# nndm library

**Andrés Felipe Gómez, Jose Miguel Muñoz Arias, Esteban Vásque**

**May 28, 2022**

# CONTENTS:

CHAPTER

# ONE

# READFILEBASE

class nndm_library.**ReadFileBase**(*path*, *recursive=False*, *ext='.txt'*, *relabel_events=True*)

Class to read the labeled data coming in a format like the following:

data1 data2 data3

v11 v12 v13

. . .

. . .

. . .

vn1 vn2 vn3

where data1, data2, … respresent names and vij, a value in the given i row and column dataj.

**Parameters**

- **path** (*string*) – the direction to the file containing all the events information.
- **recursive** (*bool*) – read all the .lhe files found in all paths inside a given files_dir
- **ext** (*str*) – extension of the files to read
- **relabel_events** (*bool*) – there is an id for each possible event. For instance a collision have an id for it and two sub ids for the particle that interact in it. When relabel_events is True, the values of id are associated unequivocally with each event.

**Variables**

- **data** – dataframe with the read events
- **files_dir** – directory with the name of the files read and its id

**add_angle**(*axes=['px', 'py', 'pz']*, *angle_axis='pz'*)

Calculate the angle of the particles starting from a list of the form [px, py, pz]. This with respect to the "axis" element.

**extract_params_from_path**()

Format is as follows: {particle_name}_{param1}_{value1}_{param2}_{value2}_{param3}_{value3}*.lhe An example would be eta_decay_events_mk_0.38_eps2_5.404557191441203e-07.lhe.

**Returns** dictionary with all extracted data

CHAPTER

TWO

# READLHE

class nndm_library.**ReadLhe**(*path*, *particle_ids=None*, *var_of_interest=None*, *outgoing=False*, *recursive=False*, *relabel_events=True*, *verbose=1*)

Class to read the data coming in lhe format. By default it will read all the particles. Filters used apply to the such default data.

**Parameters**

- **path** (*string*) – the direction to the file containing all the events information.
- **partcile_ids** (*list of integers*) – ids of the particles to extract from the file according to the pdg, By default: None, which means exctract all the particles.
- **var_of_interest** (*list of strings*) – names of the variables to extract from the lhe. eg. [“e”,”angle”], [“e”,”px”,”py”] .... By default: None, which means exctract all the variables.
- **outgoing** (*bool*) – filtrate to obtain all the outgoing particles
- **files_dir** (*string*) – directory where the files are to be found
- **recursive** (*bool*) – read all the .lhe files found in all paths inside a given files_dir
- **verbose** (*bool*) – show progress reading all the .lhe files

**Variables**

- **data** – dataframe with the read events
- **files_dir** – directory with the name of the files read and its id

**add_angle**(*axes=['px', 'py', 'pz']*, *angle_axis='pz'*)

Calculate the angle of the particles starting from a list of the form [px, py, pz]. This with respect to the “axis” element.

**extract_params_from_path**()

Format is as follows: {particle_name}_{param1}_{value1}_{param2}_{value2}_{param3}_{value3}*.lhe An example would be eta_decay_events_mk_0.38_eps2_5.404557191441203e-07.lhe.

**Returns** dictionary with all extracted data

CHAPTER

# THREE

# READROOT

class nndm_library.**ReadRoot**(*path: str*, *output_base_tree='treeout'*, *pattern_output='first'*, *output_base_middle_branch='/e/out'*, *leafs=['out.t', 'out.x', 'out.y', 'out.z', 'out._mass']*, *recursive=False*, *files_dir=None*, *relabel_events=True*)

Class to read the labeled data coming in ROOT format. By default it assumes values for output_base_tree, pattern_output, output_base_middle_branch, and leafs. This is for a fast reading.

**Parameters**

- **path** (*str*) – the direction to the root file(s)
- **output_base_name** (*str*) – Name bas of the first node of the tree that has the data. For instance, if the base name is treeout, there options could be treeout1, treeout2, …., treeoutN.
- **pattern_output** (*str*) – The idea is this parameter define a methodology to choose from the possible first nodes that have a given output_base_name. As an example, first would choose treeoout1 in the example before.
- **output_base_middle_branch** (*str*) – middle branch that goes after the selected first node chosen by the output pattern. If this variable is "e/out", following the example the tree to consult at the moment would be treeout1/e/out/.
- **leafs** (*list of strings*) – what are the leafs to exaplore in the actual branch. If out.a is the ouput name for the a momenta, giving a list [out.x, out.y] will give the data to consult. That is, treeout1/e/out/out.x and treeout1/e/out/out.y
- **relabel_events** (*bool*) – there is an id for each possible event. For instance a collision have an id for it and two sub ids for the particle that interact in it. When relabel_events is True, the values of id are associated unequivocally with each event.

**Variables**

- **data** – dataframe with the read events
- **files_dir** – directory with the name of the files read and its id

**add_angle**(*axes=['px', 'py', 'pz']*, *angle_axis='pz'*)

Calculate the angle of the particles starting from a list of the form [px, py, pz]. This with respect to the "axis" element.

**extract_params_from_path**()

Format is as follows: {particle_name}_{param1}_{value1}_{param2}_{value2}_{param3}_{value3}*.lhe An example would be eta_decay_events_mk_0.38_eps2_5.404557191441203e-07.lhe.

**Returns** dictionary with all extracted data

CHAPTER

# FOUR

# FILESMANIPULATOR

class nndm_library.**FilesManipulator**(*path*, *particle_ids=None*, *var_of_interest=None*, *outgoing=False*, *verbose=0*)

General methods intended for the manipulation of the files and its names in a given directory(ies).

**Parameters**

- **path** (*string*) – the direction to the file containing all the events information.
- **partcile_ids** – ids of the particles to extract from the file according to the pdg,

By default: None, which means exctract all the particles. :type partcile_ids: list of integers

**Parameters**

- **var_of_interest** (*list of strings*) – names of the variables to extract from the lhe. eg. [“e”,”angle”], [“e”,”px”,”py”] .... By default: None, which means exctract all the variables.
- **outgoing** (*bool*) – filtrate to obtain all the outgoing particles

**Variables** **scan** – it is a dictionary with the values of the benchmark points, each with its respective result sets of vaues of the variable of interest (var_of_interest). Note that the benchmark points are read from the name. Format is as follows: {particle_name}_{param1}_{value1}_{param2}_{value2}_{param3}_{value3}*.lhe An example would be eta_decay_events_mk_0.38_eps2_5.404557191441203e-07.lhe. Here we have pictorical description of the scan:

First, a list of ints is [int, int, ...] == [(int)]. So a list of a list of floats is: [[(float)], [(float)], ...] == [( [(float)] )] {id: [(int)], typ: [(str)], mk: [(float)], eps2: [(float)], px: [[(float)], [(float)], ...], py: [( [(float)] )], pz: [( [(float)] )] } Note that momentum and energy are a list of arrays, where each array correspons to a param point

CHAPTER

FIVE

# INDICES AND TABLES

- genindex
- modindex
- search

# INDEX