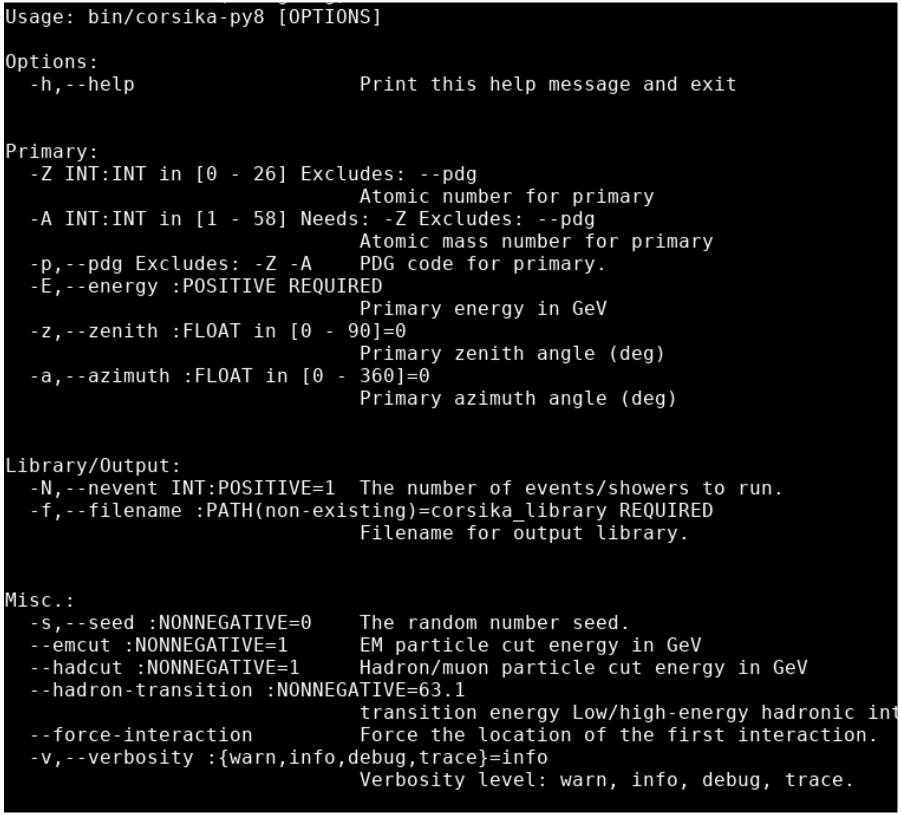
Steering / Output

Maximilian Reininghaus
CORSIKA 8 Workshop
2022-07-13

Steering

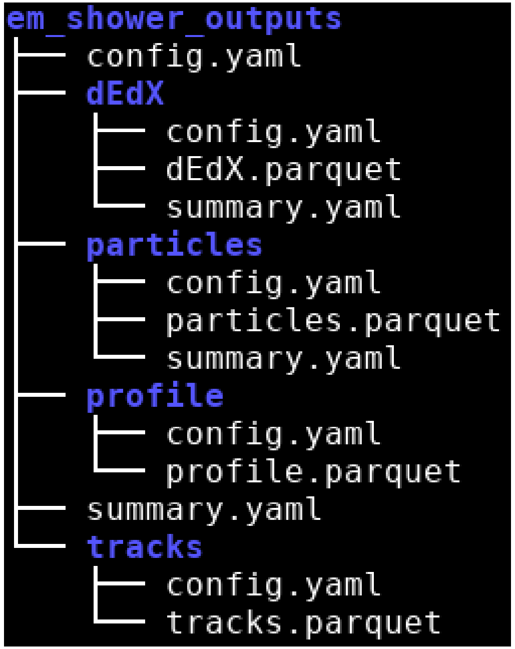
- process sequence fixed at compile time
- most settings used to be hard-coded in examples
- now via cmd-line parameters
- not everything covered yet
- CLI11 library, supports reading from config file, too



```
Usage: bin/corsika-py8 [OPTIONS]

Options:
  -h,--help                   Print this help message and exit


Primary:
  -Z INT:INT in [0 - 26] Excludes: --pdg
                              Atomic number for primary
  -A INT:INT in [1 - 58] Needs: -Z Excludes: --pdg
                              Atomic mass number for primary
  -p,--pdg Excludes: -Z -A    PDG code for primary.
  -E,--energy :POSITIVE REQUIRED
                              Primary energy in GeV
  -z,--zenith :FLOAT in [0 - 90]=0
                              Primary zenith angle (deg)
  -a,--azimuth :FLOAT in [0 - 360]=0
                              Primary azimuth angle (deg)


Library/Output:
  -N,--nevent INT:POSITIVE=1  The number of events/showers to run.
  -f,--filename :PATH(non-existing)=corsika_library REQUIRED
                              Filename for output library.


Misc.:
  -s,--seed :NONNEGATIVE=0    The random number seed.
  --emcut :NONNEGATIVE=1      EM particle cut energy in GeV
  --hadcut :NONNEGATIVE=1     Hadron/muon particle cut energy in GeV
  --hadron-transition :NONNEGATIVE=63.1
                              transition energy Low/high-energy hadronic int
  --force-interaction         Force the location of the first interaction.
  -v,--verbosity :{warn,info,debug,trace}=info
                              Verbosity level: warn, info, debug, trace.
```

Output

- docs for users
- combination of directories, yaml and parquet files
- yaml for metadata (human readable)
- parquet for binary, tabular data
- single files for multiple showers per run
- output of all input variables still missing! (reproducibility!)
- logging to stdout



```
em_shower_outputs
├── config.yaml
├── dEdX
│   ├── config.yaml
│   ├── dEdX.parquet
│   └── summary.yaml
├── particles
│   ├── config.yaml
│   ├── particles.parquet
│   └── summary.yaml
├── profile
│   ├── config.yaml
│   └── profile.parquet
├── summary.yaml
└── tracks
    ├── config.yaml
    └── tracks.parquet
```

Architecture

- `OutputManager(directory)`

  `-.add("subdir", someWriter)`

- `Writer`
  - `startOfLibrary(subdirectory)`
  - `startOfShower(showerID)`
  - `endOfShower(showerID)`
  - `endOfLibrary()`
- `Cascade(…, outputManager)`
  - informs `OutputMananger` about start/end of shower, which then calls `Writer` callbacks

Other formats

- Radio stores signal traces in numpy (.npz) archive
- boost::histogram serialization into .npz
  - Python module to de-serialize into `boost_histogram.Histogram`

Issues / disucssion

- shower library: N showers / 1 run vs. 1 shower / N runs have different output
- parquet heavy dependency (arrow) (in C++ and Python)
- parquet for more than 2 dim?
- Python: what if file size > memory? chunked automatically when reading/analysing?