"Array-Oriented Python Interface for the Pythia Event Generator"

Presented by: Artem Havryliuk

Mentors: Jim Pivarski

                Philip James Ilten

What is Pythia?

- *Purpose*: Simulates outcomes of high-energy particle collisions for theoretical and experimental physics

- *Applications*: Used in particle physics research, experimental data analysis, and model testing.

- *Languages*: Fortran and C++

- distributed as open-source software, allowing users to modify and extend its capabilities

What is the Problem?

- **Need for easy and quick installation**

  - It is necessary to make Pythia installable via pip install pythia8 by providing pre-built wheels.
  - it is necessary to reduce the size of the wheels

- **Inefficiency with Large Datasets**

  - Pythia's existing Python interface processes data one event at a time.

- **Need for Enhanced Data Handling**

  - Current interface limitations hinder the ability to leverage Pythia effectively in interactive environments such as Jupyter notebooks.
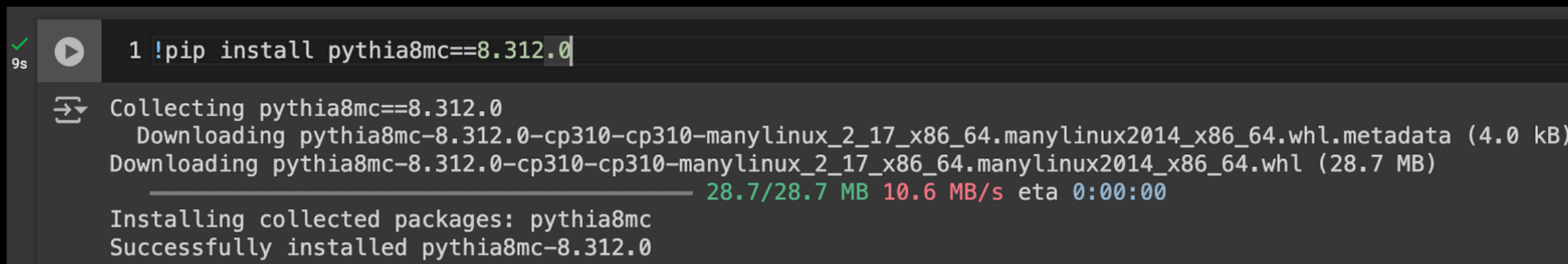
Our plans

- **Develop Array-Oriented Interface:**

  - Create a new Python interface for Pythia leveraging the Awkward Array library.

  - Focus on optimizing for speed and efficient data handling.

- **Enhance Usability**

  - Streamline feature set to improve user experience in interactive environments like Jupyter notebooks.

  - Ensure the interface supports intuitive, array-based operations on large datasets.

Achievements

- Enabled installation of Pythia with pip install pythia8 using pre-built wheels, allowing for quick setup in Google Colab (under 10 seconds).

- Reduced the wheel size significantly—from over 100 MB to 25 MB.

- Integrated Awkward Arrays into the nextBatch() function, enabling users to specify the desired number of events and receive them in a lightweight Awkward Array format.

Example of using nextBatch function

```python
import pythia8 # type: ignore
py = pythia8.Pythia("", False)
py.readString("SoftQCD:all = on")
number_of_events = 3
test_array = py.nextBatch(number_of_events)
```

```
type: 3 * var * {
    id: int32,
    status: int32,
    mother1: int32,
    mother2: int32,
    daughter1: int32,
    daughter2: int32,
    col: int32,
    acol: int32,
    m: float64,
    scale: float64,
    pol: float64,
    tau: float64,
    vProd: ?{
        px: float64,
        py: float64,
        pz: float64,
        E: float64
    },
    p: {
        px: float64,
        py: float64,
        pz: float64,
        E: float64
    }
}
```

- Using nextBatch() function to generate 3 events

- structure of the created Awkward Array

Content and structure of one of the arrays

```
type: 57 * {
    id: int32,
    status: int32,
    mother1: int32,
    mother2: int32,
    daughter1: int32,
    daughter2: int32,
    col: int32,
    acol: int32,
    m: float64,
    scale: float64,
    pol: float64,
    tau: float64,
    vProd: ?{
        px: float64,
        py: float64,
        pz: float64,
        E: float64
    },
    p: {
        px: float64,
        py: float64,
        pz: float64,
        E: float64
    }
}
```

```
[{id: 90, status: -11, mother1: 0, mother2: 0, daughter1: 0, ...},
 {id: 2212, status: -12, mother1: 0, mother2: 0, daughter1: 3, ...},
 {id: 2212, status: -12, mother1: 0, mother2: 0, daughter1: 4, ...},
 {id: 2212, status: 14, mother1: 1, mother2: 0, daughter1: 0, ...},
 {id: 9902210, status: -15, mother1: 2, mother2: 0, daughter1: 5, ...},
 {id: 990, status: -13, mother1: 4, mother2: 0, daughter1: 15, ...},
 {id: 2212, status: -13, mother1: 4, mother2: 0, daughter1: 16, ...},
 {id: 21, status: -21, mother1: 15, mother2: 15, daughter1: 9, ...},
 {id: 1, status: -21, mother1: 16, mother2: 16, daughter1: 9, ...},
 {id: 21, status: -23, mother1: 7, mother2: 8, daughter1: 17, ...},
 ...,
 {id: 211, status: 91, mother1: 30, mother2: 0, daughter1: 0, ...},
 {id: -211, status: 91, mother1: 30, mother2: 0, daughter1: 0, ...},
 {id: 221, status: -91, mother1: 30, mother2: 0, daughter1: 55, ...},
 {id: 22, status: 91, mother1: 38, mother2: 0, daughter1: 0, ...},
 {id: 22, status: 91, mother1: 38, mother2: 0, daughter1: 0, ...},
 {id: 22, status: 91, mother1: 42, mother2: 0, daughter1: 0, ...},
 {id: 22, status: 91, mother1: 42, mother2: 0, daughter1: 0, ...},
 {id: 22, status: 91, mother1: 50, mother2: 0, daughter1: 0, ...},
 {id: 22, status: 91, mother1: 50, mother2: 0, daughter1: 0, ...}]
```

Next Steps

- Test and troubleshoot newly added features

Thank you!