other complex algorithms
- **That means more information in smaller files!**



Result: Faster processing for analysts ⚡

Image from Jake's DP talk last year

Analysis Skims

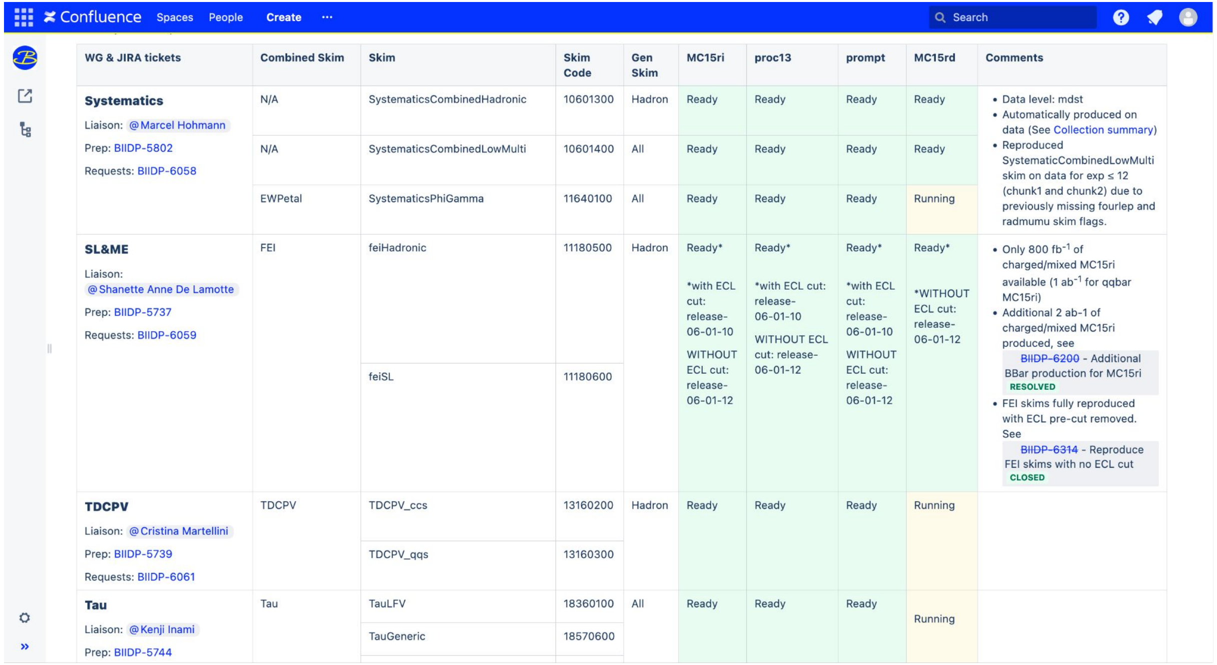
Skims are defined in specially formatted python scripts that are essentially just basf2 steering files, written by analysts or working groups to meet their specific physics needs.

There are some requirements the skim group imposes to allow for successful grid production:

- Event retention must be <10% (with few exceptions)
- Processing time must be <2.95 HS06 per event
- Log files must be < 30 MB
- Maximum memory usage cannot exceed 2 GB
- Avg. candidate multiplicity should be below 20

Analysis Skims - Current status

- ~60 analysis skims produced this campaign (MC15/proc13)
- Fully available for data (proc13+buckets) and MC15ri
- MC15rd partially available (almost done!)
- Check Confluence for production status



WG & JIRA tickets	Combined Skim	Skim	Skim Code	Gen Skim	MC15ri	proc13	prompt	MC15rd	Comments
Systematics Liaison: @Marcel Hohmann Prep: BIIDP-5802 Requests: BIIDP-6058	N/A	SystematicsCombinedHadronic	10601300	Hadron	Ready	Ready	Ready	Ready	• Data level: mdst • Automatically produced on data (See Collection summary) • Reproduced SystematicCombinedLowMulti skim on data for exp ≤ 12 (chunk1 and chunk2) due to previously missing fourlep and radmumu skim flags.
	N/A	SystematicsCombinedLowMulti	10601400	All	Ready	Ready	Ready	Ready	
	EWPetal	SystematicsPhiGamma	11640100	All	Ready	Ready	Ready	Running	
SL&ME Liaison: @Shanette Anne De Lamotte Prep: BIIDP-5737 Requests: BIIDP-6059	FEI	feiHadronic	11180500	Hadron	Ready* *with ECL cut: release-06-01-10 WITHOUT ECL cut: release-06-01-12	Ready* *with ECL cut: release-06-01-10 WITHOUT ECL cut: release-06-01-12	Ready* *with ECL cut: release-06-01-10 WITHOUT ECL cut: release-06-01-12	Ready* *WITHOUT ECL cut: release-06-01-12	• Only 800 fb^{-1} of charged/mixed MC15ri available (1 ab^{-1} for qqbar MC15ri) • Additional 2 ab-1 of charged/mixed MC15ri produced, see ~~BIIDP-6200~~ - Additional BBar production for MC15ri RESOLVED • FEI skims fully reproduced with ECL pre-cut removed. See ~~BIIDP-6314~~ - Reproduce FEI skims with no ECL cut CLOSED
		feiSL	11180600						
TDCPV Liaison: @Cristina Martellini Prep: BIIDP-5739 Requests: BIIDP-6061	TDCPV	TDCPV_ccs	13160200	Hadron	Ready	Ready	Ready	Running	
		TDCPV_qqs	13160300						
Tau Liaison: @Kenji Inami Prep: BIIDP-5744	Tau	TauLFV	18360100	All	Ready	Ready	Ready	Running	
		TauGeneric	18570600						

A quick note on “Combined Skims”

From a production standpoint, we use something called **Combined Skims**

As the name implies, this is a group of skims (typically 5-15) that are combined into a single production in order to ease the number of productions required in a campaign.

Nothing is affected in terms of output (each skim still has their own separate output), they just share a production number.

You can imagine of the number of productions needed if we submitted every skim individually. For example, for MC15rd:

- 60 skims
- 14 MCtypes
- 14 experiments
- 60*14*14 = ~12k productions

Skim names and skim codes

- Each skim is given a plain language name, and an eight digit skim code.
- There is a method for devising a skim code, based on the physics of the skim*.
- You can find the skim code on Confluence, Sphinx, or in registry.py on gitlab

Some examples:

Skim Name	Skim Code
`feiSL`	`11180600`
`TDCPV_ccs`	`13160200`
`TauLFV`	`18360100`

* https://confluence.desy.de/display/BI/Skim+Expert+Page

bucket30 skim disk usage

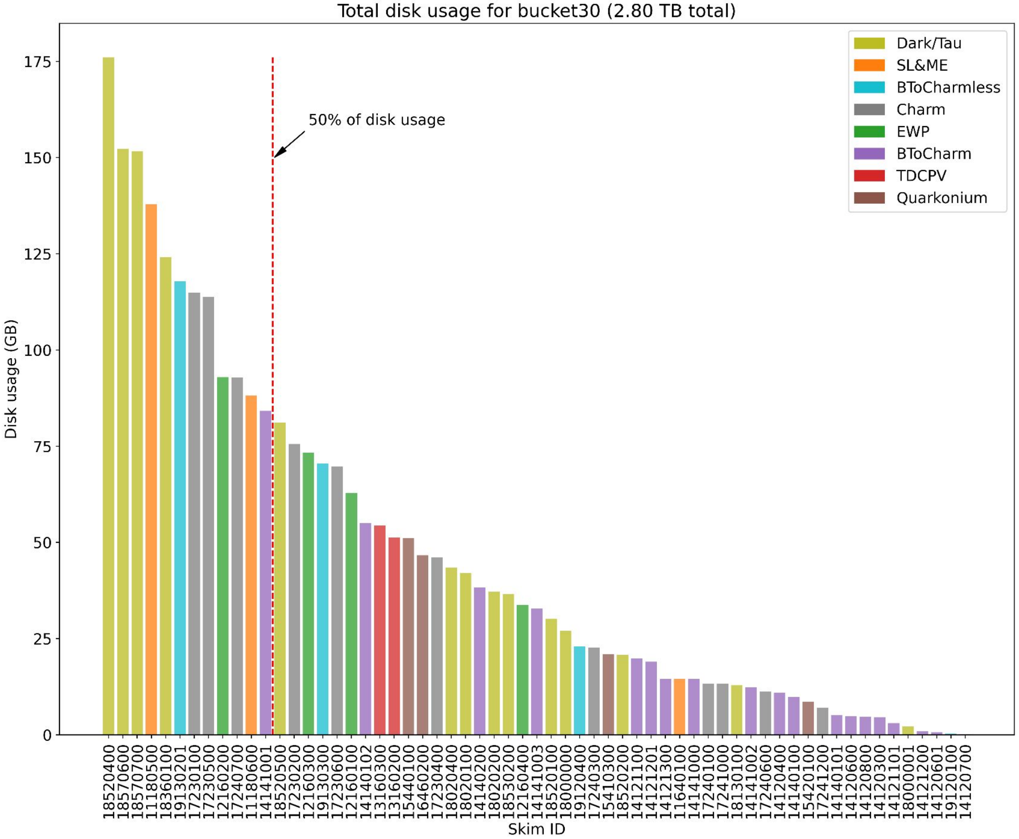
mdst numbers for comparison:

bucket30 (all): **1.5 TB**

bucket30 (hadron): **670 GB**

Notice:

- Even the biggest skims are ~10% of mdst file size.
- Many skims are < 5%
- WG strategies differ: some have a few large skims, some have many small skims



Understanding skims

There are two primary sources for information on what cuts & selection skims make, as well as what information they output:

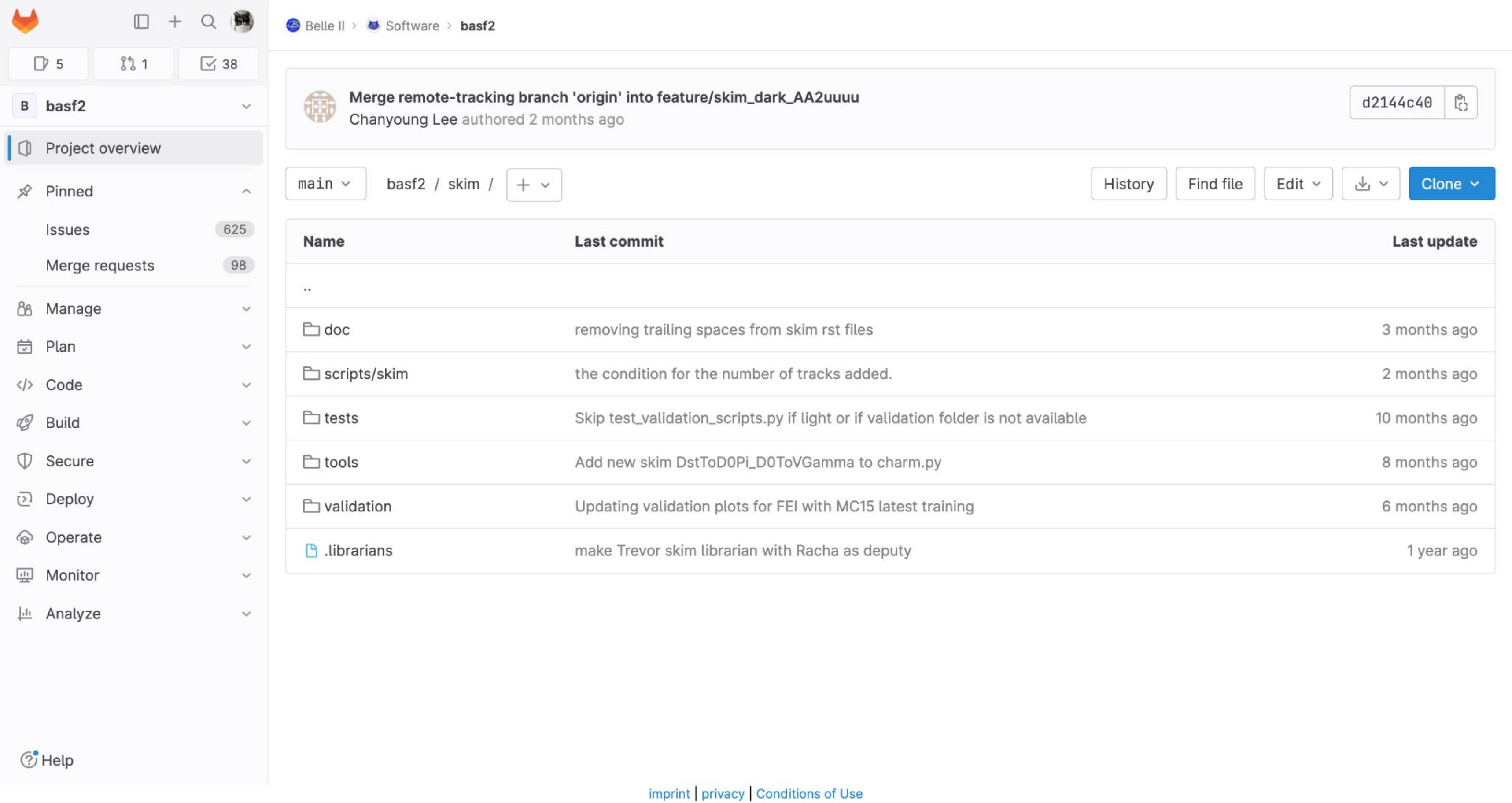
1. **Source code on gitlab**
    1. **Best - full information and always up to date**
    2. **Easily check specific releases**

2. **Sphinx Documentation**
    1. **Good for general information and tutorials**

It is important for analysts to understand the skim they are using to make sure that it is aligned with their analysis goals.

We will now do a quick walk through of skim content on **gitlab** and **Sphinx**

Skim package - gitlab

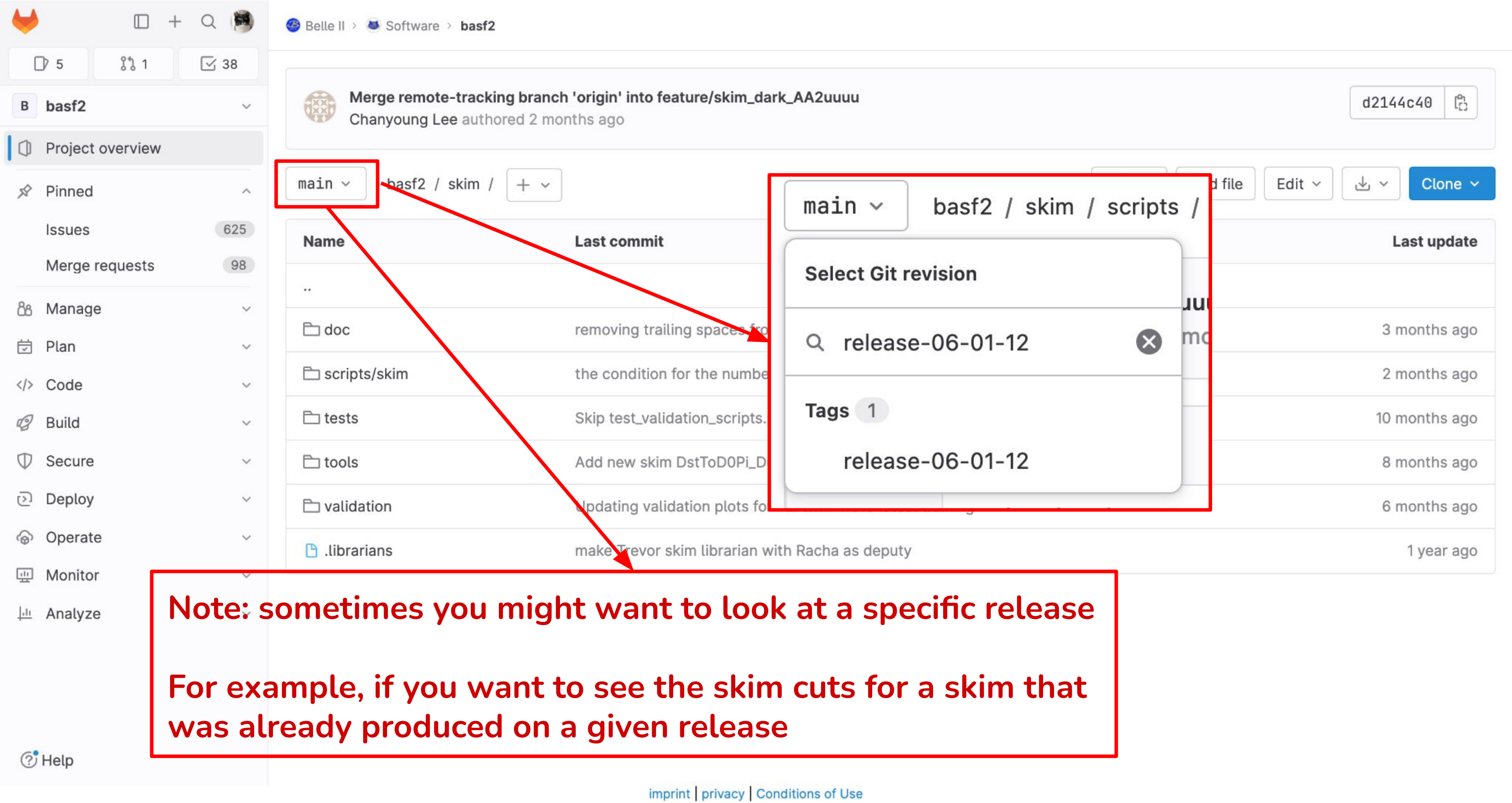


Merge remote-tracking branch 'origin' into feature/skim_dark_AA2uuuu
Chanyoung Lee authored 2 months ago

d2144c40

Name	Last commit	Last update
..		
doc	removing trailing spaces from skim rst files	3 months ago
scripts/skim	the condition for the number of tracks added.	2 months ago
tests	Skip test_validation_scripts.py if light or if validation folder is not available	10 months ago
tools	Add new skim DstToD0Pi_D0ToVGamma to charm.py	8 months ago
validation	Updating validation plots for FEI with MC15 latest training	6 months ago
.librarians	make Trevor skim librarian with Racha as deputy	1 year ago

Skim package - gitlab



Merge remote-tracking branch 'origin' into feature/skim_dark_AA2uuuu
Chanyoung Lee authored 2 months ago

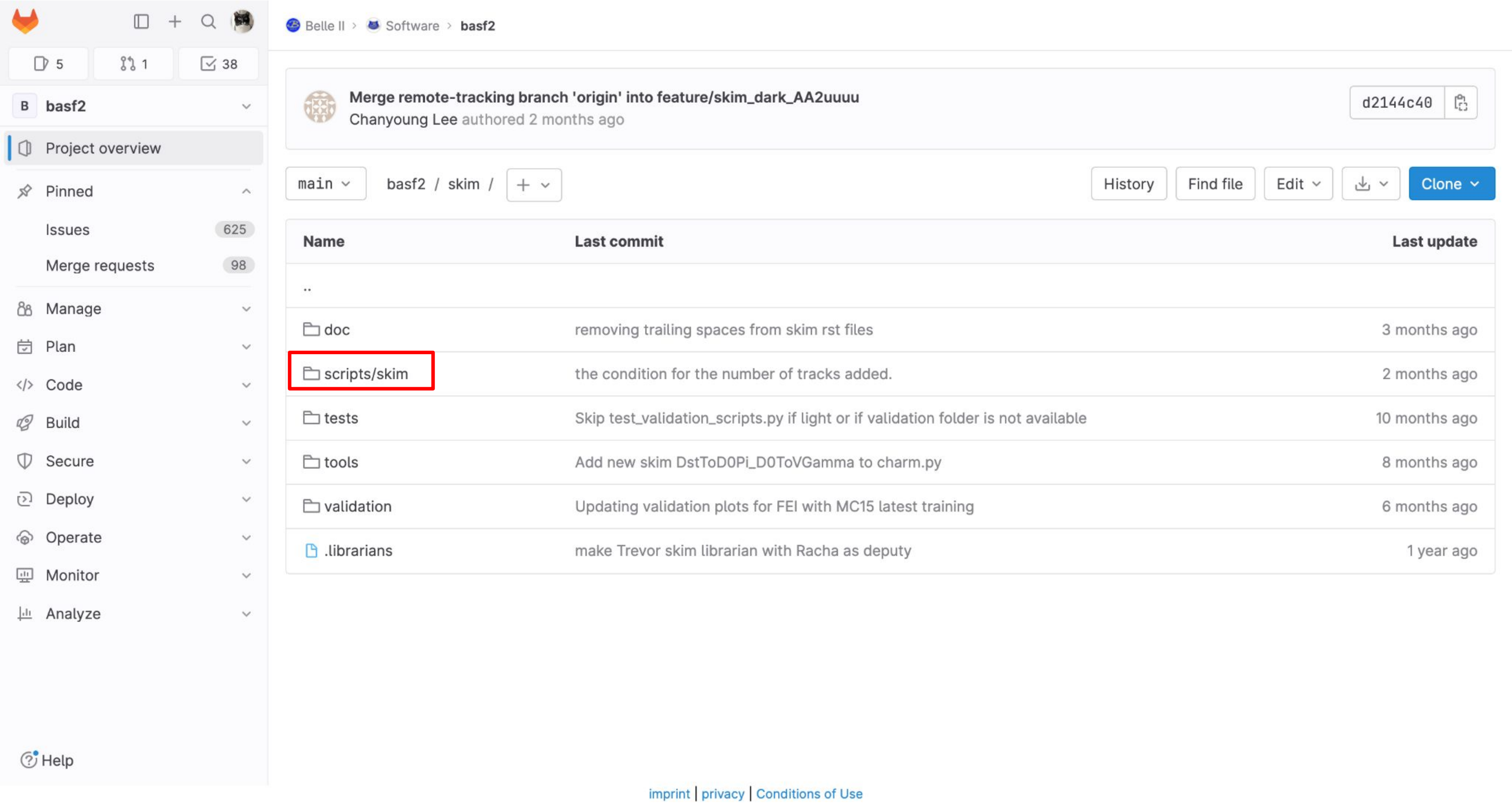
d2144c40

Name	Last commit	Last update
..		
doc	removing trailing spaces fro	3 months ago
scripts/skim	the condition for the numbe	2 months ago
tests	Skip test_validation_scripts.	10 months ago
tools	Add new skim DstToD0Pi_D	8 months ago
validation	Updating validation plots fo	6 months ago
.librarians	make Trevor skim librarian with Racha as deputy	1 year ago

Note: sometimes you might want to look at a specific release

For example, if you want to see the skim cuts for a skim that was already produced on a given release

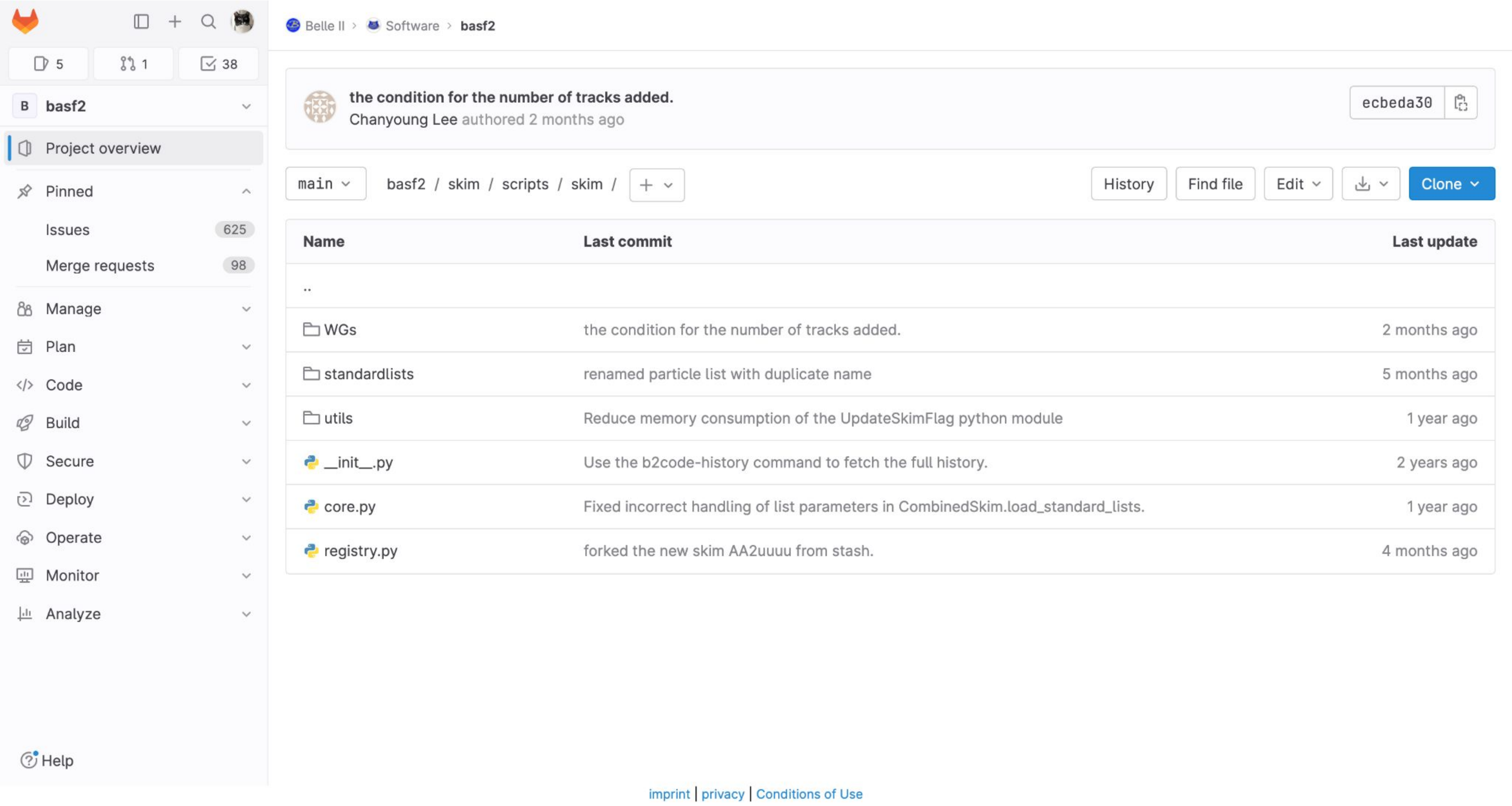
Skim package - gitlab



Merge remote-tracking branch 'origin' into feature/skim_dark_AA2uuuu
Chanyoung Lee authored 2 months ago

Name	Last commit	Last update
..		
doc	removing trailing spaces from skim rst files	3 months ago
scripts/skim	the condition for the number of tracks added.	2 months ago
tests	Skip test_validation_scripts.py if light or if validation folder is not available	10 months ago
tools	Add new skim DstToD0Pi_D0ToVGamma to charm.py	8 months ago
validation	Updating validation plots for FEI with MC15 latest training	6 months ago
.librarians	make Trevor skim librarian with Racha as deputy	1 year ago

Skim package - gitlab



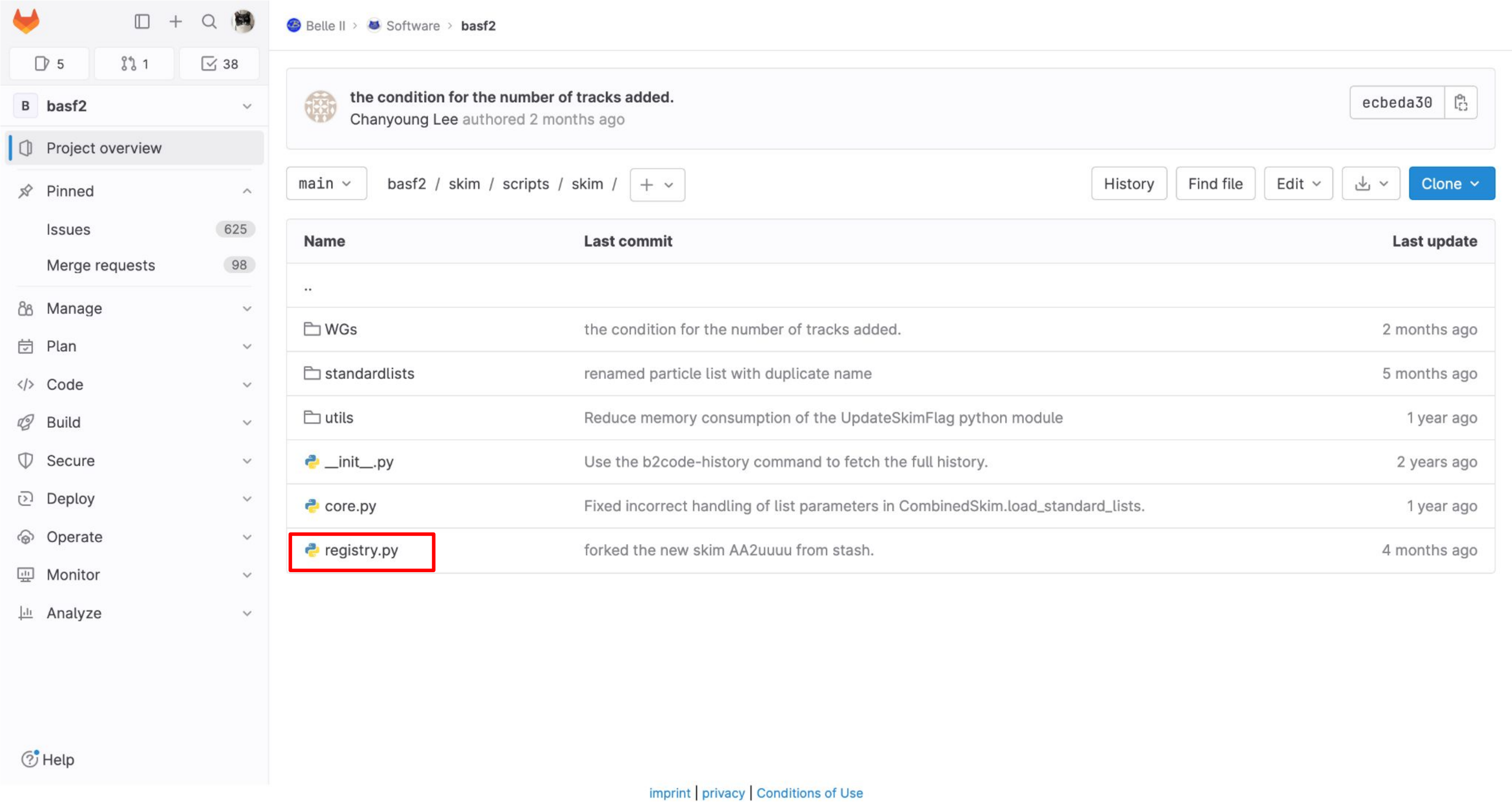
the condition for the number of tracks added.
Chanyoung Lee authored 2 months ago

ecbeda30

basf2 / skim / scripts / skim /

Name	Last commit	Last update
..		
WGs	the condition for the number of tracks added.	2 months ago
standardlists	renamed particle list with duplicate name	5 months ago
utils	Reduce memory consumption of the UpdateSkimFlag python module	1 year ago
__init__.py	Use the b2code-history command to fetch the full history.	2 years ago
core.py	Fixed incorrect handling of list parameters in CombinedSkim.load_standard_lists.	1 year ago
registry.py	forked the new skim AA2uuuu from stash.	4 months ago

Skim package - gitlab



the condition for the number of tracks added.
Chanyoung Lee authored 2 months ago

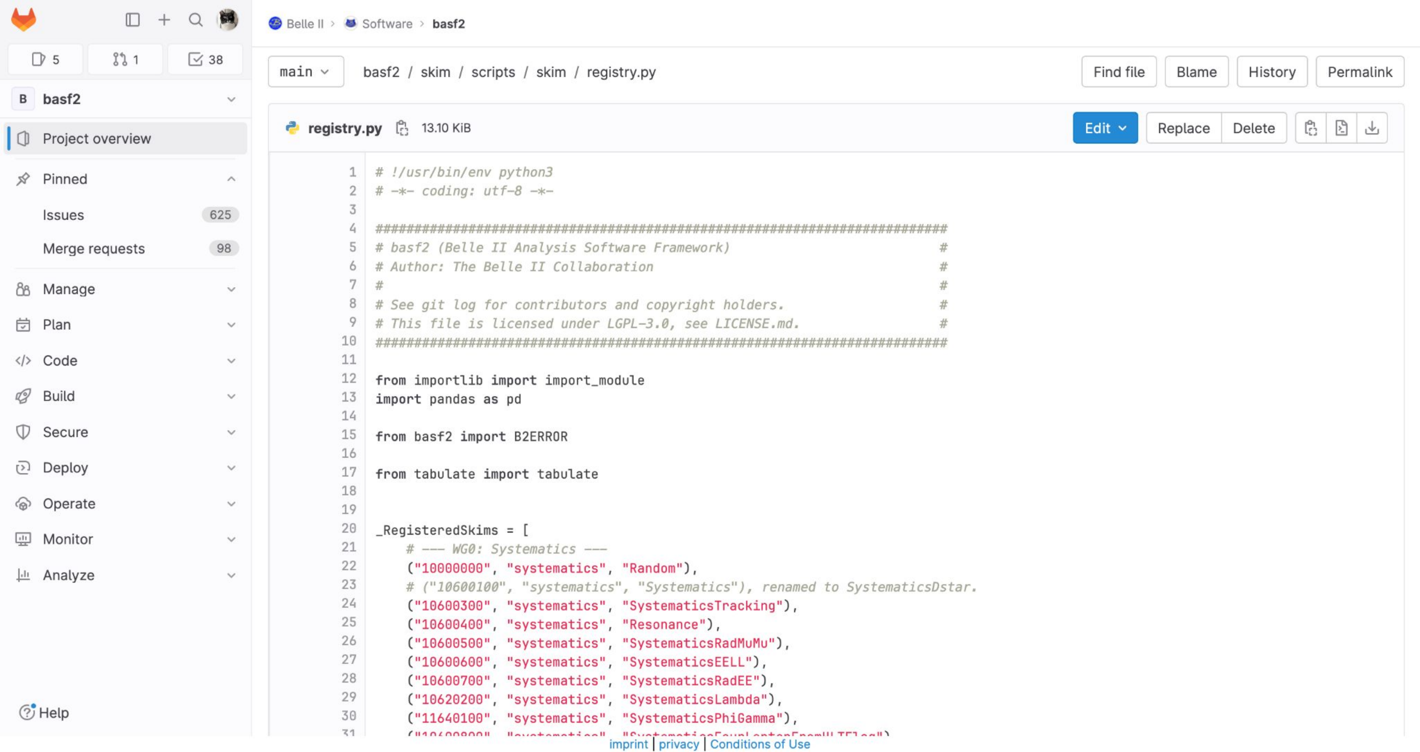
basf2 / skim / scripts / skim /

Name	Last commit	Last update
..		
WGs	the condition for the number of tracks added.	2 months ago
standardlists	renamed particle list with duplicate name	5 months ago
utils	Reduce memory consumption of the UpdateSkimFlag python module	1 year ago
__init__.py	Use the b2code-history command to fetch the full history.	2 years ago
core.py	Fixed incorrect handling of list parameters in CombinedSkim.load_standard_lists.	1 year ago
registry.py	forked the new skim AA2uuuu from stash.	4 months ago

Registry.py

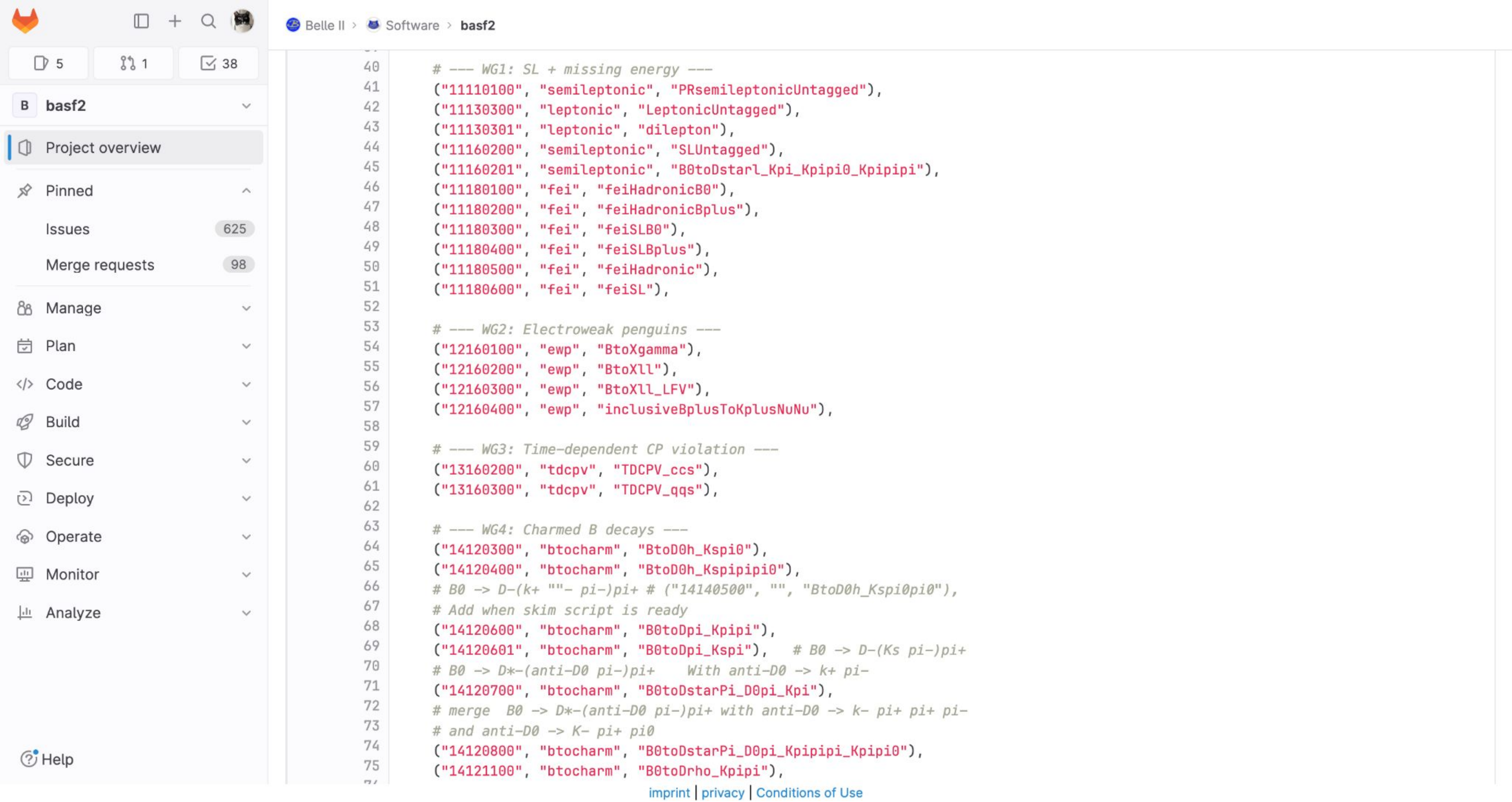
All skims are registered with their skim code in our skim registry.

Good place to look up skim codes and see what skims exist.



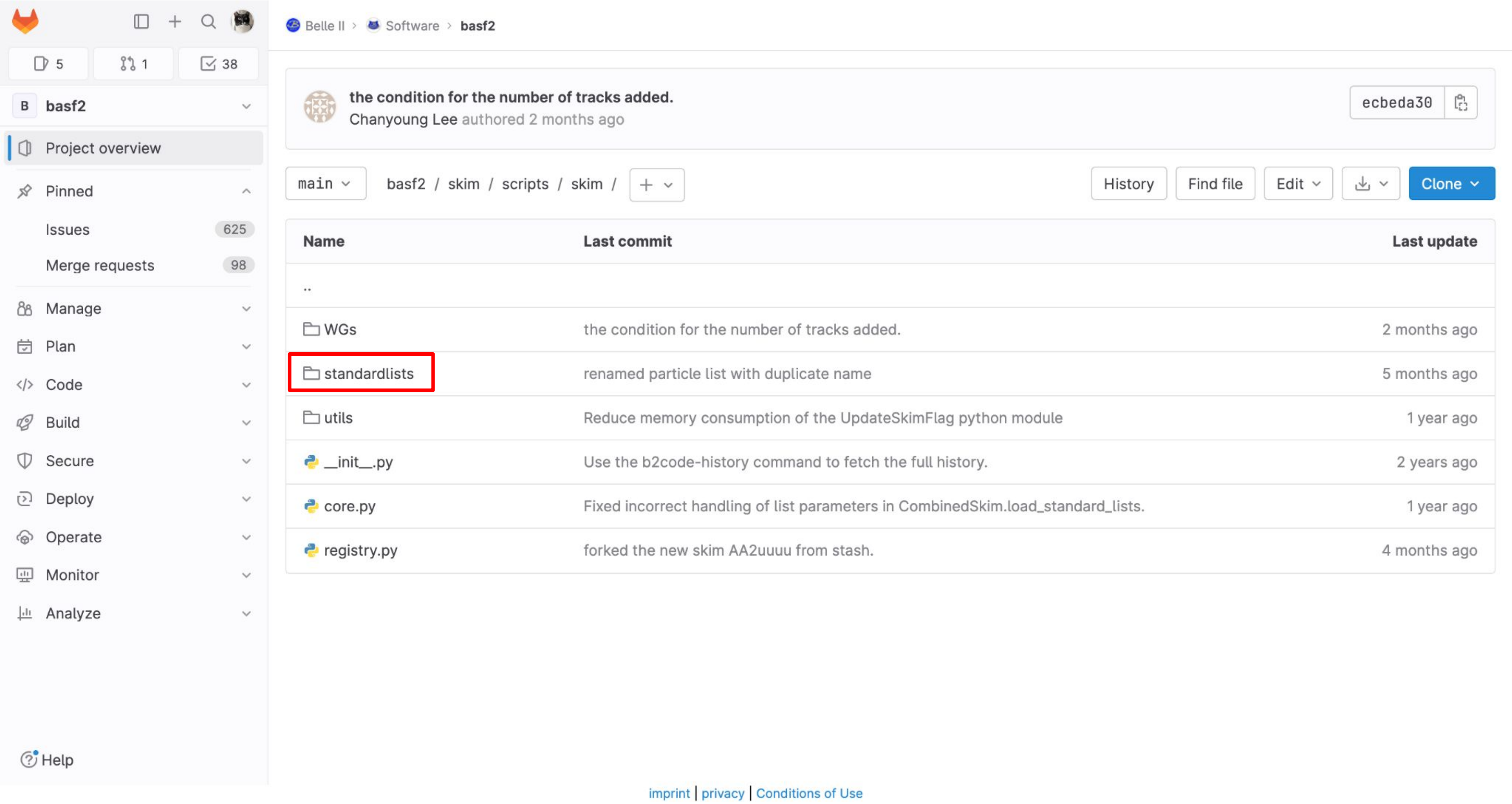
```python
# !/usr/bin/env python3
# -*- coding: utf-8 -*-

##########################################################################
# basf2 (Belle II Analysis Software Framework)                           #
# Author: The Belle II Collaboration                                     #
#                                                                        #
# See git log for contributors and copyright holders.                    #
# This file is licensed under LGPL-3.0, see LICENSE.md.                  #
##########################################################################

from importlib import import_module
import pandas as pd

from basf2 import B2ERROR

from tabulate import tabulate


_RegisteredSkims = [
    # --- WG0: Systematics ---
    ("10000000", "systematics", "Random"),
    # ("10600100", "systematics", "Systematics"), renamed to SystematicsDstar.
    ("10600300", "systematics", "SystematicsTracking"),
    ("10600400", "systematics", "Resonance"),
    ("10600500", "systematics", "SystematicsRadMuMu"),
    ("10600600", "systematics", "SystematicsEELL"),
    ("10600700", "systematics", "SystematicsRadEE"),
    ("10620200", "systematics", "SystematicsLambda"),
    ("11640100", "systematics", "SystematicsPhiGamma"),
    ("10600800", "systematics", "SystematicsFourLeptonFromHLTFlag"),
```

Registry.py



```
# --- WG1: SL + missing energy ---
("11110100", "semileptonic", "PRsemileptonicUntagged"),
("11130300", "leptonic", "LeptonicUntagged"),
("11130301", "leptonic", "dilepton"),
("11160200", "semileptonic", "SLUntagged"),
("11160201", "semileptonic", "B0toDstarl_Kpi_Kpipi0_Kpipipi"),
("11180100", "fei", "feiHadronicB0"),
("11180200", "fei", "feiHadronicBplus"),
("11180300", "fei", "feiSLB0"),
("11180400", "fei", "feiSLBplus"),
("11180500", "fei", "feiHadronic"),
("11180600", "fei", "feiSL"),

# --- WG2: Electroweak penguins ---
("12160100", "ewp", "BtoXgamma"),
("12160200", "ewp", "BtoXll"),
("12160300", "ewp", "BtoXll_LFV"),
("12160400", "ewp", "inclusiveBplusToKplusNuNu"),

# --- WG3: Time-dependent CP violation ---
("13160200", "tdcpv", "TDCPV_ccs"),
("13160300", "tdcpv", "TDCPV_qqs"),

# --- WG4: Charmed B decays ---
("14120300", "btocharm", "BtoD0h_Kspi0"),
("14120400", "btocharm", "BtoD0h_Kspipipi0"),
# B0 -> D-(k+ ""- pi-)pi+ # ("14140500", "", "BtoD0h_Kspi0pi0"),
# Add when skim script is ready
("14120600", "btocharm", "B0toDpi_Kpipi"),
("14120601", "btocharm", "B0toDpi_Kspi"),   # B0 -> D-(Ks pi-)pi+
# B0 -> D*-(anti-D0 pi-)pi+    With anti-D0 -> k+ pi-
("14120700", "btocharm", "B0toDstarPi_D0pi_Kpi"),
# merge  B0 -> D*-(anti-D0 pi-)pi+ with anti-D0 -> k- pi+ pi+ pi-
# and anti-D0 -> K- pi+ pi0
("14120800", "btocharm", "B0toDstarPi_D0pi_Kpipipi_Kpipi0"),
("14121100", "btocharm", "B0toDrho_Kpipi"),
```

Skim package - gitlab

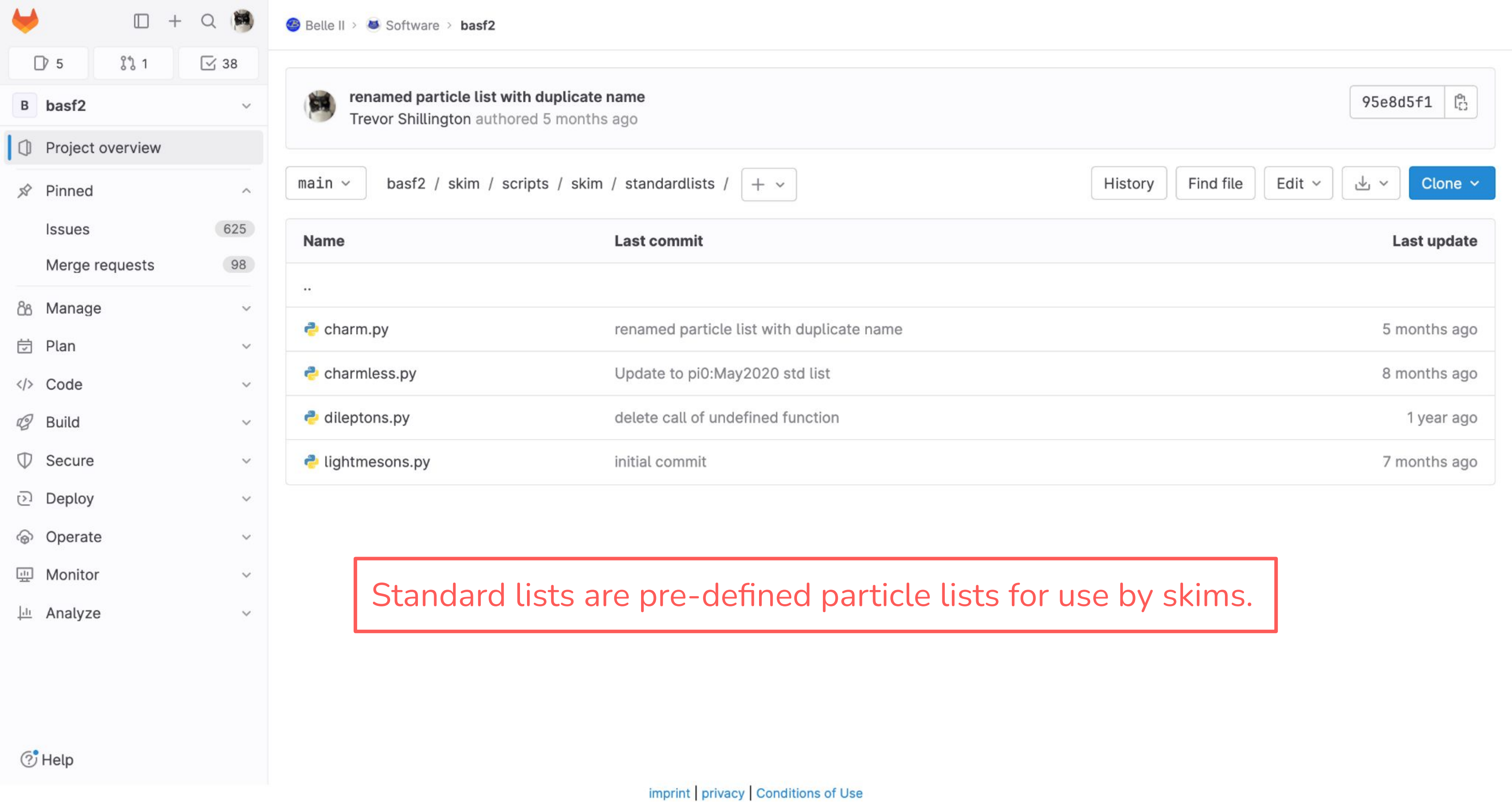


the condition for the number of tracks added.
Chanyoung Lee authored 2 months ago

ecbeda30

Name	Last commit	Last update
..		
WGs	the condition for the number of tracks added.	2 months ago
standardlists	renamed particle list with duplicate name	5 months ago
utils	Reduce memory consumption of the UpdateSkimFlag python module	1 year ago
__init__.py	Use the b2code-history command to fetch the full history.	2 years ago
core.py	Fixed incorrect handling of list parameters in CombinedSkim.load_standard_lists.	1 year ago
registry.py	forked the new skim AA2uuuu from stash.	4 months ago

Standard lists



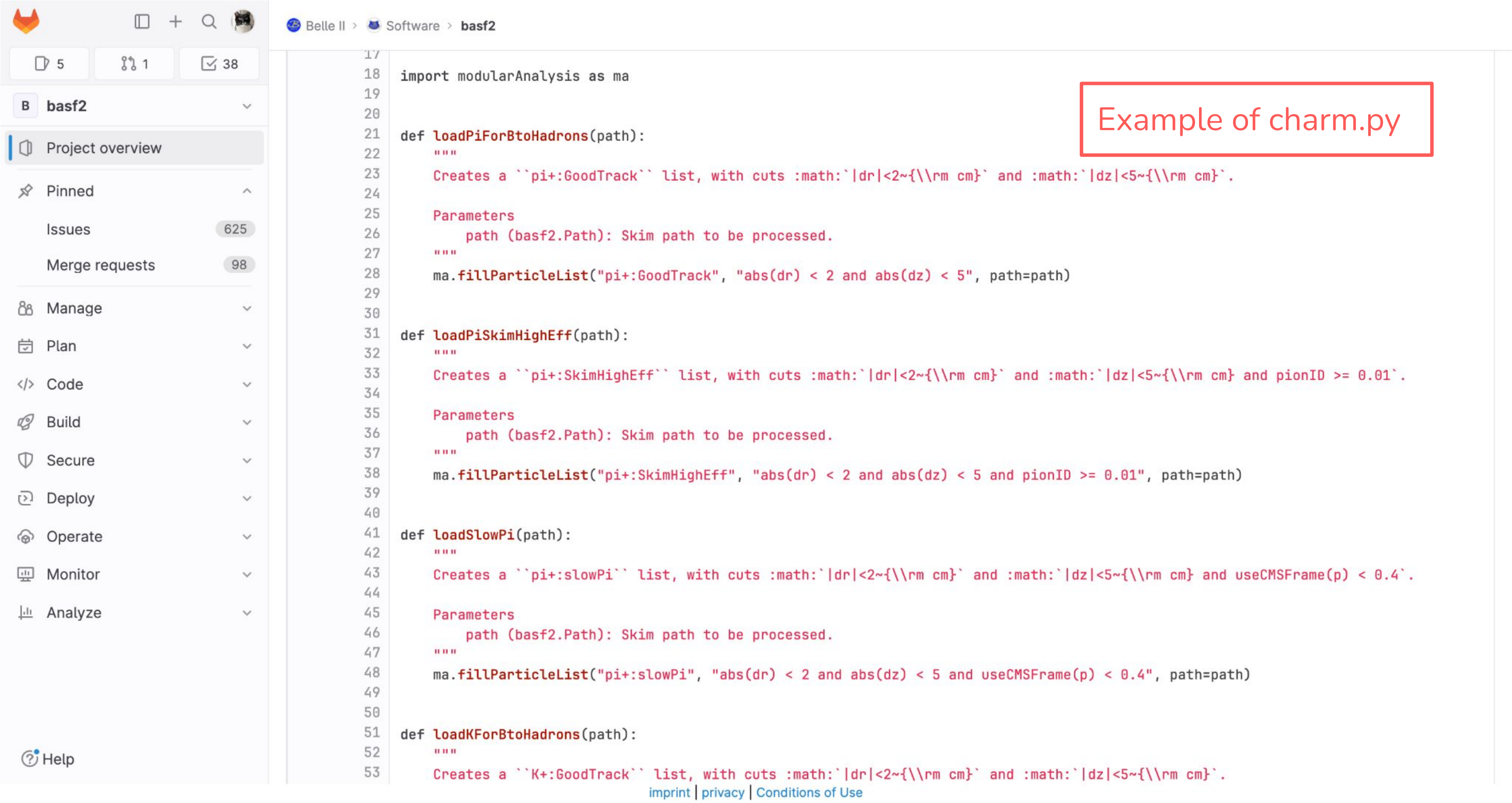
renamed particle list with duplicate name
Trevor Shillington authored 5 months ago
95e8d5f1

basf2 / skim / scripts / skim / standardlists /

Name	Last commit	Last update
..		
charm.py	renamed particle list with duplicate name	5 months ago
charmless.py	Update to pi0:May2020 std list	8 months ago
dileptons.py	delete call of undefined function	1 year ago
lightmesons.py	initial commit	7 months ago

Standard lists are pre-defined particle lists for use by skims.

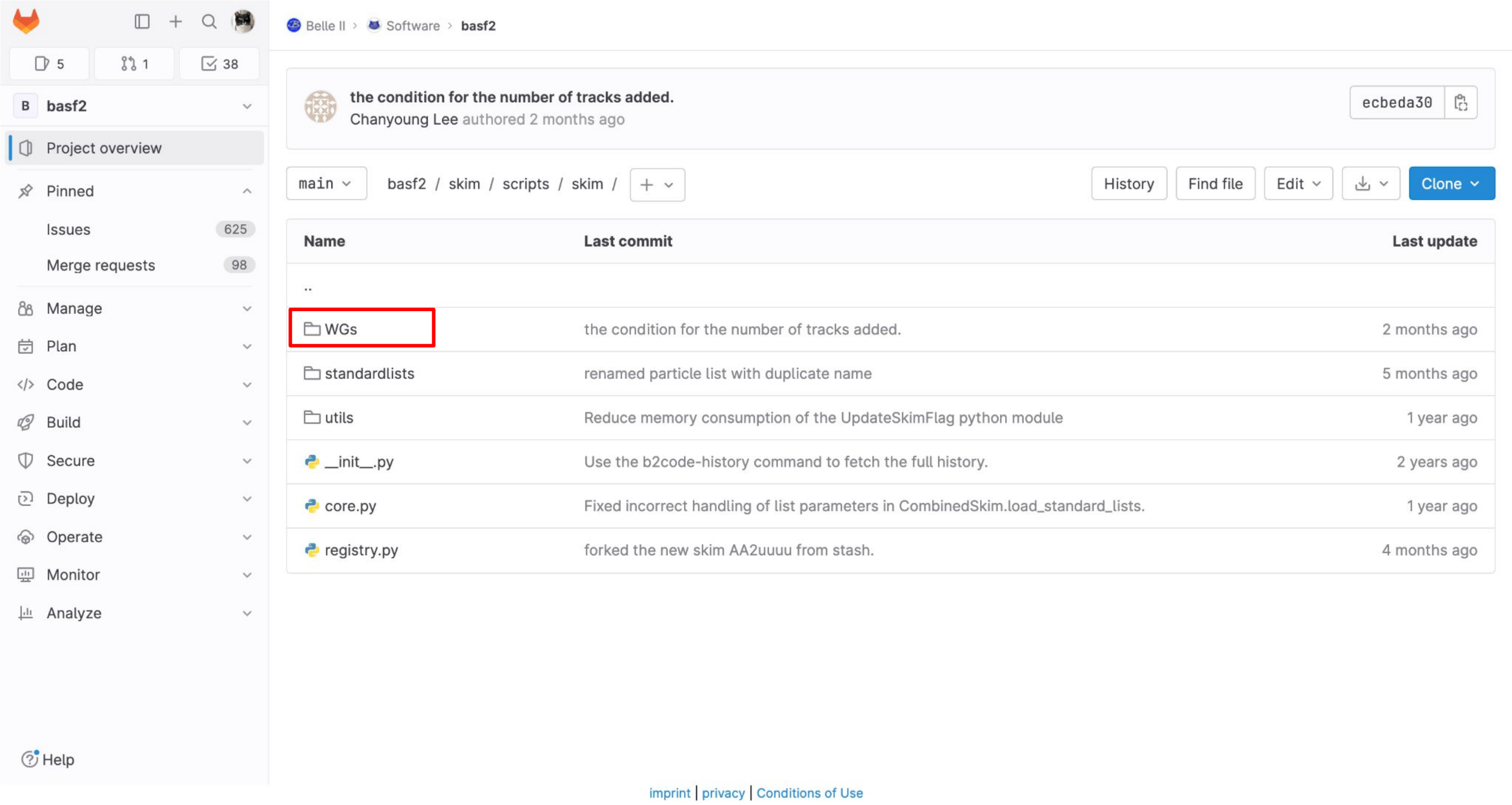
Standard lists



```python
import modularAnalysis as ma


def loadPiForBtoHadrons(path):
    """
    Creates a ``pi+:GoodTrack`` list, with cuts :math:`|dr|<2~{\\rm cm}` and :math:`|dz|<5~{\\rm cm}`.

    Parameters
        path (basf2.Path): Skim path to be processed.
    """
    ma.fillParticleList("pi+:GoodTrack", "abs(dr) < 2 and abs(dz) < 5", path=path)


def loadPiSkimHighEff(path):
    """
    Creates a ``pi+:SkimHighEff`` list, with cuts :math:`|dr|<2~{\\rm cm}` and :math:`|dz|<5~{\\rm cm} and pionID >= 0.01`.

    Parameters
        path (basf2.Path): Skim path to be processed.
    """
    ma.fillParticleList("pi+:SkimHighEff", "abs(dr) < 2 and abs(dz) < 5 and pionID >= 0.01", path=path)


def loadSlowPi(path):
    """
    Creates a ``pi+:slowPi`` list, with cuts :math:`|dr|<2~{\\rm cm}` and :math:`|dz|<5~{\\rm cm} and useCMSFrame(p) < 0.4`.

    Parameters
        path (basf2.Path): Skim path to be processed.
    """
    ma.fillParticleList("pi+:slowPi", "abs(dr) < 2 and abs(dz) < 5 and useCMSFrame(p) < 0.4", path=path)


def loadKForBtoHadrons(path):
    """
    Creates a ``K+:GoodTrack`` list, with cuts :math:`|dr|<2~{\\rm cm}` and :math:`|dz|<5~{\\rm cm}`.
```

Example of charm.py

Skim package - gitlab

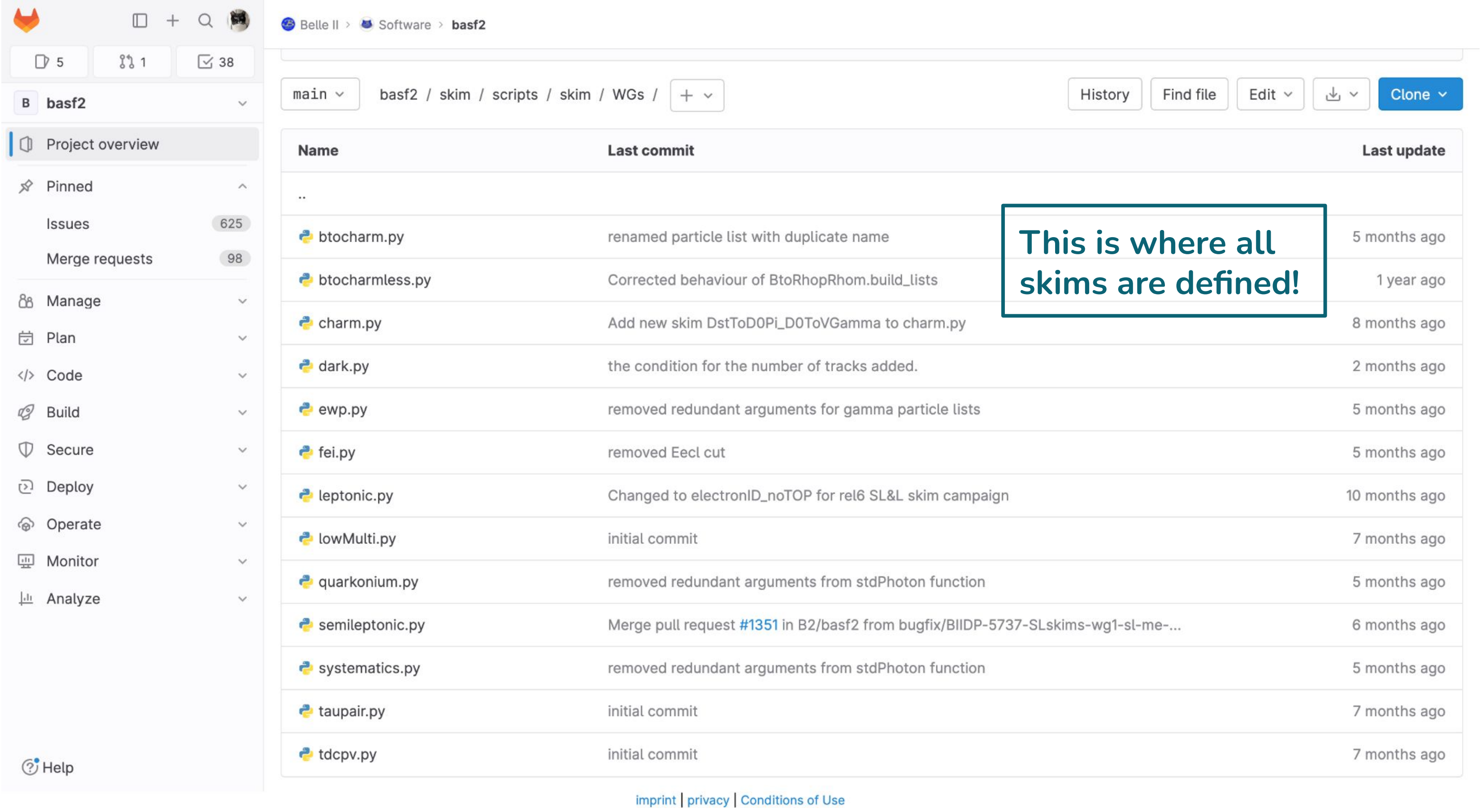


the condition for the number of tracks added.
Chanyoung Lee authored 2 months ago

basf2 / skim / scripts / skim /

Name	Last commit	Last update
..		
WGs	the condition for the number of tracks added.	2 months ago
standardlists	renamed particle list with duplicate name	5 months ago
utils	Reduce memory consumption of the UpdateSkimFlag python module	1 year ago
__init__.py	Use the b2code-history command to fetch the full history.	2 years ago
core.py	Fixed incorrect handling of list parameters in CombinedSkim.load_standard_lists.	1 year ago
registry.py	forked the new skim AA2uuuu from stash.	4 months ago

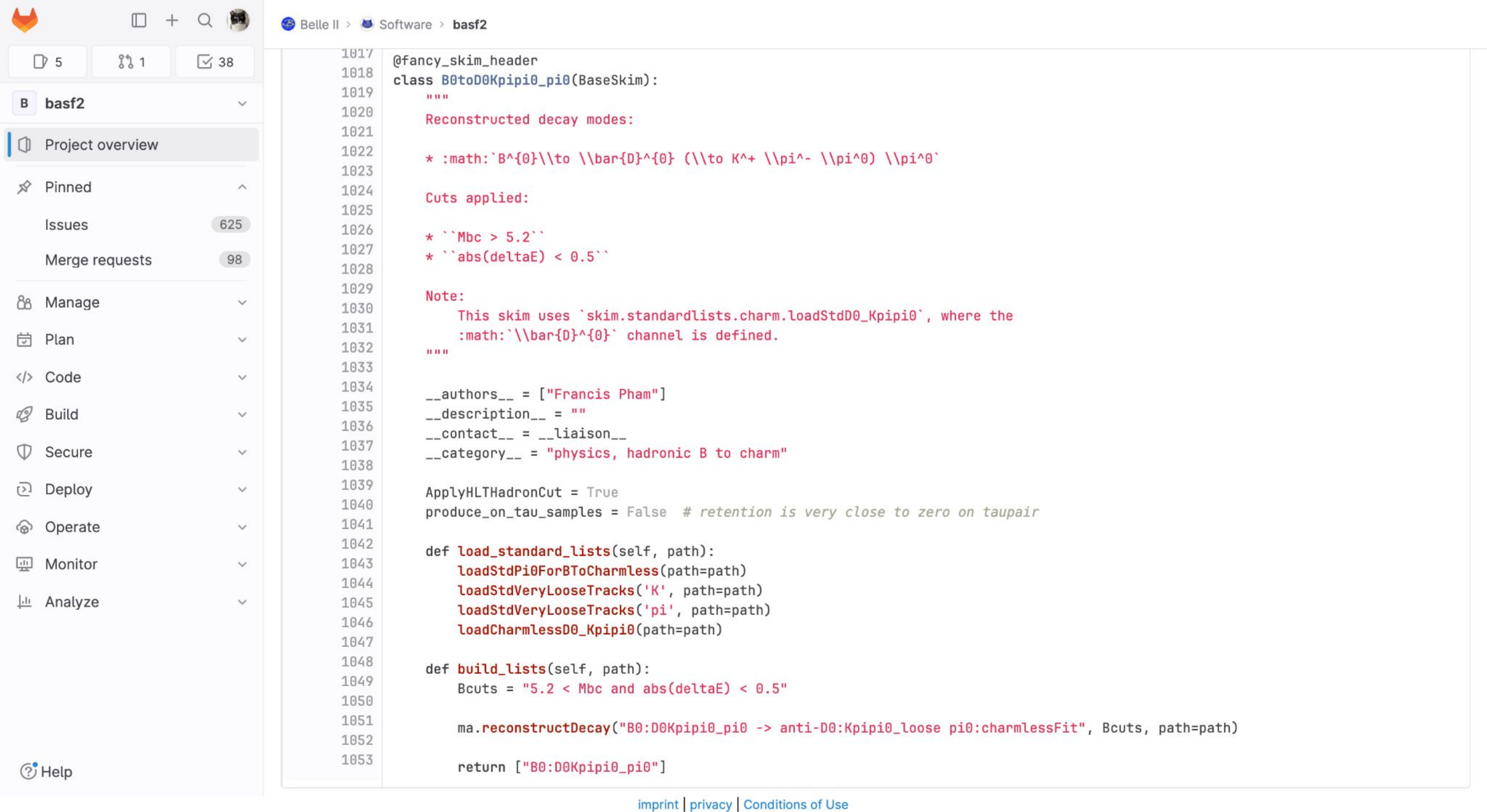
Skim package - gitlab



Name	Last commit	Last update
..		
btocharm.py	renamed particle list with duplicate name	5 months ago
btocharmless.py	Corrected behaviour of BtoRhopRhom.build_lists	1 year ago
charm.py	Add new skim DstToD0Pi_D0ToVGamma to charm.py	8 months ago
dark.py	the condition for the number of tracks added.	2 months ago
ewp.py	removed redundant arguments for gamma particle lists	5 months ago
fei.py	removed Eecl cut	5 months ago
leptonic.py	Changed to electronID_noTOP for rel6 SL&L skim campaign	10 months ago
lowMulti.py	initial commit	7 months ago
quarkonium.py	removed redundant arguments from stdPhoton function	5 months ago
semileptonic.py	Merge pull request #1351 in B2/basf2 from bugfix/BIIDP-5737-SLskims-wg1-sl-me-...	6 months ago
systematics.py	removed redundant arguments from stdPhoton function	5 months ago
taupair.py	initial commit	7 months ago
tdcpv.py	initial commit	7 months ago

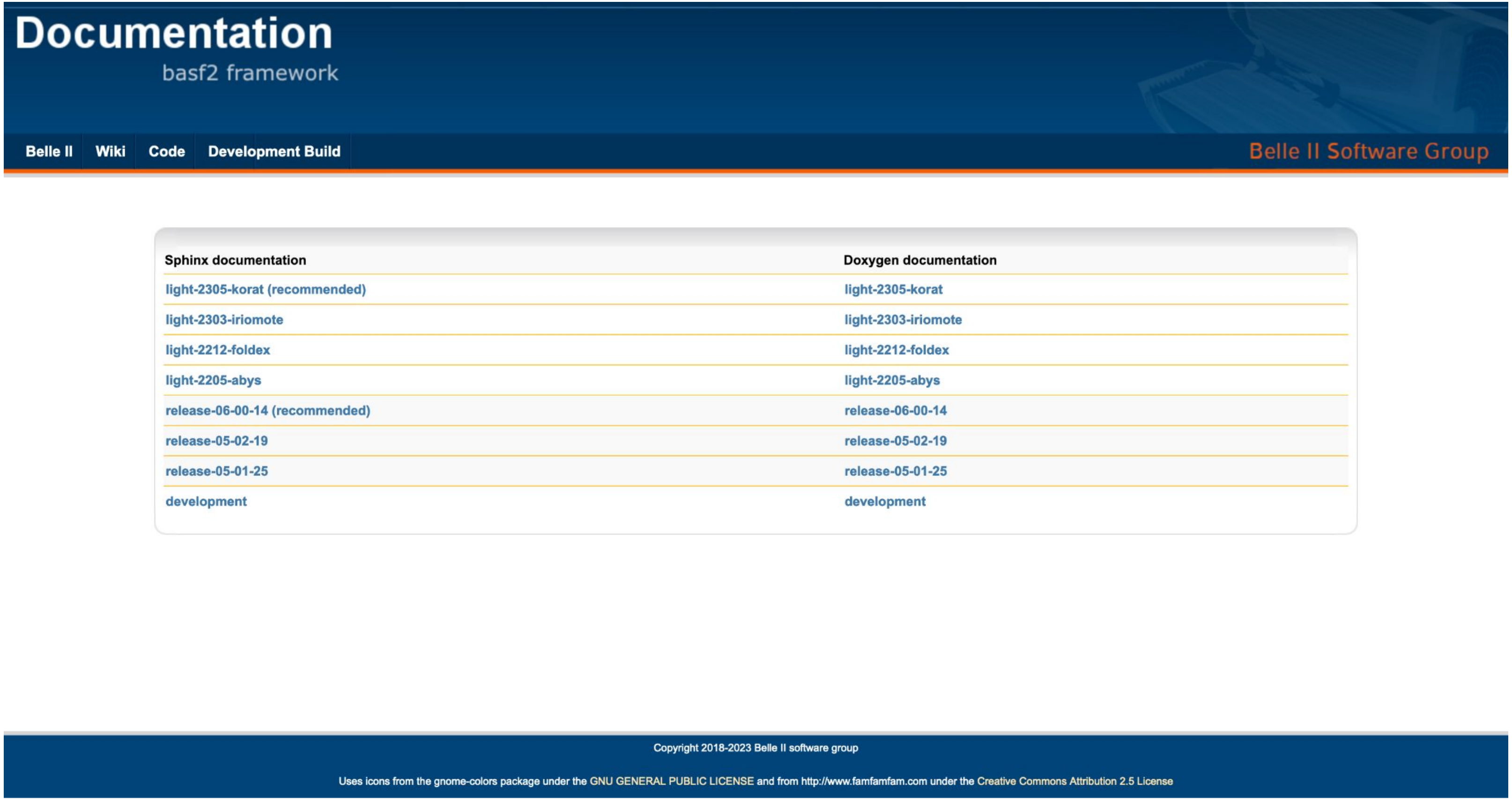
This is where all skims are defined!

Random example from btocharm.py



```python
@fancy_skim_header
class B0toD0Kpipi0_pi0(BaseSkim):
    """
    Reconstructed decay modes:

    * :math:`B^{0}\\to \\bar{D}^{0} (\\to K^+ \\pi^- \\pi^0) \\pi^0`

    Cuts applied:

    * ``Mbc > 5.2``
    * ``abs(deltaE) < 0.5``

    Note:
        This skim uses `skim.standardlists.charm.loadStdD0_Kpipi0`, where the
        :math:`\\bar{D}^{0}` channel is defined.
    """

    __authors__ = ["Francis Pham"]
    __description__ = ""
    __contact__ = __liaison__
    __category__ = "physics, hadronic B to charm"

    ApplyHLTHadronCut = True
    produce_on_tau_samples = False  # retention is very close to zero on taupair

    def load_standard_lists(self, path):
        loadStdPi0ForBToCharmless(path=path)
        loadStdVeryLooseTracks('K', path=path)
        loadStdVeryLooseTracks('pi', path=path)
        loadCharmlessD0_Kpipi0(path=path)

    def build_lists(self, path):
        Bcuts = "5.2 < Mbc and abs(deltaE) < 0.5"

        ma.reconstructDecay("B0:D0Kpipi0_pi0 -> anti-D0:Kpipi0_loose pi0:charmlessFit", Bcuts, path=path)

        return ["B0:D0Kpipi0_pi0"]
```

Skim Documentation - Sphinx



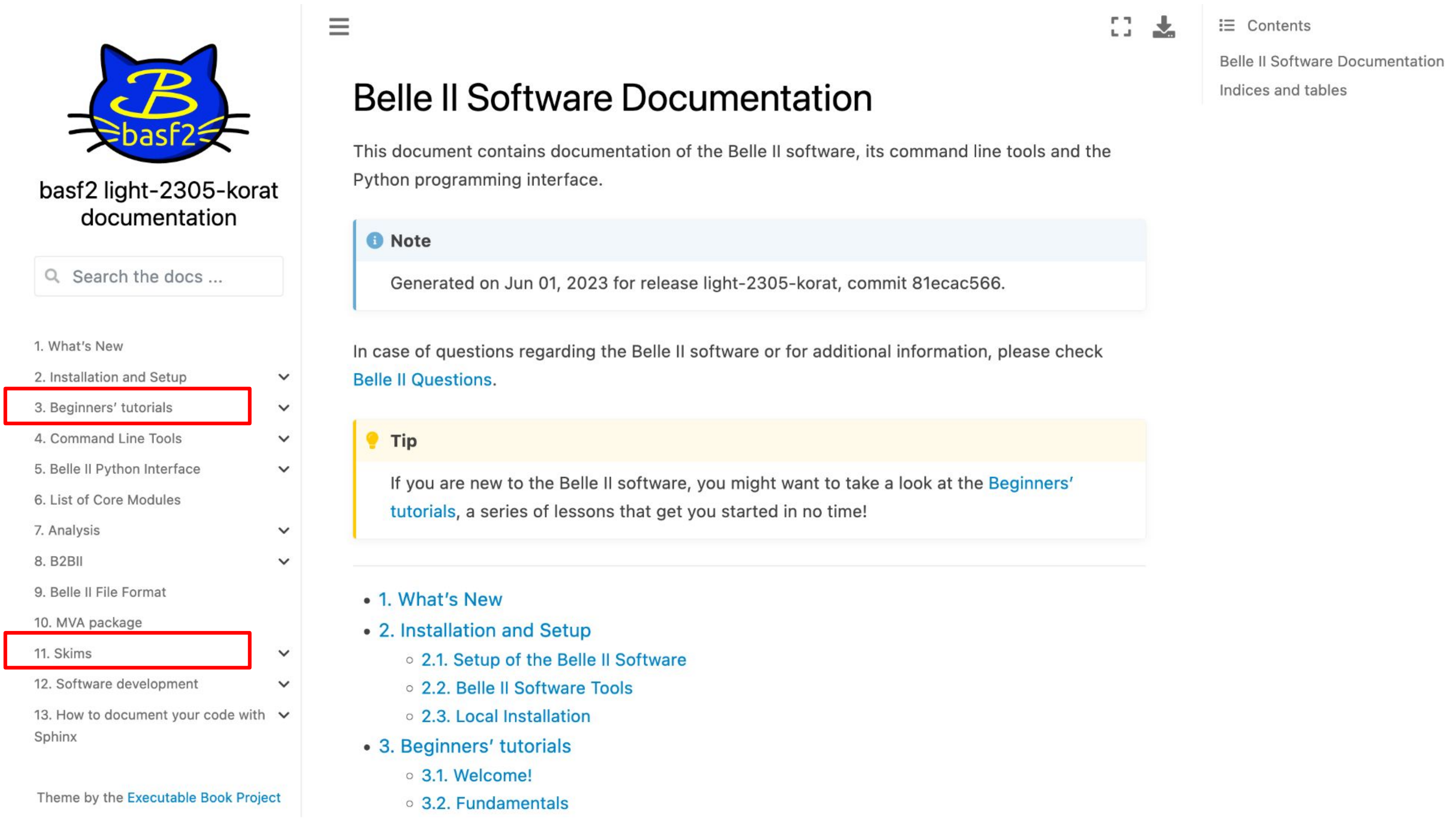
Documentation

basf2 framework

Sphinx documentation	Doxygen documentation
light-2305-korat (recommended)	light-2305-korat
light-2303-iriomote	light-2303-iriomote
light-2212-foldex	light-2212-foldex
light-2205-abys	light-2205-abys
release-06-00-14 (recommended)	release-06-00-14
release-05-02-19	release-05-02-19
release-05-01-25	release-05-01-25
development	development

https://software.belle2.org/

Skim Documentation - Sphinx



basf2 light-2305-korat documentation

Search the docs ...

Theme by the Executable Book Project

Contents

Belle II Software Documentation

Indices and tables

Belle II Software Documentation

This document contains documentation of the Belle II software, its command line tools and the Python programming interface.

Note

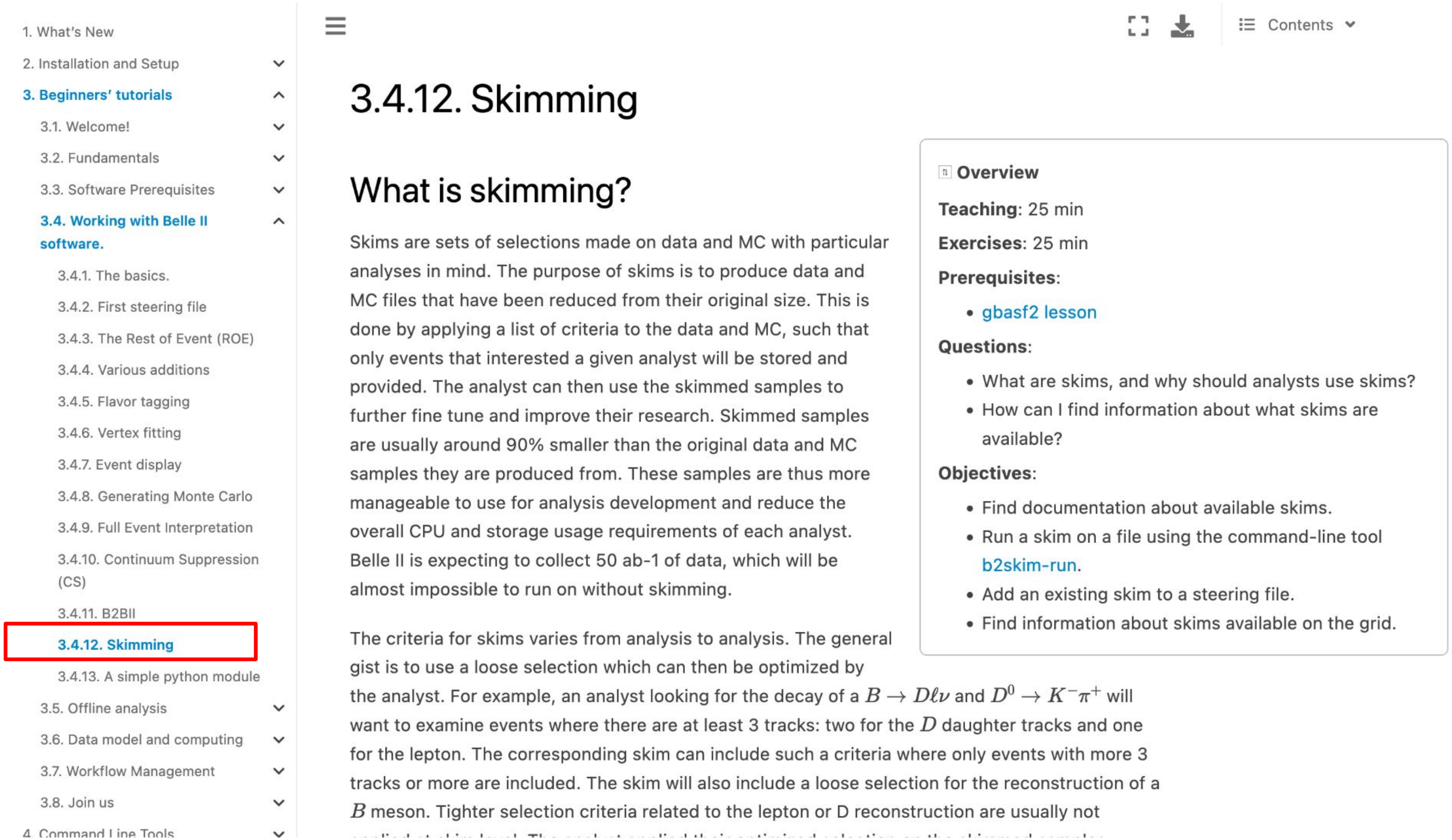
Generated on Jun 01, 2023 for release light-2305-korat, commit 81ecac566.

In case of questions regarding the Belle II software or for additional information, please check Belle II Questions.

Tip

If you are new to the Belle II software, you might want to take a look at the Beginners' tutorials, a series of lessons that get you started in no time!

Beginners' Tutorial - Sphinx



3.4.12. Skimming

What is skimming?

Skims are sets of selections made on data and MC with particular analyses in mind. The purpose of skims is to produce data and MC files that have been reduced from their original size. This is done by applying a list of criteria to the data and MC, such that only events that interested a given analyst will be stored and provided. The analyst can then use the skimmed samples to further fine tune and improve their research. Skimmed samples are usually around 90% smaller than the original data and MC samples they are produced from. These samples are thus more manageable to use for analysis development and reduce the overall CPU and storage usage requirements of each analyst. Belle II is expecting to collect 50 ab-1 of data, which will be almost impossible to run on without skimming.

Overview

Teaching: 25 min

Exercises: 25 min

Prerequisites:

- gbasf2 lesson

Questions:

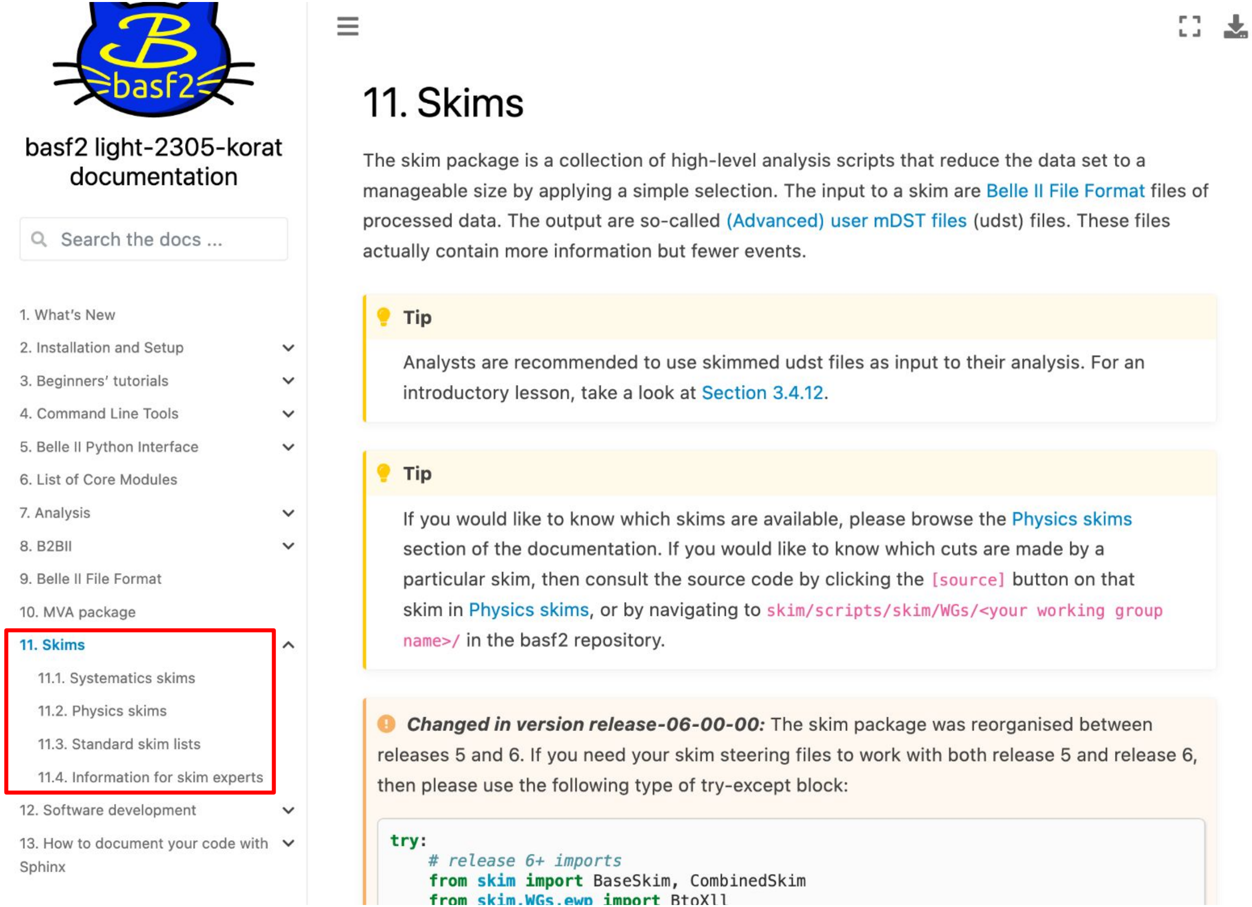
- What are skims, and why should analysts use skims?
- How can I find information about what skims are available?

Objectives:

- Find documentation about available skims.
- Run a skim on a file using the command-line tool b2skim-run.
- Add an existing skim to a steering file.
- Find information about skims available on the grid.

The criteria for skims varies from analysis to analysis. The general gist is to use a loose selection which can then be optimized by the analyst. For example, an analyst looking for the decay of a $B \to D\ell\nu$ and $D^0 \to K^-\pi^+$ will want to examine events where there are at least 3 tracks: two for the D daughter tracks and one for the lepton. The corresponding skim can include such a criteria where only events with more 3 tracks or more are included. The skim will also include a loose selection for the reconstruction of a B meson. Tighter selection criteria related to the lepton or D reconstruction are usually not

Skim Documentation - Sphinx



basf2 light-2305-korat documentation

Search the docs ...

1. What's New
2. Installation and Setup
3. Beginners' tutorials
4. Command Line Tools
5. Belle II Python Interface
6. List of Core Modules
7. Analysis
8. B2BII
9. Belle II File Format
10. MVA package
11. Skims
    - 11.1. Systematics skims
    - 11.2. Physics skims
    - 11.3. Standard skim lists
    - 11.4. Information for skim experts
12. Software development
13. How to document your code with Sphinx

11. Skims

The skim package is a collection of high-level analysis scripts that reduce the data set to a manageable size by applying a simple selection. The input to a skim are Belle II File Format files of processed data. The output are so-called (Advanced) user mDST files (udst) files. These files actually contain more information but fewer events.

Tip

Analysts are recommended to use skimmed udst files as input to their analysis. For an introductory lesson, take a look at Section 3.4.12.

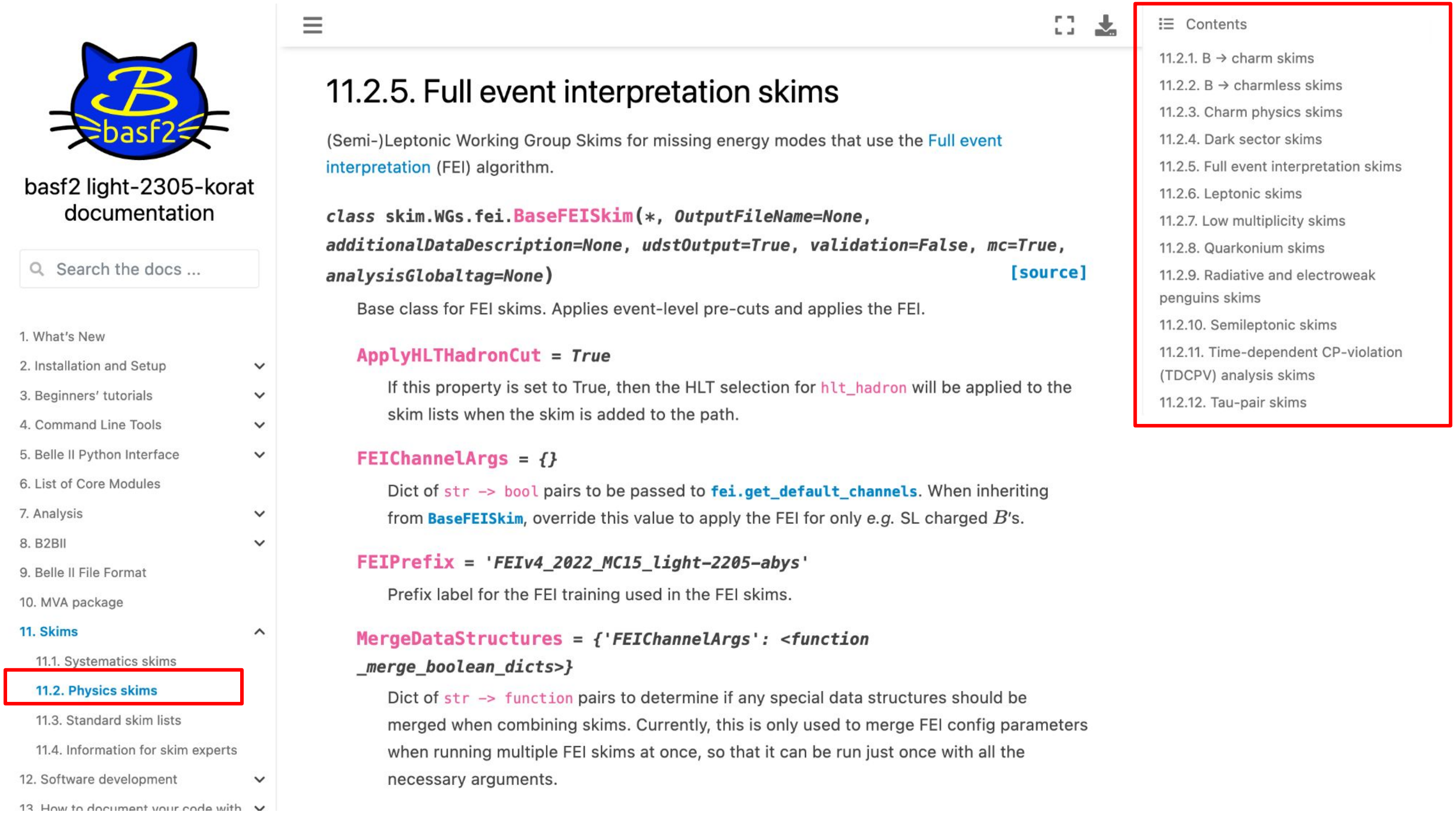
Tip

If you would like to know which skims are available, please browse the Physics skims section of the documentation. If you would like to know which cuts are made by a particular skim, then consult the source code by clicking the `[source]` button on that skim in Physics skims, or by navigating to `skim/scripts/skim/WGs/<your working group name>/` in the basf2 repository.

Changed in version release-06-00-00: The skim package was reorganised between releases 5 and 6. If you need your skim steering files to work with both release 5 and release 6, then please use the following type of try-except block:

```
try:
    # release 6+ imports
    from skim import BaseSkim, CombinedSkim
    from skim.WGs.ewp import BtoXll
```

Skim Documentation - Sphinx



basf2 light-2305-korat documentation

Search the docs ...

1. What's New
2. Installation and Setup
3. Beginners' tutorials
4. Command Line Tools
5. Belle II Python Interface
6. List of Core Modules
7. Analysis
8. B2BII
9. Belle II File Format
10. MVA package
11. Skims
    - 11.1. Systematics skims
    - 11.2. Physics skims
    - 11.3. Standard skim lists
    - 11.4. Information for skim experts
12. Software development
13. How to document your code with

11.2.5. Full event interpretation skims

(Semi-)Leptonic Working Group Skims for missing energy modes that use the Full event interpretation (FEI) algorithm.

class `skim.WGs.fei.BaseFEISkim(*, OutputFileName=None, additionalDataDescription=None, udstOutput=True, validation=False, mc=True, analysisGlobaltag=None)` [source]

Base class for FEI skims. Applies event-level pre-cuts and applies the FEI.

`ApplyHLTHadronCut = True`

If this property is set to True, then the HLT selection for `hlt_hadron` will be applied to the skim lists when the skim is added to the path.

`FEIChannelArgs = {}`

Dict of `str -> bool` pairs to be passed to `fei.get_default_channels`. When inheriting from `BaseFEISkim`, override this value to apply the FEI for only *e.g.* SL charged B's.

`FEIPrefix = 'FEIv4_2022_MC15_light-2205-abys'`

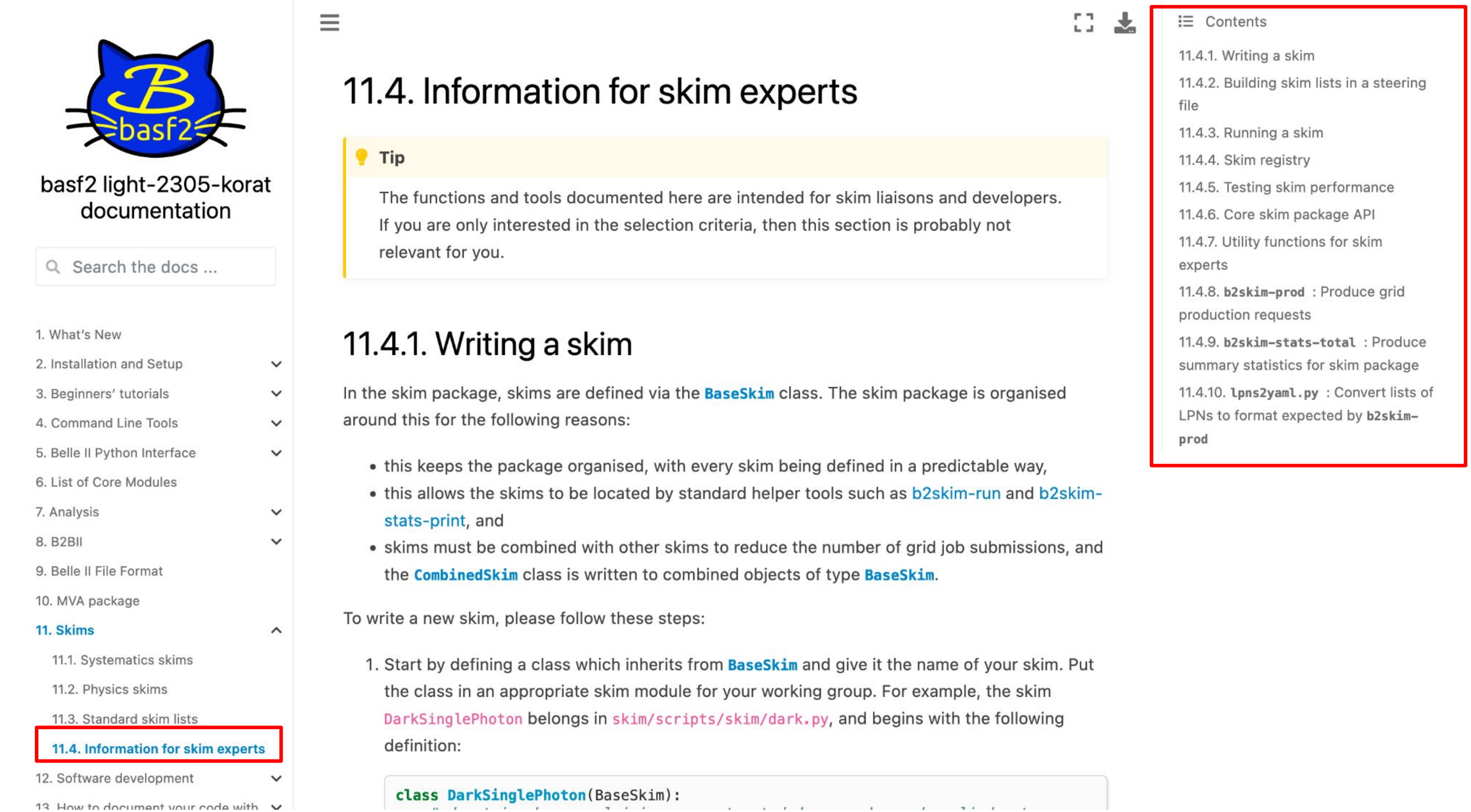
Prefix label for the FEI training used in the FEI skims.

`MergeDataStructures = {'FEIChannelArgs': <function _merge_boolean_dicts>}`

Dict of `str -> function` pairs to determine if any special data structures should be merged when combining skims. Currently, this is only used to merge FEI config parameters when running multiple FEI skims at once, so that it can be run just once with all the necessary arguments.

Contents

Skim Documentation - Sphinx



basf2 light-2305-korat documentation

Search the docs ...

1. What's New
2. Installation and Setup
3. Beginners' tutorials
4. Command Line Tools
5. Belle II Python Interface
6. List of Core Modules
7. Analysis
8. B2BII
9. Belle II File Format
10. MVA package
11. Skims
    - 11.1. Systematics skims
    - 11.2. Physics skims
    - 11.3. Standard skim lists
    - 11.4. Information for skim experts
12. Software development
13. How to document your code with

11.4. Information for skim experts

Tip

The functions and tools documented here are intended for skim liaisons and developers. If you are only interested in the selection criteria, then this section is probably not relevant for you.

11.4.1. Writing a skim

In the skim package, skims are defined via the `BaseSkim` class. The skim package is organised around this for the following reasons:

- this keeps the package organised, with every skim being defined in a predictable way,
- this allows the skims to be located by standard helper tools such as b2skim-run and b2skim-stats-print, and
- skims must be combined with other skims to reduce the number of grid job submissions, and the `CombinedSkim` class is written to combined objects of type `BaseSkim`.

To write a new skim, please follow these steps:

1. Start by defining a class which inherits from `BaseSkim` and give it the name of your skim. Put the class in an appropriate skim module for your working group. For example, the skim `DarkSinglePhoton` belongs in `skim/scripts/skim/dark.py`, and begins with the following definition:

```
class DarkSinglePhoton(BaseSkim):
```

Contents

Running a skim locally with basf2

Two ways of running a skim locally (I strongly recommend option 1)

1. Using the basf2 command **b2skim-run**
2. Incorporating some lines of code into your analysis script

Option 1:

```
b2skim-run single MySkim -i SomeFile.mdst.root
```

Where MySkim is just the name of the skim (e.g. feiSL, BtoXgamma, BtoD0h_hh, etc.)

Optional arguments:

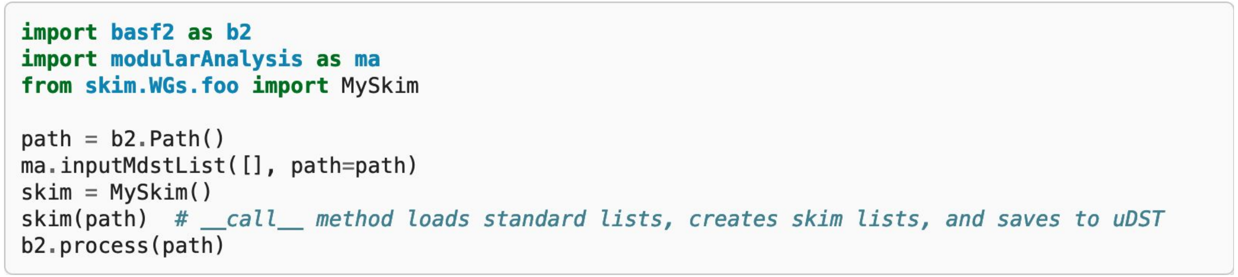
- Output filename: **-o outputFileName.udst.root**
- Number of events to run on: **-n 1000**
- Indicate if you are running on data (so MC quantities aren't saved): **--data**
- Analysis globaltag (Needed for FEI to use correct FEI training): **--analysis-globatag analysis_tools_light-2212-foldex**

Running a skim locally with basf2

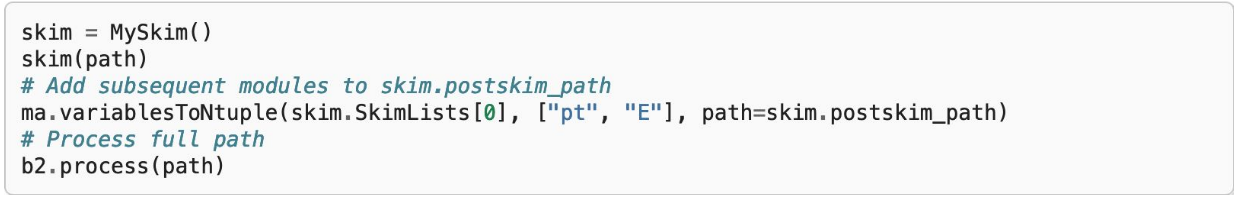
Option 2: Incorporating some lines of code into your analysis script (Only for experts!)

Warning: There are some technical subtleties here. Once skim(path) gets called, if you want to do further reconstruction/variablesToNtuple using the particleLists from the skim you have to use path=skim.postskim_path (but b2.process must call the main path at the end)

After skim(path) has been called, the skim list names are stored in the Python list skim.SkimLists



```
import basf2 as b2
import modularAnalysis as ma
from skim.WGs.foo import MySkim

path = b2.Path()
ma.inputMdstList([], path=path)
skim = MySkim()
skim(path)  # __call__ method loads standard lists, creates skim lists, and saves to uDST
b2.process(path)
```



```
skim = MySkim()
skim(path)
# Add subsequent modules to skim.postskim_path
ma.variablesToNtuple(skim.SkimLists[0], ["pt", "E"], path=skim.postskim_path)
# Process full path
b2.process(path)
```

Note: you can also use

skim = MySkim(udstOutput=False)

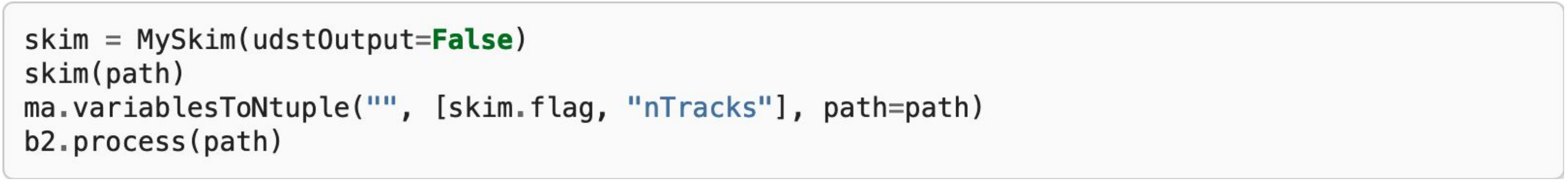
so udst files aren't produced and you just have your nTuples.

https://software.belle2.org/light-2305-korat/sphinx/skim/doc/04-experts.html#building-skim-lists-in-a-steering-file

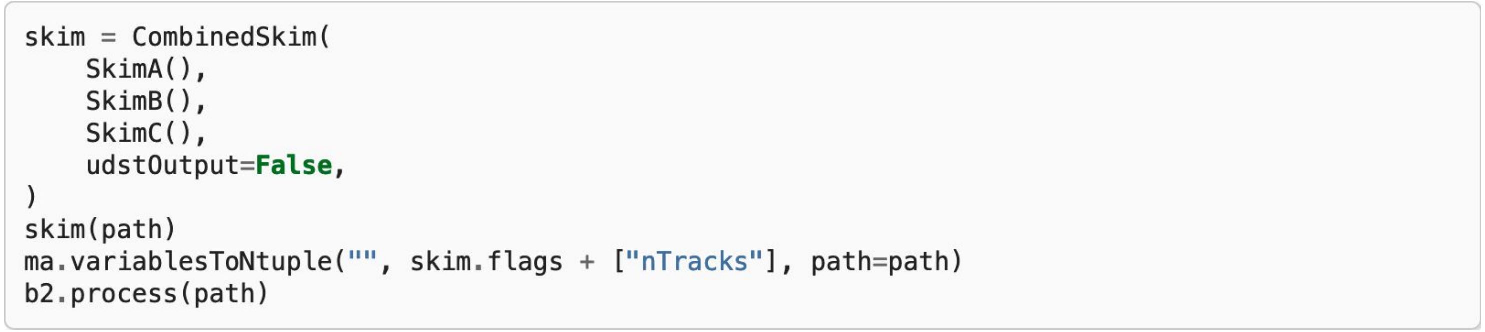
Skim Flags

When a skim is added to the path, an entry is added to the event extra info to indicate whether an event passes the skim or not (see below link)

These can be used to gather information about which events pass the skim.
Note: Skim Flags can be used on the main path.



```
skim = MySkim(udstOutput=False)
skim(path)
ma.variablesToNtuple("", [skim.flag, "nTracks"], path=path)
b2.process(path)
```



```
skim = CombinedSkim(
    SkimA(),
    SkimB(),
    SkimC(),
    udstOutput=False,
)
skim(path)
ma.variablesToNtuple("", skim.flags + ["nTracks"], path=path)
b2.process(path)
```

https://software.belle2.org/light-2305-korat/sphinx/skim/doc/04-experts.html#building-skim-lists-in-a-steering-file

Running a skim on the grid with gbasf2

Very easy, just use the basf2 command **b2skim-generate**:

```
b2skim-generate MySkim -o MySkim.py
```

Where MySkim is the name of your skim (e.g. feiSL, BtoXgamma, BtoD0h_hh, etc.)

Use -h flag to get more information about the command, or see Sphinx (link below)

Then just use gbasf2 as usual with **mySkim.py** as your steering file

https://software.belle2.org/light-2305-korat/sphinx/skim/doc/04-experts.html#b2skim-generate-generate-skim-steering-files

Finding your skim

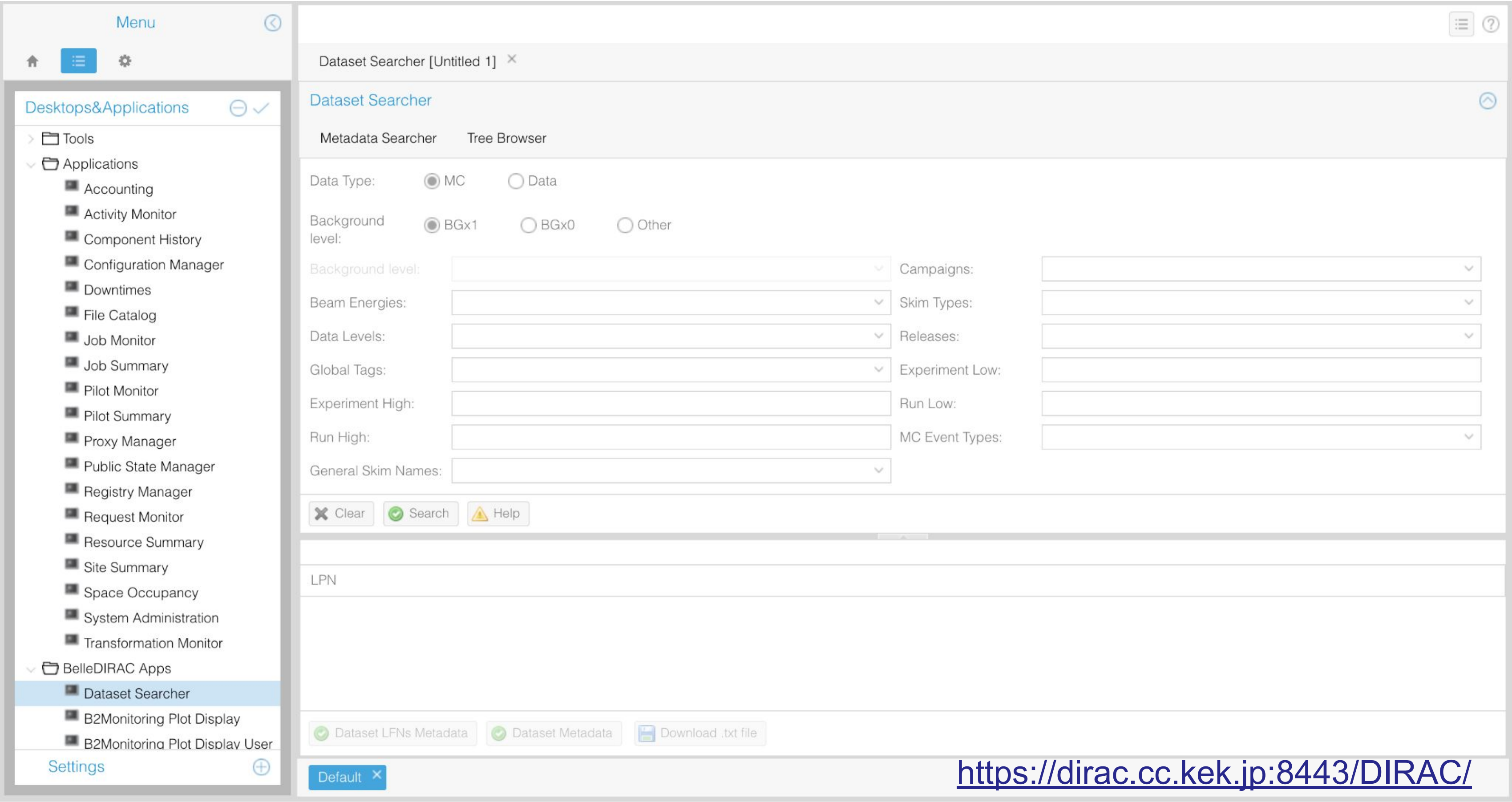
3 ways to find your skims:

1. **Dataset searcher on DIRAC (web version)**
    1. https://dirac.cc.kek.jp:8443/DIRAC/

2. **Dataset searcher via command line (gbasf2 environment)**

3. **Collections**
    1. **Recommended!**
    2. … but still early stages so not available for all skims yet
    3. Available upon request (email Trevor/Racha)
    4. https://confluence.desy.de/display/BI/Skim+Information+for+Analysts#SkimInformationforAnalysts-SkimCollections:

Finding your skim - DSS on DIRAC



Data Type: ● MC ○ Data

Background level: ● BGx1 ○ BGx0 ○ Other

Background level:

Campaigns:

Beam Energies:

Skim Types:

Data Levels:

Releases:

Global Tags:

Experiment Low:

Experiment High:

Run Low:

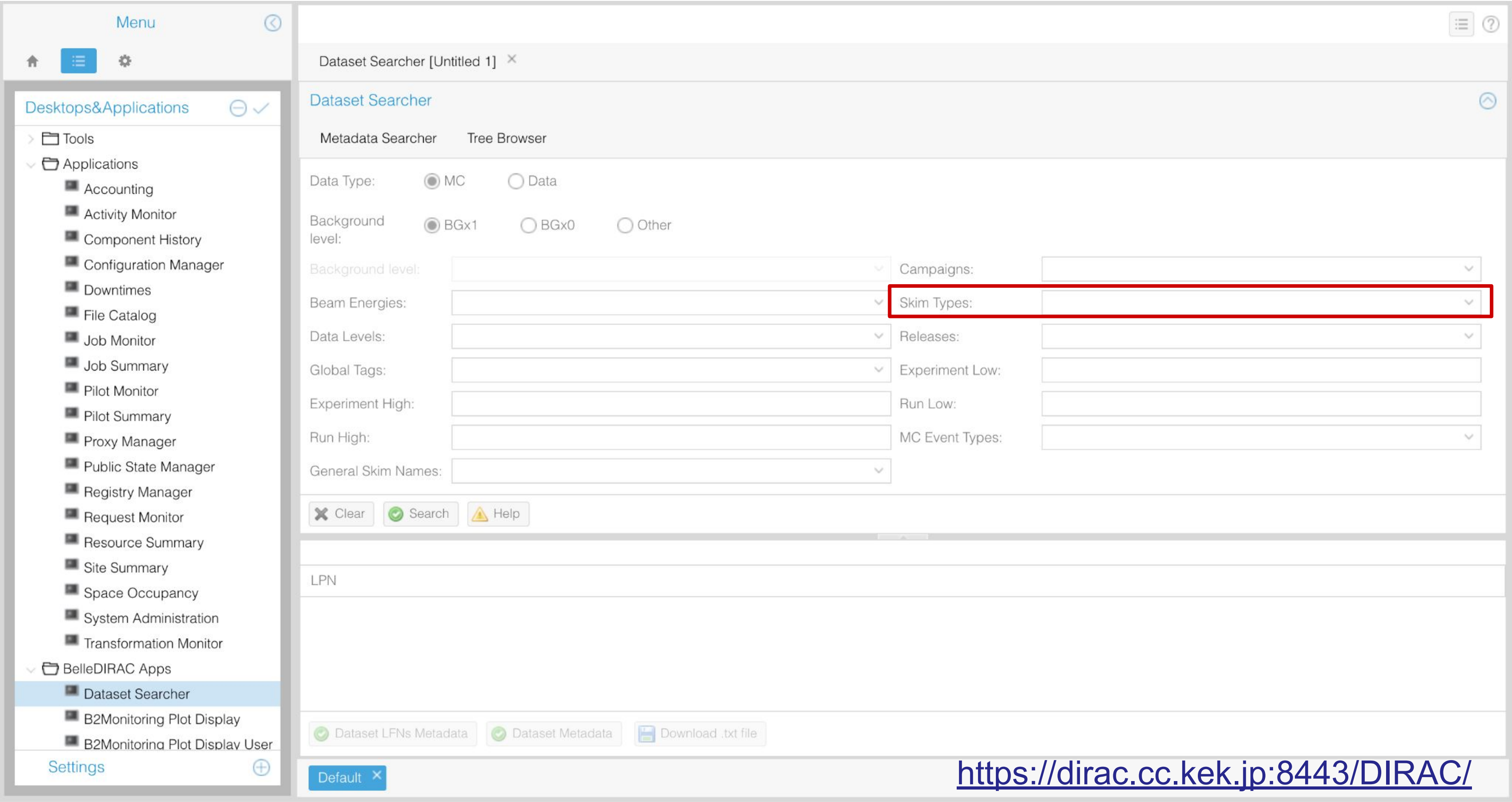
Run High:

MC Event Types:

General Skim Names:

https://dirac.cc.kek.jp:8443/DIRAC/

Finding your skim - DSS on DIRAC

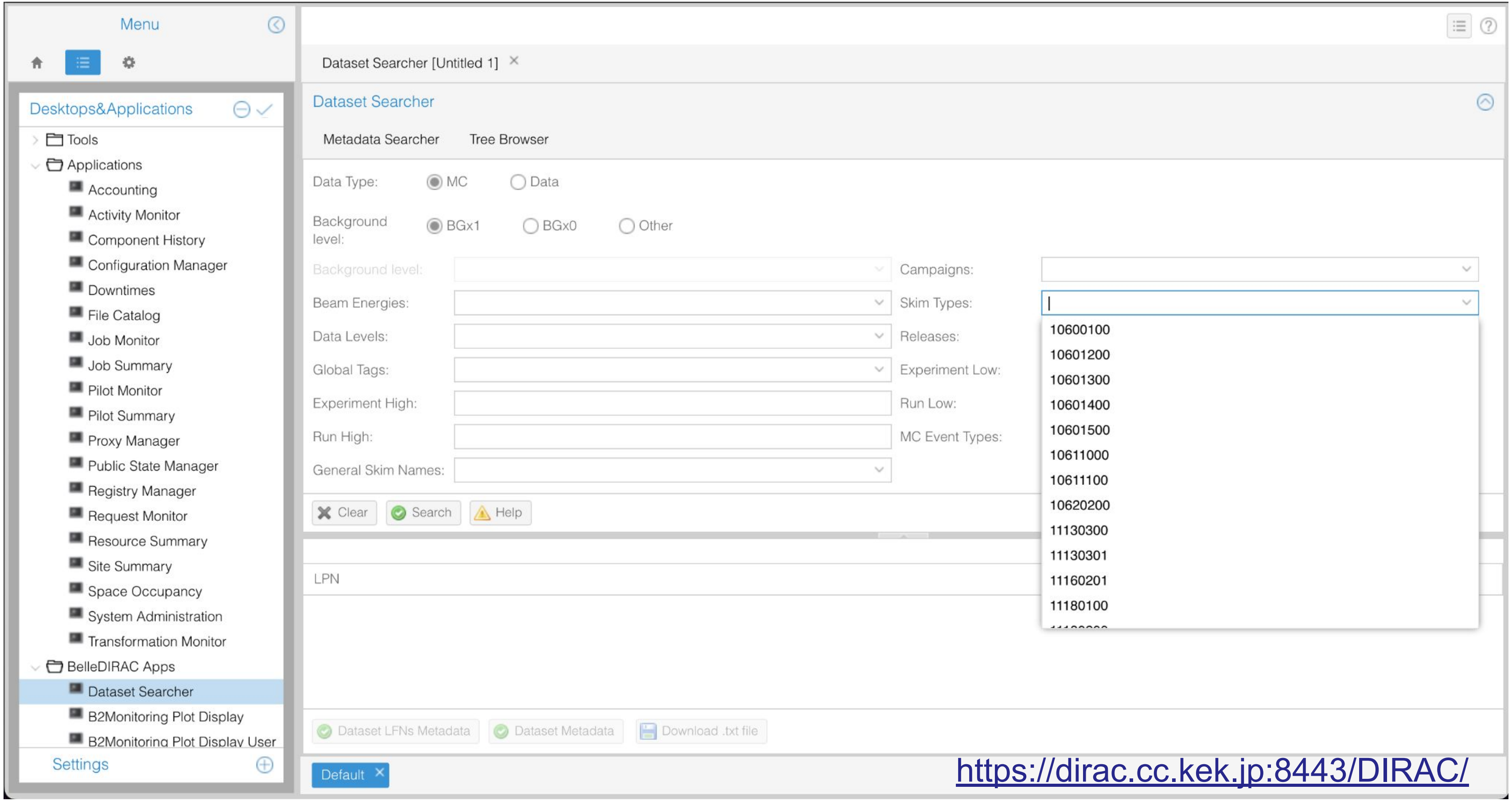


Data Type: ● MC ○ Data

Background level: ● BGx1 ○ BGx0 ○ Other

https://dirac.cc.kek.jp:8443/DIRAC/

Finding your skim - DSS on DIRAC



Data Type: ● MC ○ Data

Background level: ● BGx1 ○ BGx0 ○ Other

Background level:

Campaigns:

Beam Energies:

Skim Types: |

Data Levels:

Releases:

Global Tags:

Experiment Low:

Experiment High:

Run Low:

Run High:

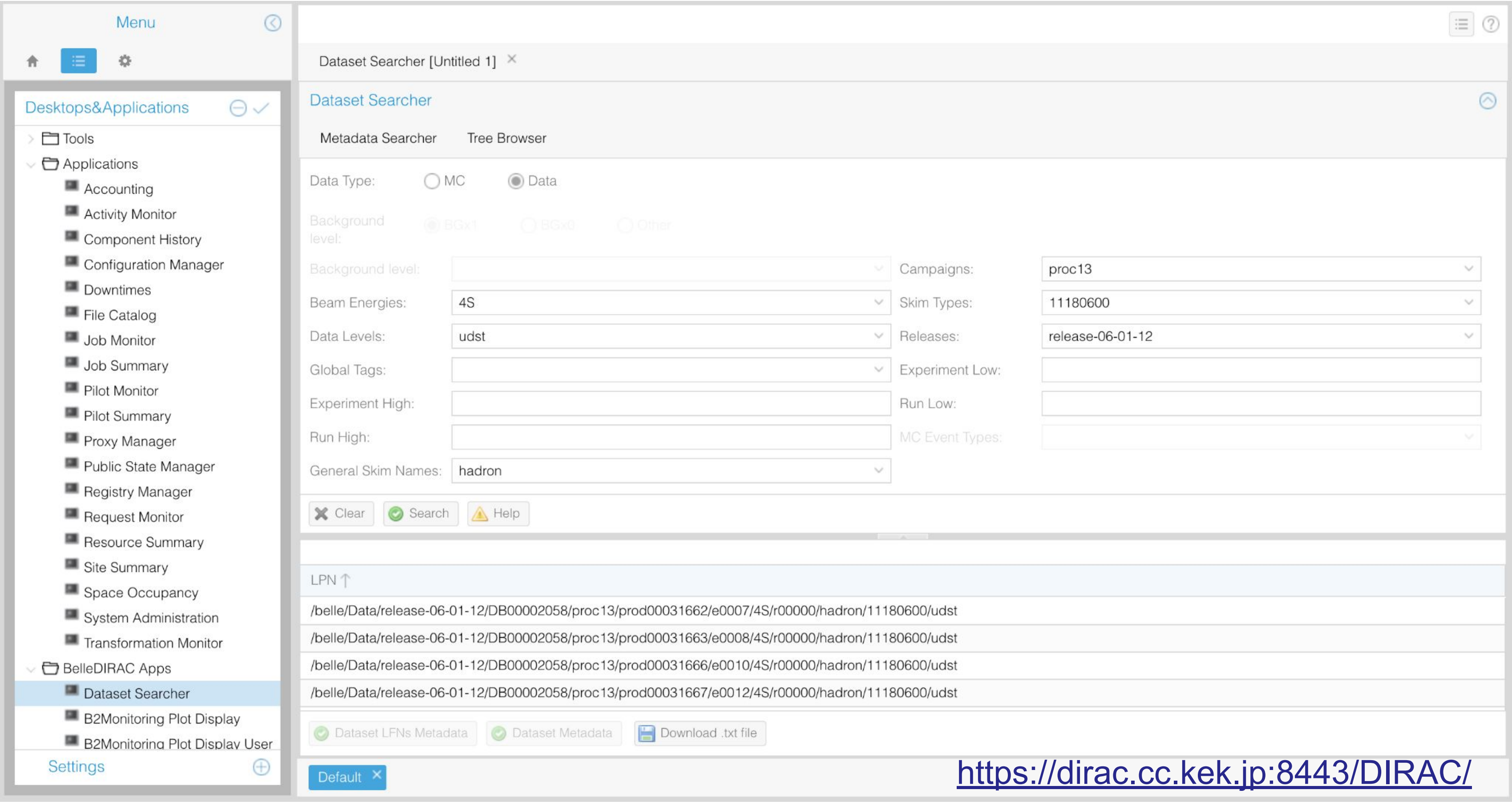
MC Event Types:

General Skim Names:

10600100
10601200
10601300
10601400
10601500
10611000
10611100
10620200
11130300
11130301
11160201
11180100

https://dirac.cc.kek.jp:8443/DIRAC/

Finding your skim - DSS on DIRAC



Data Type: ○ MC ● Data

Background level: ● BGx1 ○ BGx0 ○ Other

Background level:		Campaigns:	proc13
Beam Energies:	4S	Skim Types:	11180600
Data Levels:	udst	Releases:	release-06-01-12
Global Tags:		Experiment Low:	
Experiment High:		Run Low:	
Run High:		MC Event Types:	
General Skim Names:	hadron		

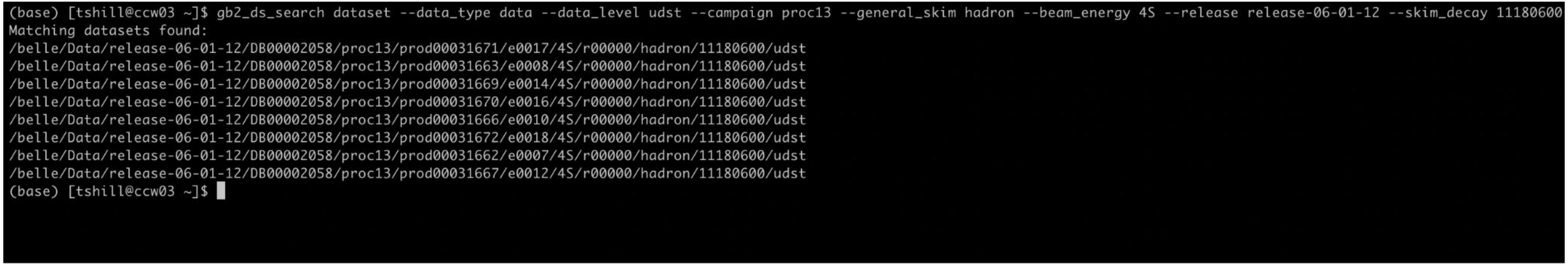
LPN ↑
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031662/e0007/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031663/e0008/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031666/e0010/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031667/e0012/4S/r00000/hadron/11180600/udst

https://dirac.cc.kek.jp:8443/DIRAC/

Finding your skim - DSS via command line

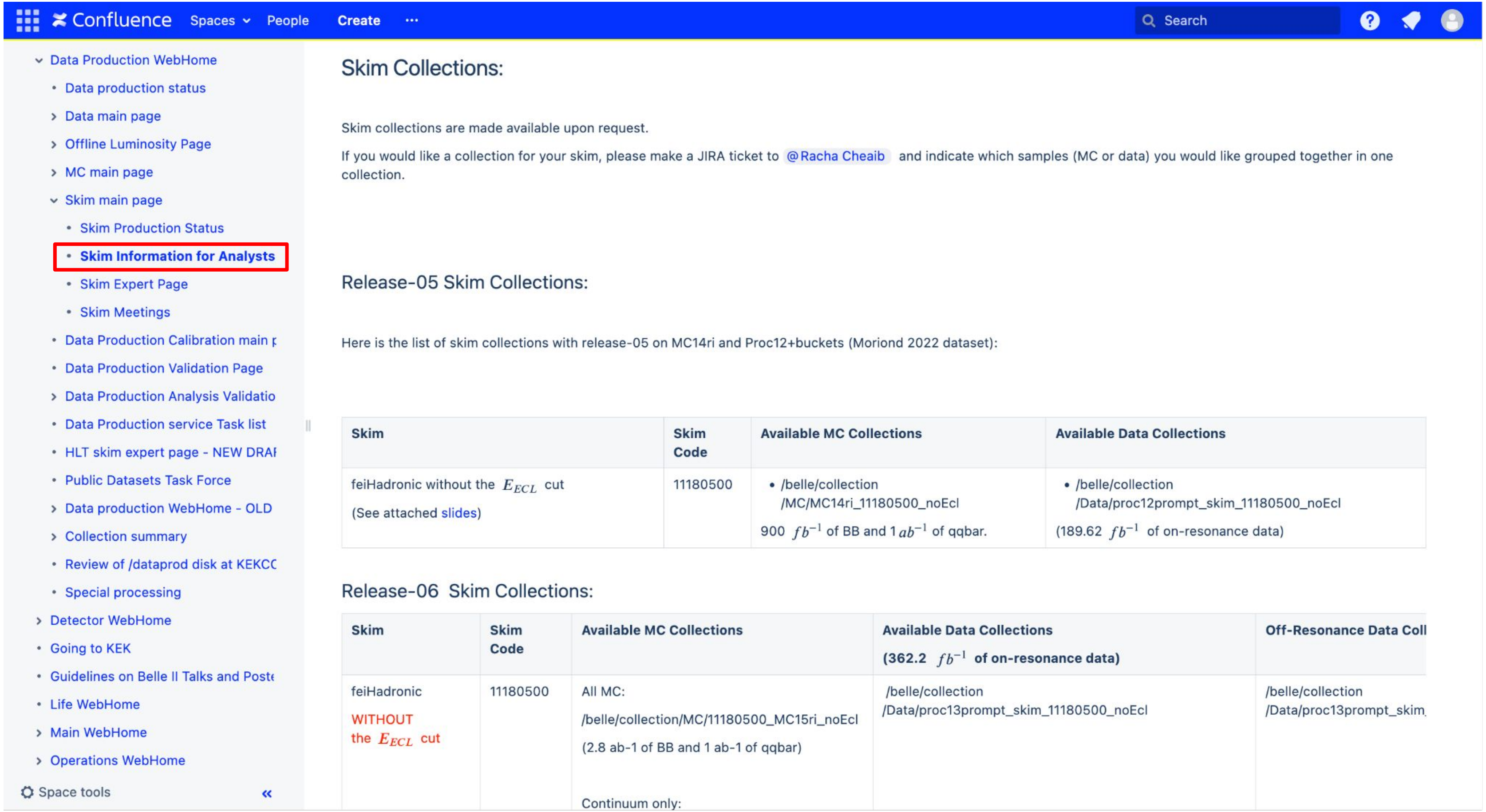
- Alternatively, you can use the command line version when you are set up in gbasf2 environment:
  - On kekcc: **source /cvmfs/belle.kek.jp/grid/gbasf2/pro/setup.sh**

gb2_ds_search dataset --data_type data --data_level udst --campaign proc13 --general_skim hadron --beam_energy 4S --release release-06-01-12 --skim_decay 11180600



```
(base) [tshill@ccw03 ~]$ gb2_ds_search dataset --data_type data --data_level udst --campaign proc13 --general_skim hadron --beam_energy 4S --release release-06-01-12 --skim_decay 11180600
Matching datasets found:
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031671/e0017/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031663/e0008/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031669/e0014/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031670/e0016/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031666/e0010/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031672/e0018/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031662/e0007/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031667/e0012/4S/r00000/hadron/11180600/udst
(base) [tshill@ccw03 ~]$ 
```

Finding your skim - Collections


Confluence
Spaces
People
Create
···
Search
Data Production WebHome
Data production status
Data main page
Offline Luminosity Page
MC main page
Skim main page
Skim Production Status
Skim Information for Analysts
Skim Expert Page
Skim Meetings
Data Production Calibration main p
Data Production Validation Page
Data Production Analysis Validatio
Data Production service Task list
HLT skim expert page - NEW DRAF
Public Datasets Task Force
Data production WebHome - OLD
Collection summary
Review of /dataprod disk at KEKCC
Special processing
Detector WebHome
Going to KEK
Guidelines on Belle II Talks and Poste
Life WebHome
Main WebHome
Operations WebHome
Space tools

Skim Collections:

Skim collections are made available upon request.

If you would like a collection for your skim, please make a JIRA ticket to @Racha Cheaib and indicate which samples (MC or data) you would like grouped together in one collection.

Release-05 Skim Collections:

Here is the list of skim collections with release-05 on MC14ri and Proc12+buckets (Moriond 2022 dataset):

Skim	Skim Code	Available MC Collections	Available Data Collections
feiHadronic without the E_{ECL} cut (See attached slides)	11180500	• /belle/collection /MC/MC14ri_11180500_noEcl 900 fb^{-1} of BB and 1 ab^{-1} of qqbar.	• /belle/collection /Data/proc12prompt_skim_11180500_noEcl (189.62 fb^{-1} of on-resonance data)

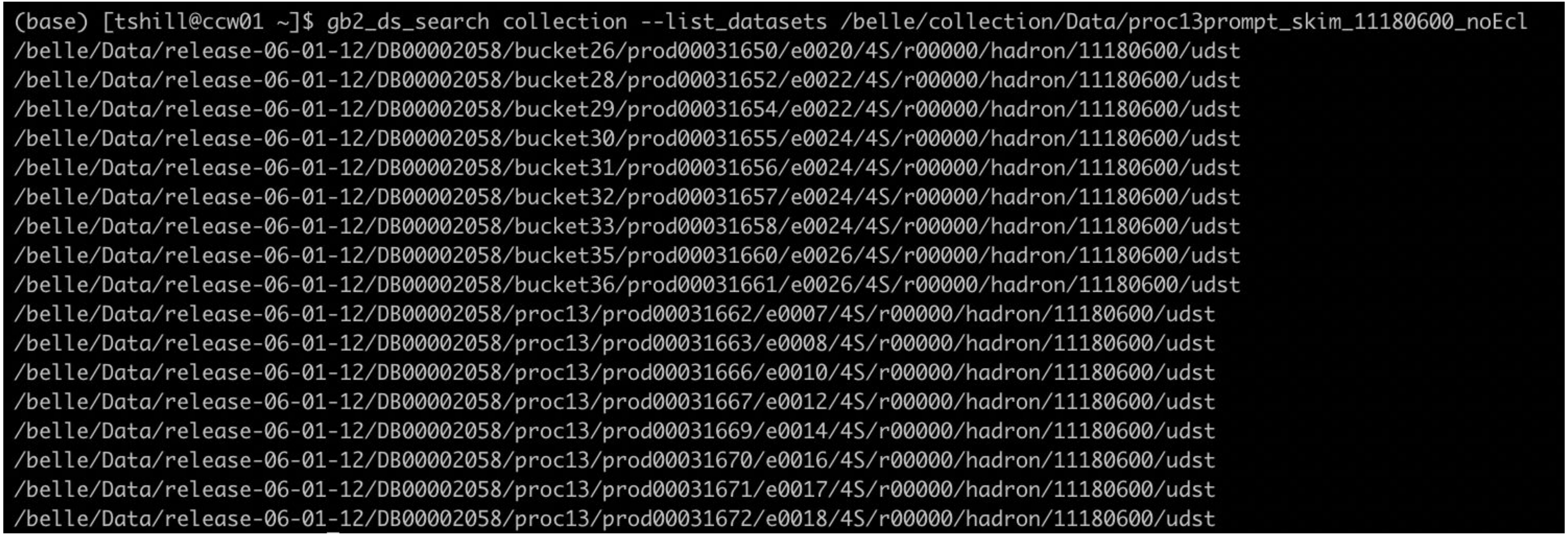
Release-06 Skim Collections:

Skim	Skim Code	Available MC Collections	Available Data Collections (362.2 fb^{-1} of on-resonance data)	Off-Resonance Data Coll
feiHadronic WITHOUT the E_{ECL} cut	11180500	All MC: /belle/collection/MC/11180500_MC15ri_noEcl (2.8 ab-1 of BB and 1 ab-1 of qqbar) Continuum only:	/belle/collection /Data/proc13prompt_skim_11180500_noEcl	/belle/collection /Data/proc13prompt_skim

Finding your skim - Collections

In gbasf2 environment:

gb2_ds_search collection --list_datasets /belle/collection/Data/proc13prompt_skim_11180600_noEcl



```
(base) [tshill@ccw01 ~]$ gb2_ds_search collection --list_datasets /belle/collection/Data/proc13prompt_skim_11180600_noEcl
/belle/Data/release-06-01-12/DB00002058/bucket26/prod00031650/e0020/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket28/prod00031652/e0022/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket29/prod00031654/e0022/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket30/prod00031655/e0024/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket31/prod00031656/e0024/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket32/prod00031657/e0024/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket33/prod00031658/e0024/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket35/prod00031660/e0026/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/bucket36/prod00031661/e0026/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031662/e0007/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031663/e0008/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031666/e0010/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031667/e0012/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031669/e0014/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031670/e0016/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031671/e0017/4S/r00000/hadron/11180600/udst
/belle/Data/release-06-01-12/DB00002058/proc13/prod00031672/e0018/4S/r00000/hadron/11180600/udst
```

Summary

- Skims are analysis-oriented data and MC
- They contain more information than mdst, but have less events
- **Meaning your analysis will run much quicker if using skims!**
- There is a lot of work to be done on skims & we are always looking for more help!
    - Contact Trevor, Racha, Stefano, or Umberto if interested

More questions? Great resources:

Confluence pages: https://confluence.desy.de/display/BI/Skim+main+page

B2questions: https://questions.belle2.org/questions/

Mailing list: dataprod-skim@belle2.org, dataprod@belle2.org

Basf2 documentation (Sphinx): https://software.belle2.org/ (checkout the beginners' tutorial 3.4.12 and Skims)

Collection Summary: https://confluence.desy.de/display/BI/Collection+summary (skims at bottom)

Gitlab (source code): https://gitlab.desy.de/belle2/software/basf2/-/tree/main/skim

FEI: https://software.belle2.org/light-2305-korat/sphinx/online_book/basf2/fei.html

DIRAC (for dataset searcher) : https://dirac.cc.kek.jp:8443/DIRAC/