

nESSie - Event Shape Sorter

nESSie is Event Shape Sorting algorithm, which sort your events according to their similar shapes. The main advantage of this algorithm is that it takes into account all orders of anisotropies at the same time.

Running program

The program is running in Linux (tested on Ubuntu 16.04). To compile it, you can use Makefile and just type in terminal:

```
1 make
```

Then you can run program by typing:

```
1 ./main.exe
```

Input to program

The input file containing data should be in OSCAR1997A format. That means the file should looks like this

```
1 # OSC1997A
2 # final_id-p-x
3 # UrQMD
4 1 3223 8.547 0
5 1 -2212 -0.27254 0.0829967 0.31003 1.02817 0.938 -0.829524 5.68779 10.2909 32.7802
6 ...
7 2 2038 9.967 0
8 1 111 -0.279362 0.112469 10.1903 10.1957 0.138 5.02633 3.11084 199.578 200
9 ...
```

The first three lines are information about format and generator, so their are skipped. Then follows information about event: its number, number of particles in this event, impact parameter¹ and 0. After this line follows list of all particles in event with its properties: number, PDG particle code, p_1 , p_2 , p_3 , p_0 , mass of particle, x_1 , x_2 , x_3 and x_0 .

Input parameters

To change parameters of algorithm, use file *params.h*. The parameters you can set are:

```
1 const int64_t NoEvents = 10000;
2 const int NoAzibins = 20;
3 const int NoBins = 10;
4
5 const int RelevantParticles = 0;
6 const float RapidityCut = 1.0;
7 const int AnisotropyOrder = 2;
8 const float ptMin = 0.0;
9 const float ptMax = 10.0;
10 const int64_t MaxNoCycles = 1000;
11
12 const char* DataDirectory = "./";
13 const char* DataFile = "GENERATOR_events.out";
```

NoEvents - Number of events in datafile. This number has to be equal to actual number of events.

NoAzibins - To find out shape of event, algorithm uses azimuthal angle histograms. This parameter sets number of bins for azimuthal angle. More bins, better shape description, but also worse statistics. Default value: 20.

NoBins - Number of event classes, in which all events are splitted. Bigger the number is, classes become smaller and more similar, but again there is worse statistics. Default value: 10.

RelevantParticles - Simple switch which choose, which particles will be taken in account. Rotations of events and also sorting will assume only these particles. Possible values are:

¹which is not important for sorting algorithm

- 0 - all charged particles
- 1 - all charged pions
- 2 - all charged kaons
- 3 - all protons

RapidityCut - Cut on rapidity. Particles with higher rapidity will not be taken in account. Default value: 1.0.

AnisotropyOrder - First part of the program rotate all events to the same direction of $\vec{q}_n = (\langle \cos(n\phi) \rangle, \langle \sin(n\phi) \rangle)$. This parameter is simple the n , so all events are rotated according to n -th anisotropy. Default value: 2.

ptMin & ptMax - Cut on p_t in GeV. Default values are set large enough to accept all particles. Default values: 0.0 and 10.0.

MaxNoCycles - In some initial conditions, algoritm can't find the ideal shapes of classes and program loop forever. For these cases there must be some limit of how many cycles can be done. Default value: 1000.

DataDirectory - Path to the input datafile. Can be set relative to the program directory. Default value: `"/"`

DataFile - Name of the input datafile, depends on generator.

Output of program

Output of this program is located in directory results. All result files contain date and time in their names, so you don't have to worry that result files will be replaced. There are three types of result files:

- *yymmdd_hhmmss_id_events* - This is the order of the events after sorting process. The numbers in this files are original order numbers of events.
- *yymmdd_hhmmss_av_bin* - There are histograms of average events for each class. That means you will find here *NoBins* columns and *NoAzibins* rows of data. For better plotting (in gnuplot for example) the first column contains middle azimuthal angle for each azimuthal bin.
- *yymmdd_hhmmss_bin_i* - Each of these files contain list of all events in class, their azimuthal histograms and also mean class number μ , which is the last number in each row.
- *yymmdd_hhmmss_rotations* - In this file you'll find directions of q_n vector for each event. If you want to rotate an event, you have to rotate all particles by an angle $\frac{q_n}{\text{AnisotropyOrder}}$.