Hadron PID tutorial
D*→D⁰(→Kπ)π

Sam Cunliffe[1], Jake Bennett[2] ← this guy wrote the original scripts

[1]Pacific Northwest National Laboratory, [2]Carnegie Mellon University

Pre-B2GM physics performance tutorials.   17 June 2017

U.S. DEPARTMENT OF
ENERGY

- ▶ Everything you need is in a **git repo** and also a **tarball on kekcc**.
  https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

- ▶ Whilst I'm talking please open up a terminal either KEK (preferred) or somewhere with `/cvmfs` mounted and do the following:

- ▶ If you like git:

```
$ git clone \
ssh://git@stash.desy.de:7999/~scunliff/b2gm-hadronpid-tutorial-jun2017.git
$ cd b2gm-hadronpid-tutorial-jun2017/
```

- ▶ If you dislike git / are already working at kekcc / like tab completion:

```
$ cp -r ~scunliffe/public/b2gm-hardronpid-tutorial-jun2017.tar.gz
$ tar -zxvf b2gm-hadronpid-tutorial-jun2017.tar.gz
$ cd b2gm-hadronpid-tutorial-jun2017/
```

- ▶ Then normal basf2 setup steps (we're working in `release-00-08-00`)

```
$ . /cvmfs/belle.cern.ch/sl6/tools/setup_belle2
$ setuprel release-00-08-00
```

Preamble

▶ Once you've sourced the Belle II setup script, and setup a release you should have the environment variable `$BELLE2_RELEASE_DIR`

```
$ echo $BELLE2_RELEASE_DIR
$ ls $BELLE2_RELEASE_DIR
```

https://stash.desy.de/projects/B2/repos/software/browse

▶ All of these scripts for this tutorial were copied and adapted from `$BELLE2_RELEASE_DIR/analysis/validation`

- ■ This place is a good source of example scripts in general.

Particle identification performance

► For physics performance, we care about maintaining a large **efficiency** for a low **fake rate**. Here are some very verbose definitions:

  ■ **Efficiency** = number of correctly identified particles under their own hypothesis for some cut on a figure of merit divided by the total true number of particles that really were whatever hypothesis.

  ■ **Fake rate** = number of incorrectly identified particles under the alternate hypothesis for the same cut divided by the total number of particles that really were the alternate hypothesis.

  ■ **'Separation'** an imprecise jargony term (in this context) means how far apart the efficiency and fake rate are for two hypotheses. i.e. **'K/π separation'**

► The physics performance group also provides you, the user, with ability to ask the reverse e.g.:
"*I'm prepared to accept 95% efficiency what is the corresponding fake rate and where are the figure of merit cuts?*"

Particle identification performance

▶ For physics performance, we care about maintaining a large **efficiency** for a low **fake rate**. Here are some maths definitions:

  ■ **Efficiency**

$$\varepsilon_{\pi id} = n_{\text{true } \pi, \text{ selected}} \, / \, N_{\text{true } \pi \text{ total}}$$
$$\varepsilon_{Kid} = n_{\text{true } K, \text{ selected}} \, / \, N_{\text{true } K \text{ total}}$$

  ■ **Fake rate**

$$R_{K \to \pi} = n_{\text{true } K \text{ selected as } \pi} \, / \, N_{\text{true } K \text{ total}}$$
$$R_{\pi \to K} = n_{\text{true } \pi \text{ selected as } K} \, / \, N_{\text{true } \pi \text{ total}}$$

  ■ **'Separation'**

  loosely: $\varepsilon_{\pi id} - R_{K \to \pi}$ and $\varepsilon_{Kid} - R_{\pi \to K}$

▶ The physics performance group also provides you, the user, with ability to ask the reverse e.g.:
"*I'm prepared to accept 95% efficiency what is the corresponding fake rate and where are the figure of merit cuts?*"

Particle identification performance

► For physics performance, we care about maintaining a large **efficiency** for a low **fake rate**. Here are some maths definitions:

■ **Efficiency**

$$\varepsilon_{\pi id} = n_{true\ \pi,\ selected} / N_{true\ \pi\ total}$$
$$\varepsilon_{Kid} = n_{true\ K,\ selected} / N_{true\ K\ total}$$

■ **Fake rate**

$$R_{K\rightarrow\pi} = n_{true\ K\ selected\ as\ \pi} / N_{true\ K\ total}$$
$$R_{\pi\rightarrow K} = n_{true\ \pi\ selected\ as\ K} / N_{true\ \pi\ total}$$

■ **'Separation'**

loosely: $\varepsilon_{\pi id} - R_{K\rightarrow\pi}$ and $\varepsilon_{Kid} - R_{\pi\rightarrow K}$

► The physics performance group also provides you, the user, with ability to ask the reverse e.g.:
"*I'm prepared to accept 95% efficiency what is the corresponding fake rate and where are the figure of merit cuts?*"

Particle identification performance

▶ For physics performance, we care about maintaining a large **efficiency** for a low **fake rate**. Here are some maths definitions:

■ **Efficiency**

$$\varepsilon_{\pi\text{id}} = n_{\text{true }\pi,\text{ selected}} / N_{\text{true }\pi\text{ total}}$$
$$\varepsilon_{K\text{id}} = n_{\text{true }K,\text{ selected}} / N_{\text{true }K\text{ total}}$$

π/K separation

■ **Fake rate**

$$R_{K\to\pi} = n_{\text{true }K\text{ selected as }\pi} / N_{\text{true }K\text{ total}}$$
$$R_{\pi\to K} = n_{\text{true }\pi\text{ selected as }K} / N_{\text{true }\pi\text{ total}}$$

K/π separation

■ **'Separation'**

loosely: $\varepsilon_{\pi\text{id}} - R_{K\to\pi}$ and $\varepsilon_{K\text{id}} - R_{\pi\to K}$

▶ The physics performance group also provides you, the user, with ability to ask the reverse e.g.:
"I'm prepared to accept 95% efficiency what is the corresponding fake rate and where are the figure of merit cuts?"

Physics description of the method

$$D^{*+} \rightarrow D^0(\rightarrow K^-\pi^+)\pi^+$$

$$D^{*-} \rightarrow \overline{D}^0(\rightarrow K^+\pi^-)\pi^-$$

Physics description of the method

$$D^{*+} \rightarrow D^0 (\rightarrow K^- \pi^+) \pi^+{}_{slow}$$

$$D^{*-} \rightarrow \overline{D}^0 (\rightarrow K^+ \pi^-) \pi^-{}_{slow}$$

▶ Reconstruct the 'slow **pion**' and determine its **charge**...

▶ …therefore you determine the **charge** of the D*

▶ …and therefore the **flavour** of the D^0 ($\overline{D}^0$)

▶ …and therefore **which is which** in the $K^\mp \pi^\pm$ pair, solely from charge information (because $D^0 \rightarrow K^- \pi^+$ and $\overline{D}^0 \rightarrow K^+ \pi^-$)

Physics description of the method

$$D^{*+} \rightarrow D^0(\rightarrow K^- \pi^+) \pi^+_{slow}$$

$$D^{*-} \rightarrow \overline{D}^0(\rightarrow K^+ \pi^-) \pi^-_{slow}$$

▶ Reconstruct the 'slow **pion**' and determine its **charge**...

▶ …therefore you determine the **charge** of the D*

▶ …and therefore the **flavour** of the D^0 ($\overline{D}^0$)

▶ …and therefore **which is which** in the $K^{\mp}\pi^{\pm}$ pair, solely from charge information (because $D^0 \rightarrow K^-\pi^+$ and $\overline{D}^0 \rightarrow K^+\pi^-$)

Physics description of the method

$$D^{*+} \rightarrow D^0 (\rightarrow K^- \pi^+) \pi^+_{slow}$$

$$D^{*-} \rightarrow \overline{D}^0 (\rightarrow K^+ \pi^-) \pi^-_{slow}$$

▶ Reconstruct the 'slow **pion**' and determine its **charge**...

▶ …therefore you determine the **charge** of the D*

▶ …and therefore the **flavour** of the D^0 ($\overline{D}^0$)

▶ …and therefore **which is which** in the $K^\mp \pi^\pm$ pair, solely from charge information (because $D^0 \rightarrow K^- \pi^+$ and $\overline{D}^0 \rightarrow K^+ \pi^-$)

Physics description of the method

$$D^{*+} \rightarrow D^0 (\rightarrow K^- \pi^+) \pi^+_{slow}$$

$$D^{*-} \rightarrow \overline{D}^0 (\rightarrow K^+ \pi^-) \pi^-_{slow}$$

▶ Reconstruct the 'slow **pion**' and determine its **charge**...

▶ …therefore you determine the **charge** of the D*

▶ …and therefore the **flavour** of the D^0 ($\overline{D}^0$)

▶ …and therefore **which is which** in the $K^\mp \pi^\pm$ pair, solely from charge information (because $D^0 \rightarrow K^- \pi^+$ and $\overline{D}^0 \rightarrow K^+ \pi^-$)

Physics description of the method

$$D^{*+} \rightarrow D^0 (\rightarrow K^- \pi^+) \pi^+_{slow}$$

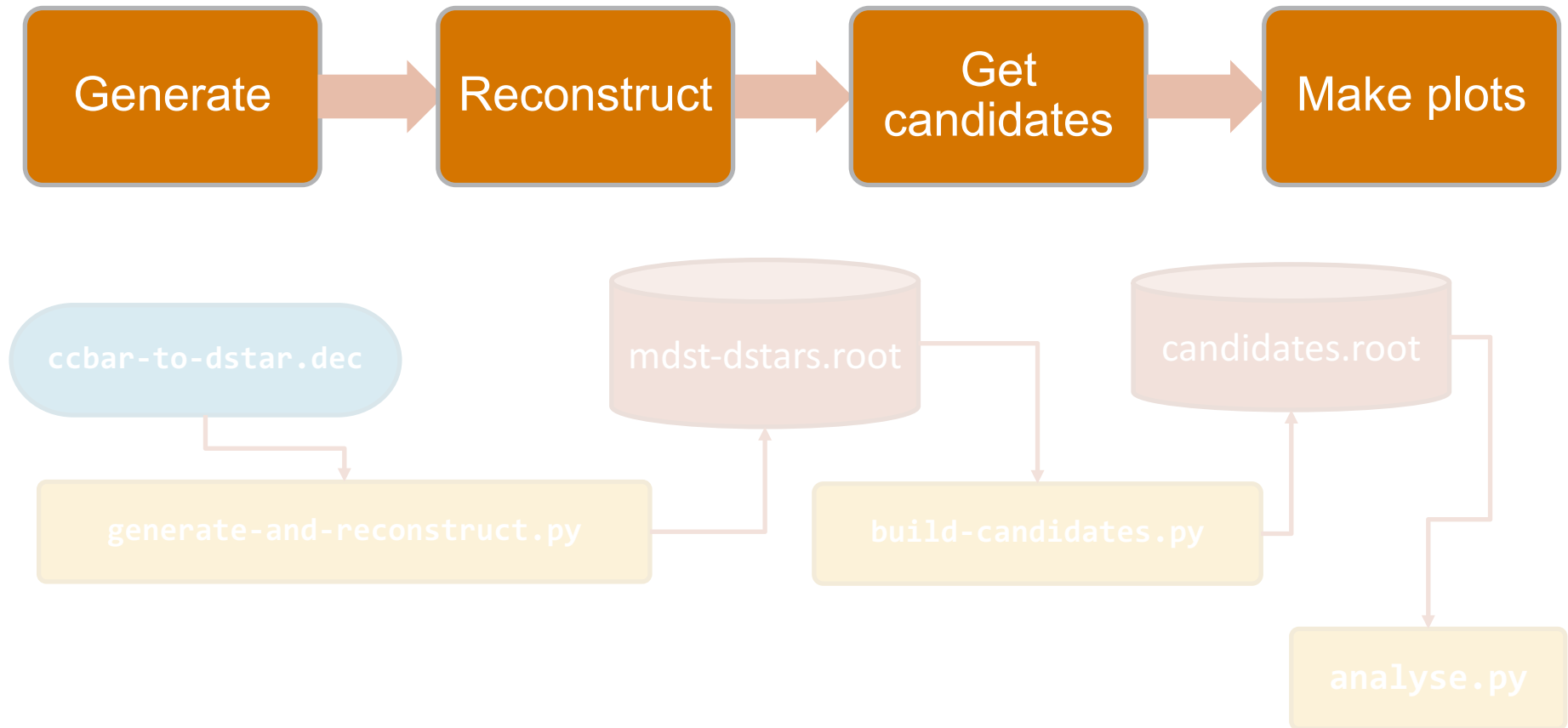$$D^{*-} \rightarrow \overline{D}^0 (\rightarrow K^+ \pi^-) \pi^-_{slow}$$

▶ Reconstruct the 'slow **pion**' and determine its **charge**...

▶ …therefore you determine the **charge** of the D*

▶ …and therefore the **flavour** of the D^0 ($\overline{D}^0$)

▶ …and therefore **which is which** in the $K^{\mp}\pi^{\pm}$ pair, solely from charge information (because $D^0 \rightarrow K^-\pi^+$ and $\overline{D}^0 \rightarrow K^+\pi^-$)

"Mother Nature's MC truth"

The basf2 steps
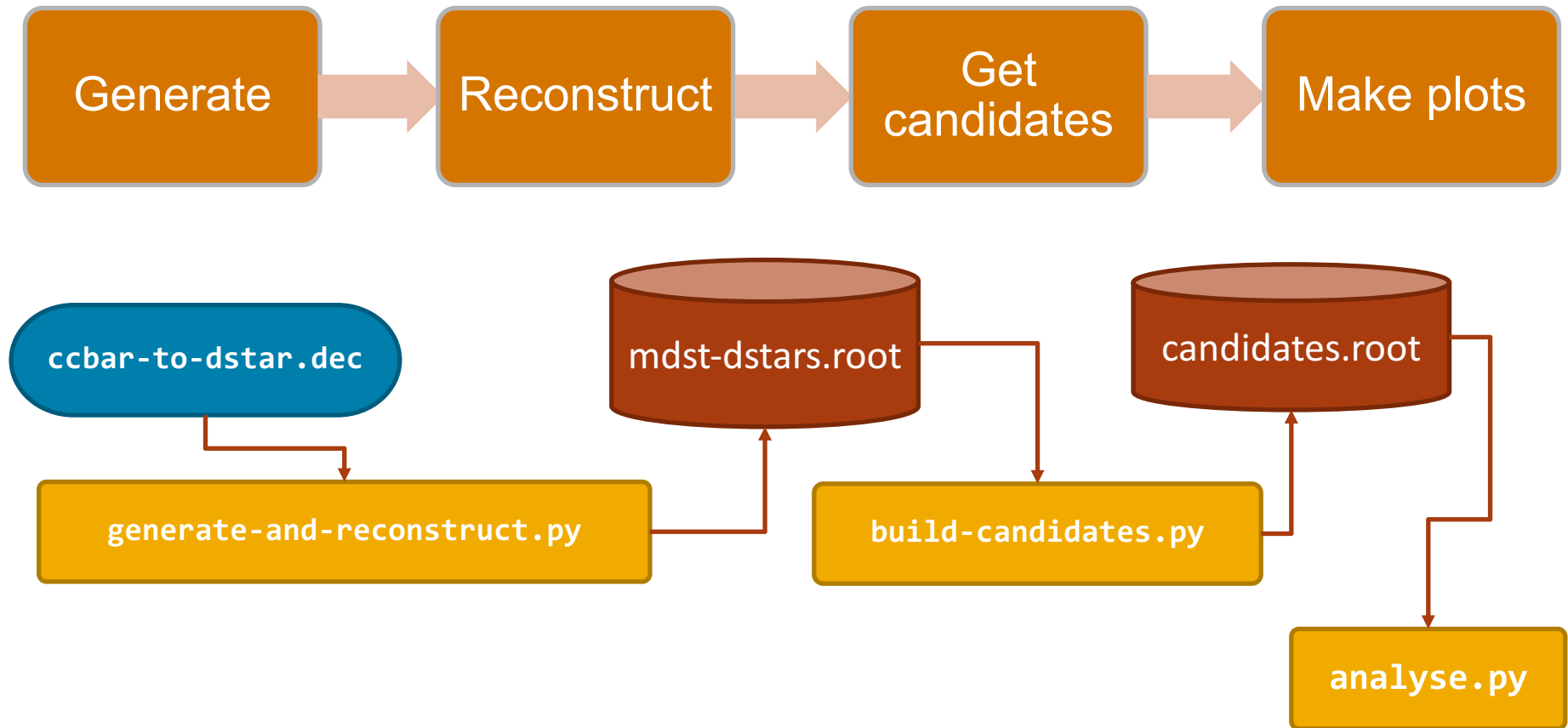**Scripts in the directory**

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse
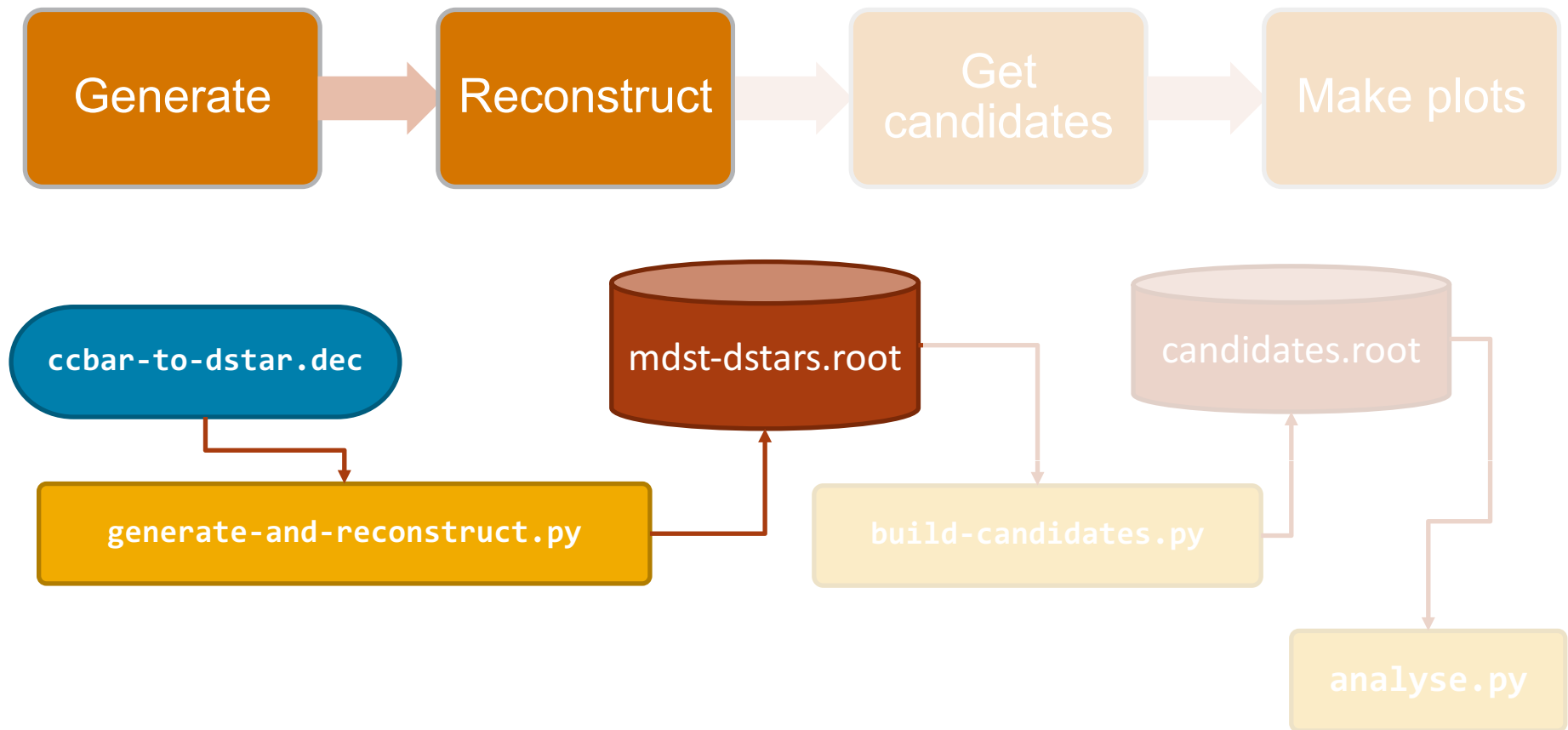
The basf2 steps

Scripts in the directory

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

Generate → Reconstruct → Get candidates → Make plots

ccbar-to-dstar.dec

mdst-dstars.root

candidates.root

generate-and-reconstruct.py

build-candidates.py

analyse.py

The basf2 steps
Scripts in the directory

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

Generate → Reconstruct → Get candidates → Make plots

ccbar-to-dstar.dec → generate-and-reconstruct.py → mdst-dstars.root → build-candidates.py → candidates.root → analyse.py

EvtGen decay file

▶ Belle II uses EvtGen to take cc continuum generated by PYTHIA into specific final-state hadrons

- User defined 'decfile'
- Falls back on **'DECAY_BELLE2.dec'** containing all the SM processes, measured branching fractions etc

▶ Look at **ccbar-to-dstar.dec**

- Note that the relevant stuff is line 41 onwards.
- Jet-Set parameters beyond the scope of this
  - Only tweak if you're an expert.

```
41 # ----- define D* decays for K/pi samples -----
42
43 Alias MyD0 D0
44 Alias MyAntiD0 anti-D0
45 ChargeConj MyD0 MyAntiD0
46
47 Decay D*-
48 1.000 MyAntiD0 pi-        VSS;
49 Enddecay
50
51 Decay D*+
52 1.000 MyD0 pi+            VSS;
53 Enddecay
54
55 Decay MyD0
56 1.000 K- pi+              PHSP;
57 Enddecay
58 CDecay MyAntiD0
59
60 End
```
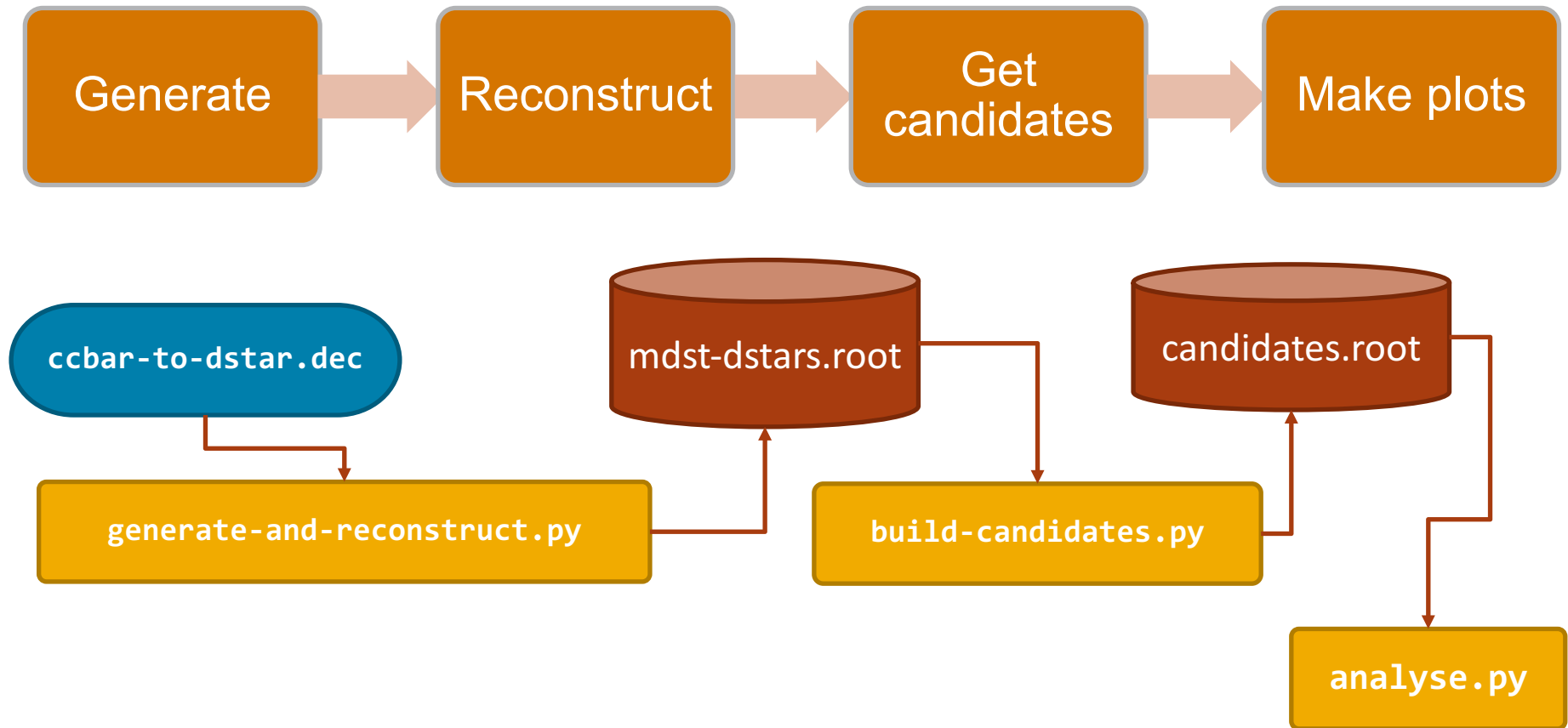ccbar-to-dstar.dec

Generate and reconstruct decays

▶ Look at **`generate-and-reconstruct.py`**

▶ Adds standard reconstruction and simulation modules (in the correct order)
`add_simulation()`
`add_reconstruction()`

▶ **Exercise**: find these functions in the software…

■ Hints: $BELLE2_RELEASE_DIR or stash.desy.de

▶ When you're happy, run it:

```
$ basf2 generate-and-reconstruct.py -n 10 # if this works then try 1k
$ basf2 generate-and-reconstruct.py -n 1000
```
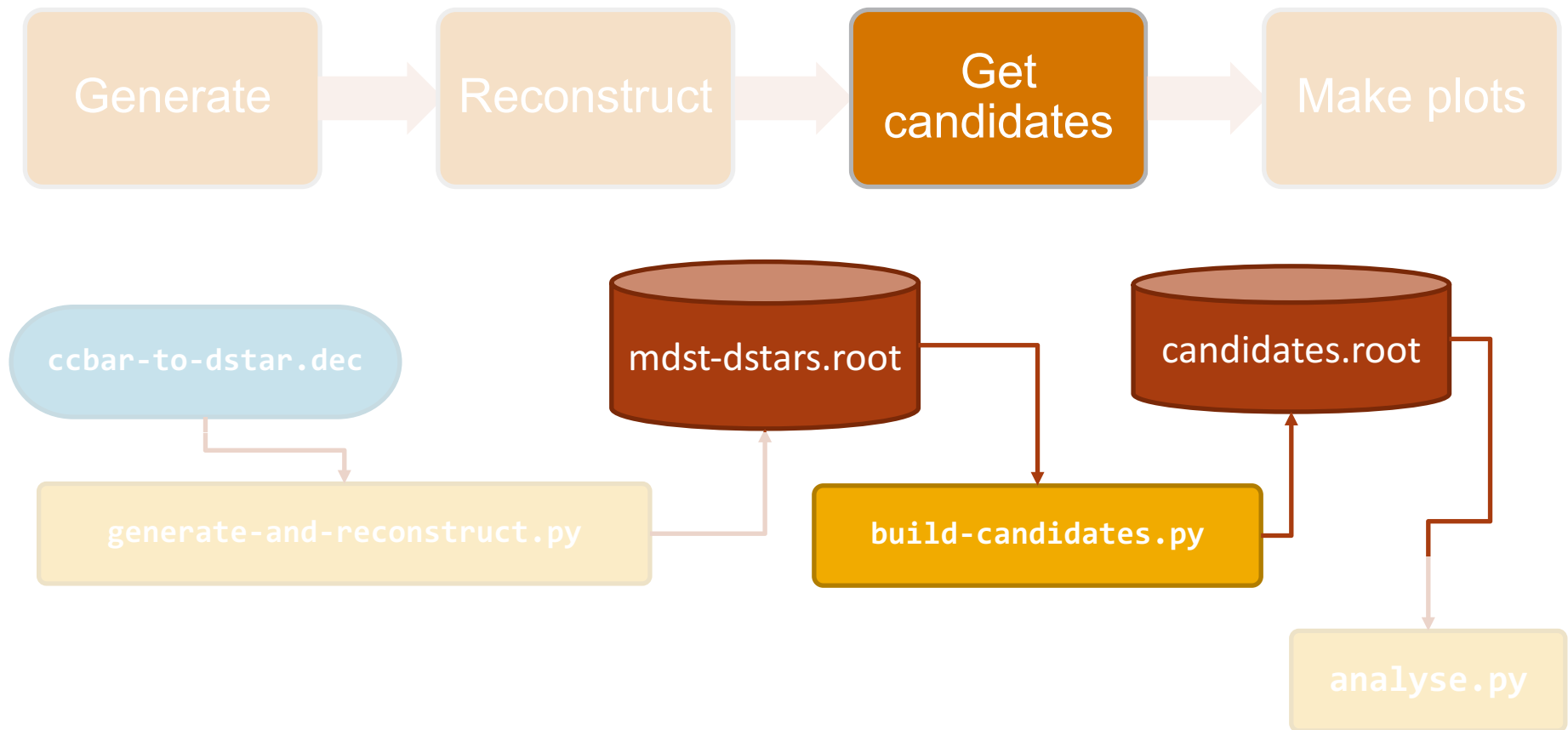
The `basf2` steps

Scripts in the directory

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

Generate → Reconstruct → Get candidates → Make plots

`ccbar-to-dstar.dec`

`mdst-dstars.root`

`candidates.root`

`generate-and-reconstruct.py`

`build-candidates.py`

`analyse.py`

The basf2 steps
Scripts in the directory

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

Get candidates from the mdst

► Do an **ls** in the local directory… see **mdst-dstars.root** ?

► This file format is **mDST** = **m**ini **D**ata **S**ummary **T**able

  ■ Nicer to have a flattened **TTree** to easily make plots.

  ■ https://confluence.desy.de/display/BI/Main+Glossary

► In **basf2** we need to get candidates from the mDST

► Look at **build-candidates.py**

► When you're happy, run it:

```
$ basf2 build-candidates.py --input mdst-dstars.root
```

► Then look at the output:

```
$ root candidates.root
root [0] TBrowser browser
```

Run over a skim (or a larger file)

▶ Now, we've only generated 1k events. Which is not enough statistics.

▶ I've run a job with 50k events.

```
$ basf2 build-candidates.py --input ~scunliff/public/mdst-dstars.root
```
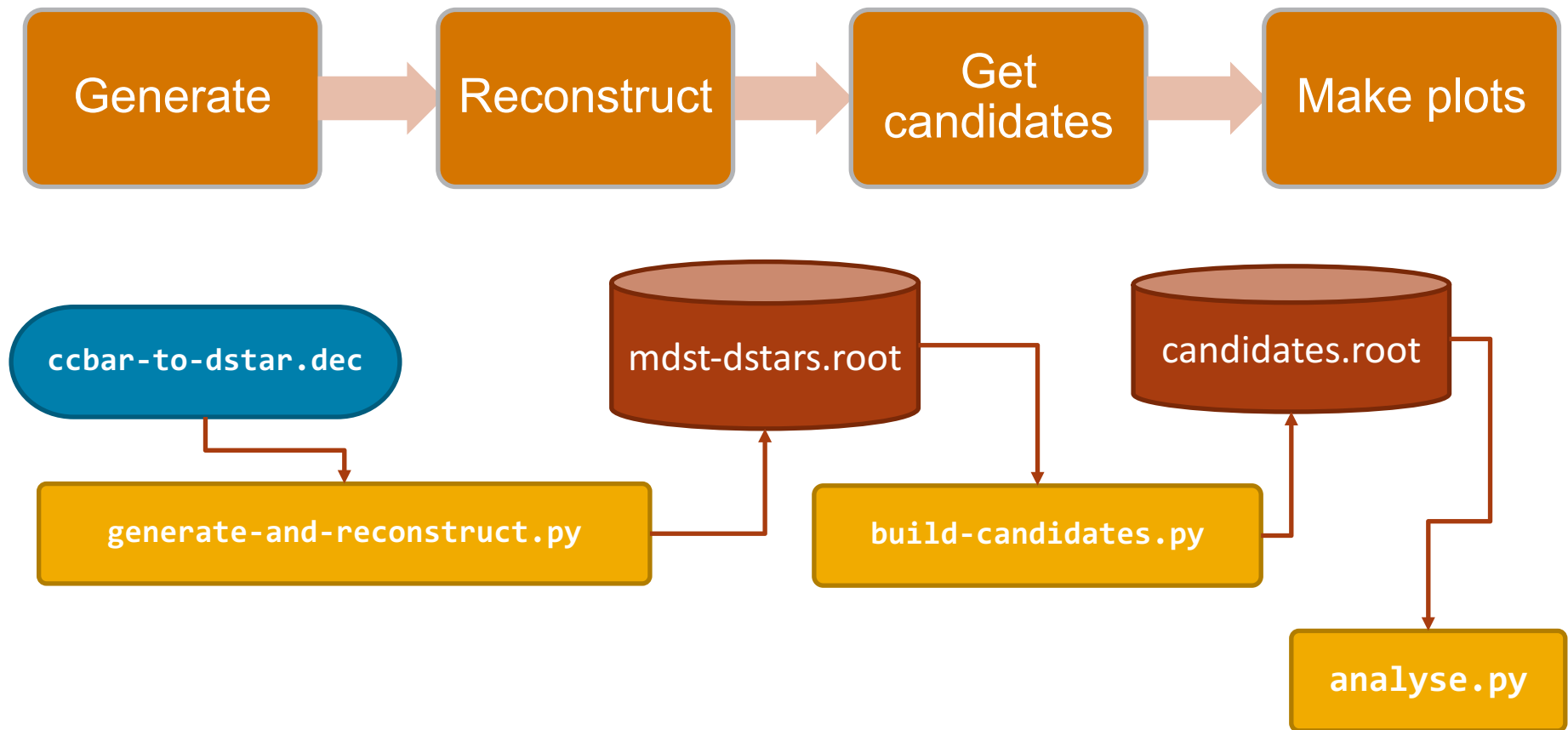
▶ So please run over that.

▶ (At time of writing) skims prepared by Racha are not quite ready.

▶ If they are, we should run over those instead/ as well

```
$ basf2 build-candidates.py --input ~rachac/skim/outputFiles/*.root
```
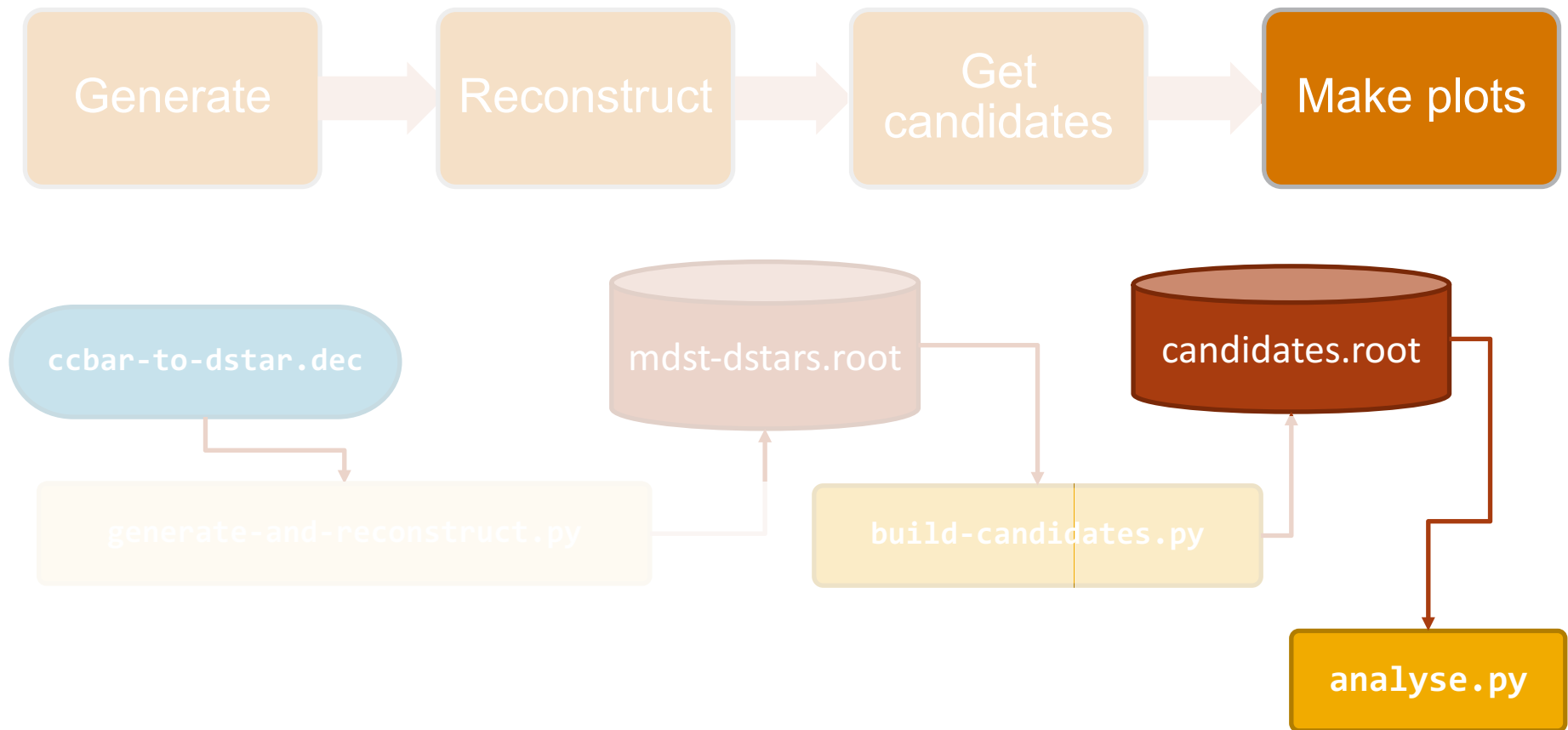
The basf2 steps

Scripts in the directory

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

The `basf2` steps
Scripts in the directory

https://stash.desy.de/users/scunliff/repos/b2gm-hadronpid-tutorial-jun2017/browse

```
Generate  →  Reconstruct  →  Get candidates  →  Make plots
```

ccbar-to-dstar.dec

mdst-dstars.root

candidates.root

generate-and-reconstruct.py

build-candidates.py

analyse.py

Run the plotting script to analyse

► Look at **`analyse.py`**

► When we're happy, run it:

```
$ python3 analyse.py
```

► Then look at the output:

Extra credit

► Can you modify `analyse.py` to also check the efficiency as a function of inclination (cosθ)? And azimuthal/radial angle φ?

  ■ Look at the plot. Does the shape make sense?

  ■ Where is the iTOP detector in this variable? Where does the ARICH start?

► How about code to make a ROC curve?

  ■ Hint: the validation scripts are a good place to look for example code