LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS
*partículas e tecnologia*

# LIP School on
# Data science in (astro)particle physics
# - data challenge -

12-14 March 2018

*Nuno Castro*
*on behalf of the organizing team*

Universidade do Minho
Escola de Ciências

IF INVESTIGADOR FCT
FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR
IF/00050/2013/CP1172/CT0002

# School on Data Science in (astro)particle physics
## mini-data challenge

- In order to exercise some of the ML techniques presented in the school, we would like to propose a mini-data challenge
  - hands-on
  - hopefully, it will be fun - that's one of the goals!

- Organize yourself in a team of 2-3 persons
  - choose your favorite ML method
  - develop it as much as you can
  - present it informally (Wednesday afternoon)

# School on Data Science in (astro)particle physics
## mini-data challenge

- open gitlab repository with examples:
  - https://gitlab.cern.ch/nfcastro/LIP-2018-DataChallenge
- large datasets available for training, testing and application
  - https://cernbox.cern.ch/index.php/s/McixV1W54X5B5l0
  - ~3 Gb: we can also pass it to you on a pen, if you prefer
- the data challenge aims to distinguish jets originating from gluons and from quarks
  - but the goal is mostly technical: the focus is not on the physics of it

# School on Data Science in (astro)particle physics
## mini-data challenge

### Challenge Rules

- The challenge lasts for the entire duration of the 2018 school on data sciences, organized by LIP
- It is allowed to form teams
- You can use any external computing resources you may have at your disposal
- The ranking will be based on the area under the ROC curve (AUC), estimated on the modified sample for a model trained on the standard sample.
- The modified sample data can be used in the training **as long as the true labels of the modified sample are not used in the training**
- Results should be presented by each team during the last hands-on session (Wed 14th)

# School on Data Science in (astro)particle physics
## mini-data challenge

### The data format

The data are provided as a simple root tree format, with no external dependencies. Each entry of the tree is a jet. The tree contains a few jet-level variables and an array of constituents (tracks and towers).

The examples discussed below show how to read this tree format using either pyroot or ROOT/C++.

This is the definition of the tree:

```
treeOut->Branch("jetPt"  ,&jetPt  , "jetPt/F");
treeOut->Branch("jetEta" ,&jetEta , "jetEta/F");
treeOut->Branch("jetPhi" ,&jetPhi , "jetPhi/F");
treeOut->Branch("jetMass",&jetMass, "jetMass/F");

treeOut->Branch("ntracks",&ntracks,"ntracks/I");
treeOut->Branch("ntowers",&ntowers,"ntowers/I");

treeOut->Branch("trackPt"     , trackPt    ,"trackPt[ntracks]/F");
treeOut->Branch("trackEta"    , trackEta   ,"trackEta[ntracks]/F");
treeOut->Branch("trackPhi"    , trackPhi   ,"trackPhi[ntracks]/F");
treeOut->Branch("trackCharge" , trackCharge,"trackCharge[ntracks]/F");
treeOut->Branch("towerE"      , towerE     ,"towerE[ntowers]/F");
treeOut->Branch("towerEem"    , towerEem   ,"towerEem[ntowers]/F");
treeOut->Branch("towerEhad"   , towerEhad  ,"towerEhad[ntowers]/F");
treeOut->Branch("towerEta"    , towerEta   ,"towerEta[ntowers]/F");
treeOut->Branch("towerPhi"    , towerPhi   ,"towerPhi[ntowers]/F");
```

# School on Data Science in (astro)particle physics
# mini-data challenge

## Location of the data sets

The datasets are available in this link, which gives you access to the following directories:

```
quarks_standard
gluons_standard
quarks_modified
gluons_modified
```

The training of your method should be based solely on the Standard data set, while the score which will be used for the challenge ranking should be based on the Modified one. The idea is that in real-life you would train your method on Monte Carlo, and then apply it to real data (and in general there are small differences between data and MC). Techniques to minimize the effect of these difference are therefore very welcome.

## How to develop code for the challenge and examples

We provide a few examples (in ROOT/C++ and python) which show you how to get started.

The methods which we implemented in the examples are simple and not computationally expensive.