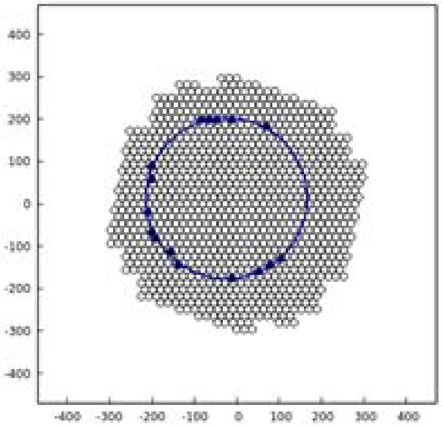
Exams:

- Students should prepare a software/data analysis project in groups of 2 (+-1) students following these steps:
  - Present the project abstract
    - 1 month in advance if the abstract is proposed by the students (it must be approved by the teachers)
    - 2 weeks in advance if the abstract is chosen among the list below.
  - Present a written document with the description of the project and the results obtained (max 2 pages of text, plus any number of figures):
    - 1 week before the exam
  - Source code of the project (puboshed on github or similar):
    - 1 week before the exam
- Exams dates
  - Around second half of january and second half of february
    - To be agreed with the teachers
    - For the first date, please present abstracts before christmas break
- Oral exams with questions both on the presented project and on the topics discussed in the course
  - The following elements will be evaluated:
    - Complexity of the tools used in the project
    - Quality of code, documentation and description of the project
    - Knowledge of the topics presented in the lectures

Abstracts:

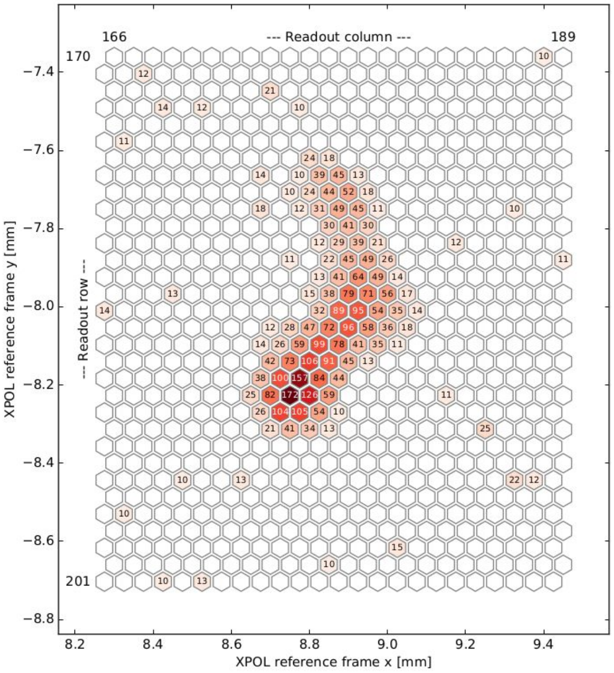
1) Demonstrate the usage and the advantage of Graph Networks on a dataset of your choice (dataset should be discussed with the teacher)
   a) E.g. start from https://github.com/deepmind/graph_nets
2) Take part in one of the kaggle.com competitions (even a finished one) or use a kaggle.com dataset and compare your results with the one from the score leader. Analyze the differences in your approach wrt the score leader. The specific dataset/competition to use should be agreed with the teachers.
3) Implement an efficient parallel histogram algorithm for an input array of ASCII characters. There are 128 ASCII characters and each character will map into its own bin for a fixed total of 128 bins. Apply the algorithms to long text in three different languages (Italian, English, German for instance), then compare the results. Use the shared memory for each thread block, then atomically modifying the global histogram. Use python to read the long text and pyCUDA to implement a C/C++ kernel on GPU. Compare the performances of your parallel implementation with a serial



version of the code. Suggestion: use appropriate python modules to read pdf to avoid the use of long text in plain txt. (GL)

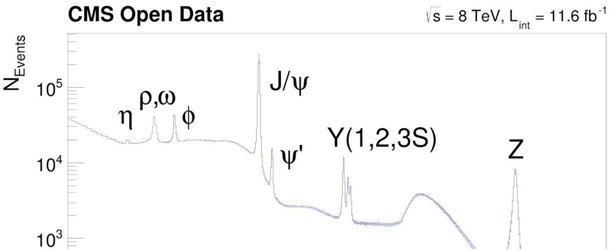
4) Implement a method to efficiently identify rings in a sparse matrix. Each ring is defined by a finite number of dots firing in a 2D array of possible positions (see figure). The data-set provided includes a large number of “events”. One or more rings are present in each event. The algorithm must be fast and accurate, a parallel implementation is strongly suggested. The resolution of the chosen method must be evaluated by comparing the results of the algorithm with the “real” center position and radius of the generated rings. (GL)
5) Develop a tool for a basic statistical exploration of the Fermi-LAT fourth source catalog (4FGL, available at https://fermi.gsfc.nasa.gov/ssc/data/access/lat/8yr_catalog/ in several different formats). The program should be able to display in a graphical fashion the basic characteristics of the sources in the catalog. In addition, use a classification technique of your choice to infer the source class (e.g., AGN or pulsar) for each entry in the catalog, and evaluate the performance of your classification scheme.
6) Explore the general problem of event reconstruction and/or visualization of low-energy photoelectron tracks from a gas-pixel detector for astronomical X-ray polarimetry (see figure). More specifically you can:
    a) Implement an (efficient) clustering algorithm to separate the track from the noise hits;
    b) Devise an algorithm to reconstruct the origin and initial direction of the track;
    c) Develop a track event display.

   (Pick one or more of the items above, and analyze the efficiency and performance of your implementation.) The input simulated datasets will provided in FITS format and will include the charge content (in ADC counts) for each of the pixels in the raw track image, along with the Monte Carlo truth for all the relevant quantities. Reference: http://glast.pi.infn.it/Papers/PIXIE/SPIE6266_102.pdf



HEP specific abstracts

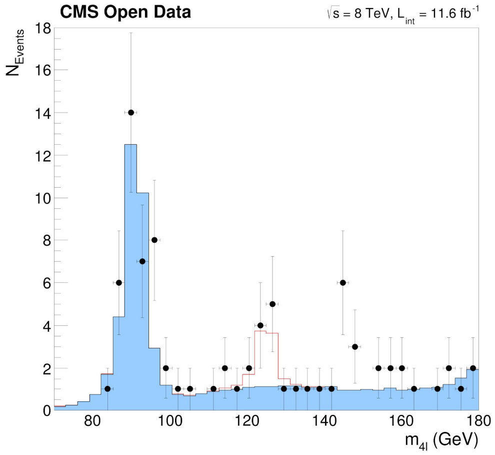
7) Use CMS di-muon open data to identify and fit the mass value of know dimuon resonances. Plot properties of the various resonances (e.g. transverse momentum).



The data analysis must be performed with computationally efficient code and the software organization should be such that people reviewing or reusing the software can understand it easily. Several measurements can be performed in different projects

a) Measure angular properties vs rapidity of the Z as a proxy to weak mixing angle (Eur. Phys. J. C 78 (2018) 701)
b) Resolving and fitting mass of upsilon states (https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsBPH12006)

8) Find the Higgs boson in the decay channel H->ZZ->4l with CMS Open data. Write an analysis program with the tools provided in the course. Extract a value for the most probable Higgs boson mass.



9) Take part to the LHC Olympics ! https://indico.cern.ch/event/809820/page/16782-lhcolympics2020-old

MedPhys specific abstracts

1) Design a segmentation algorithm for mass lesions in mammography. The segmentation procedure can be an improved version of the code shown in Lecture6 (e.g. you can apply image filters to enhance the shape of spiculated lesions). The quality of the newly implemented segmentation pipeline can be compared with the existing one in terms of a statistical

analysis (or machine learning classification) of image features aiming at distinguishing benign from malignant masses.

Reference paper: Delogu et al, Characterization of mammographic masses using a gradient-based segmentation algorithm and a neural classifier, Comput Biol Med. 2007 Oct;37(10):1479-9, https://www.ncbi.nlm.nih.gov/pubmed/17383623
Dataset*: DATASET/IMAGES/Mammography_masses/

2) Design a segmentation algorithm for mass lesions in mammography based on convolutional auto-encoders (CAE), extending the example shown in Lecture11. Choose the optimal dimensionality of the latent space that leads to maximum benign/malignant discrimination performance.

Reference paper (to be inspired): Have et al., Brain tumor segmentation with Deep Neural Networks., Med Image Anal. 2017 Jan;35:18-31, https://www.ncbi.nlm.nih.gov/pubmed/27310171
Dataset*: DATASET/IMAGES/Mammography_masses/large_sample_Im_segmented_ref/

3) Compare the performance of a CNN classification on the microcalcification image dataset (see Lecture10), with the performance obtained in an analysis pipeline where the mammographic images containing either microcalcifications or normal tissue are represented in terms of wavelet coefficients (see Lecture4, and choose the optimal family and level of decomposition).

Reference paper: Retico et al, A scalable computer-aided detection system for microcalcification cluster identification in a pan-European distributed database of mammograms, NIM A, Volume 569, Issue 2, 20 December 2006, https://www.sciencedirect.com/science/article/pii/S0168900206015439
Dataset*: DATASET/IMAGES/Mammography_micro (data folders are already partitioned in train and test sets)

4) Implement an autoencoder to compress the grey matter image segments obtained from the brain MRIs of a cohort of subjects with Alzheimer's disease and control subjects. Classify the subjects using the latent space representation. Optimize the dimension of the latent space according to the case/control discrimination performance.

Reference paper: Retico et al, Predictive Models Based on Support Vector Machines: Whole-Brain versus Regional Analysis of Structural MRI in the Alzheimer's Disease. J Neuroimaging. 2015 Jul-Aug;25(4):552-63, https://www.ncbi.nlm.nih.gov/pubmed/25291354

Dataset*: DATASET/IMAGES/AD_CTRL/
(Due to the huge size of the dataset you can work on one or few slices crossing the hippocampus).
The same approach is applicable both to the Mammography_masses and to Mammography_micro the dataset.

5) Predict the age of the healthy subjects of the ABIDE cohort comparing different regression models. Evaluate the reproducibility of the results across the different acquisition sites.

Reference paper: Cole et al., Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker, NeuroImage 163 (2017) 115–124, https://www.ncbi.nlm.nih.gov/pubmed/28765056
Dataset*:
DATASET/FEATURES/Brain_MRI_FS_ABIDE/FS_features_ABIDE_males.csv

6) Improve the standard implementation of the cross-validation (CV) method, extending the train-test splitting procedure so that it preserves the matching for confounding variables (e.g. age, gender, FIQ, site) and/or it keeps in the same partitions data from the same subject (i.e. the 4 mammographic projections of the same subject).

(Extend the Matlab cvpartition class or implement a function, extend the train_test_split of the sklearn.model_selection module)
Reference code: https://it.mathworks.com/help/stats/cvpartition.html, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
Reference paper: W Luo et at., Guidelines for Developing and Reporting Machine Learning Predictive Models in Biomedical Research: A Multidisciplinary View, J Med Internet Res 2016;18(12):e323, https://www.jmir.org/2016/12/e323
Dataset*:
DATASET/FEATURES/Brain_MRI_FS_ABIDE/FS_features_ABIDE_males.csv
(You can also build a synthetic dataset)

7) Develop and orthonormal viewer for DICOM or NIfTI images in Matlab/python; add a number of functionalities of interest, e.g. show coordinates at the cursor position (in physical units), show image intensity, display L/R information, define whether images are displayed in radiological/neurological orientation. Get inspired by Mango viewer functionalities. You can add advanced and useful features, e.g. to allow manual drawing and storing of ROI masks.

Resources: start from and improve the
https://it.mathworks.com/help/images/ref/orthosliceviewer.htm

* The datasets shared in the folders listed below are available only for the learning purposes related to this course and for the final examination. For any further possible use, ask A. Retico to get information about permissions for data reuse.
Datasets are available to download on the shared repositories:
1) INFN Pandora, https://pandora.infn.it/public/cmepda/
2) Google Drive,
https://drive.google.com/open?id=1YqK7ZkM-P2IrqfD7Pj-SCmjz-GWd_1-Y
For each proposed exercise, the subfolders and file names are indicated.