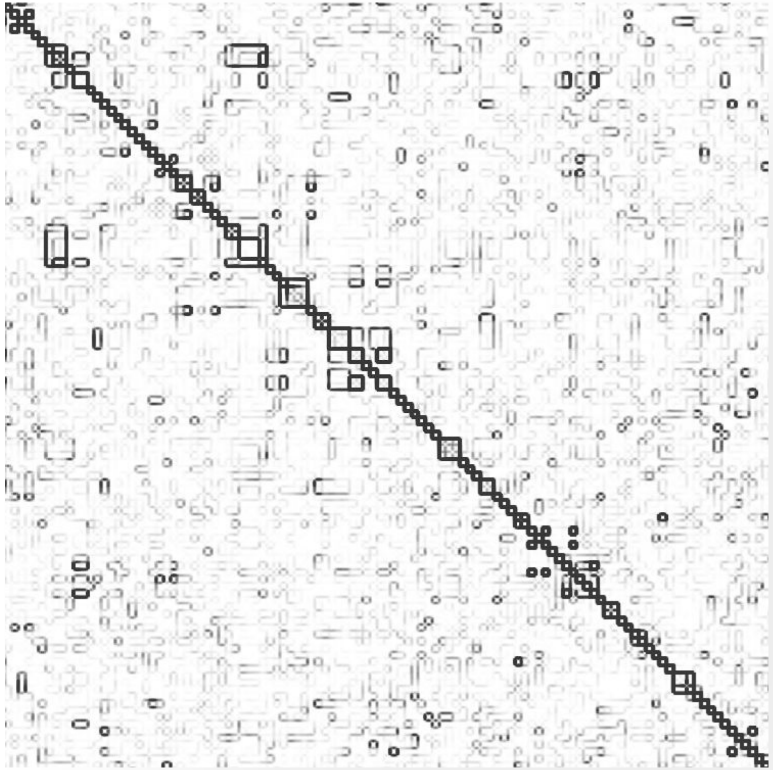


Modernising CMS Trigger Rate Monitoring tools

February 26, 2020
Antonio Vivace
on behalf of G. Dirkx, K. Mohrman, A. Wightman, J.Lawrence
avivace@cern.ch
Università di Milano Bicocca

References, Links

- RateMon repository: gitlab.cern.ch/avivace/ratemon, https://github.com/cms-tsg-fog/RateMon
- RateMon experimental API: gitlab.cern.ch/avivace/ratemon-api
- RateMon experimental UI: gitlab.cern.ch/avivace/ratemon-ui
- Test instances: brandeis.cern.ch
- A. Thea – "**Introduction to trigger concepts**" ISOTDAQ 2019
- A.A.Pol, G. Cerminara – "**Machine Learning Anomaly Detection Applications to Compact Muon Solenoid Data Quality Monitoring**"
- Geoffrey N Smith, Charles N Mueller, Andrew S Wightman – "**Tools for Trigger Rate Monitoring at CMS**"

Overview

RateMon scripts provide a “transparent” way to access a lot of data:

- PrescaleMatrix(HLT_Key)
- Find Runs with specified trigger keys
- Find Fits (linear, quad, quad2, cube, exp, sinh)
- Generate and export plots
- Compare menu rates with google docs
- Compare rates with saved fits
- Sends audio/email alerts,..
- Rate normalisations for prescales and PU
- **Rate Monitor**

Status

- Very old (first twiki revision in 2011, last updated 2018)
- Runs only on LXPLUS, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
- Fits are ROOT objects, serialised with pickle
- Output are static renders of the plots
- Configuration is a python class
- No packaging, no CI/CD
- Must be restarted manually
- Python2
- Data needs to be plugged into an OMS panel (WBM imported the PNGs) and readable by its plot engine (HighCharts)

Updates

- **Very old (first twiki revision in 2011, last updated 2018)**
  Updating the twiki page, preparing a clearer README/quickstart document, more verbose output
- Runs only on LXPLUS, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
- Fits are ROOT objects, serialised with pickle
- Output are static renders of the plots
- Configuration is a python class
- No packaging, no CI/CD
- Must be restarted manually
- Data needs to be plugged into an OMS panel (WBM imported the PNGs) and readable by its plot engine (HighCharts)
- Python2

Updates

- Very old (first twiki revision in 2011, last updated 2018)
- **Runs only on LXPLUS**, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
  Now runs on every CC7/C8 machine. There's a provided command to proxy the database connection trough LXPLUS with a SSH SOCKS5 proxy (when not on CERN network).
- Fits are ROOT objects, serialised with pickle
- Output are static renders of the plots
- Configuration is a python class
- No packaging, no CI/CD
- Must be restarted manually
- Python2
- Data needs to be plugged into an OMS panel (WBM imported the PNGs) and readable by its plot engine (HighCharts)

Updates

- Very old (first twiki revision in 2011, last updated 2018)
- Runs only on LXPLUS, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
- **Fits are ROOT objects, serialised with pickle**

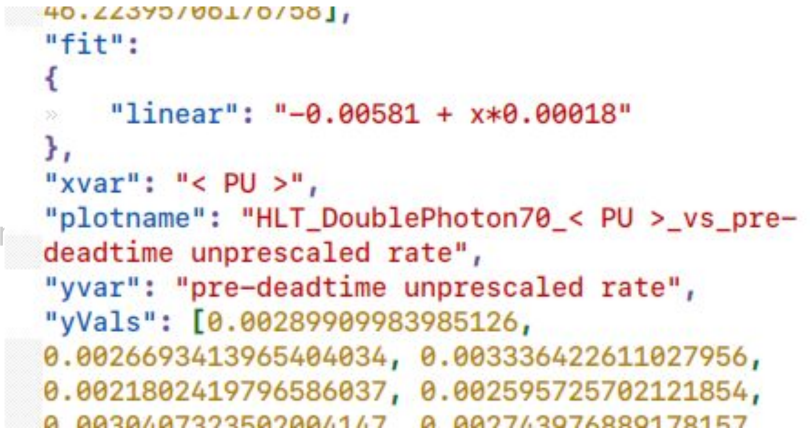
  Fits are now exported as readable and parse-able functions. Each value available separately

  `"linear": "-0.00581 + x*0.00018"`
- Output are static renders of the plots
- Configuration is a python class
- No packaging, no CI/CD
- Must be restarted manually
- Python2
- Data needs to be plugged into an OMS panel (WBM imported the PNGs) and readable by its plot engine (HighCharts)

Updates

- Very old (first twiki revision in 2011, last updated 2018)
- Runs only on LXPLUS, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
- Fits are ROOT objects, serialised with pickle
- **Output are static renders of the plots**

  JSON exports. Schema: X Values, Y Values, X Label, Y Label, fit function, Trigger Name
- Configuration is a python class
- No packaging, no CI/CD
- Must be restarted manually
- Python2
- Data needs to be plugged into an OMS panel (WBM i
  (HighCharts)



```
46.22395706176758],
"fit":
{
»   "linear": "-0.00581 + x*0.00018"
},
"xvar": "< PU >",
"plotname": "HLT_DoublePhoton70_< PU >_vs_pre-
deadtime unprescaled rate",
"yvar": "pre-deadtime unprescaled rate",
"yVals": [0.002899099983985126,
0.0026693413965404034, 0.003336422611027956,
0.0021802419796586037, 0.0025957257021218 54,
0.0030407323502004147, 0.0027439768891781 57
```

Status

- Very old (first twiki revision in 2011, last updated 2018)
- Runs only on LXPLUS, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
- Fits are ROOT objects, serialised with pickle
- Output are static renders of the plots
- **Configuration is a python class**

  Configuration is now a YAML file with a defined **schema**. Easily portable
- No packaging, no CI/CD
- Must be restarted manually
- Python2
- Data needs to be plugged into an OMS panel (WBM imported the PNGs) and readable by its plot engine (HighCharts)

Status

- Fits are ROOT objects, serialised with pickle
- Output are static renders of the plots
- Configuration is a python class
- **No packaging**

  RPM Packaging streamlined. Easily **produce** a package installable in every CC7 machine with "make rpm". No more git clones: just rpm install the distributed packages.
- **No CI/CD (Continuos Integration/Continuos Deployment)**

  Now, on each new pushed commit on git:

  - A gitlab-ci job using docker images for the dependencies automatically builds an RPM package.
  - Tagged commits trigger different filenames (develop vs tagged version packages).
  - Packages are automatically uploaded on a public CERNBox folder (accessible also on /eos/).
- **Must be restarted manually**

  Now comes with a proper systemd service
- Python2
- Data needs to be plugged into an OMS panel (WBM imported the PNGs) and readable by its plot engine (HighCharts)

Next steps

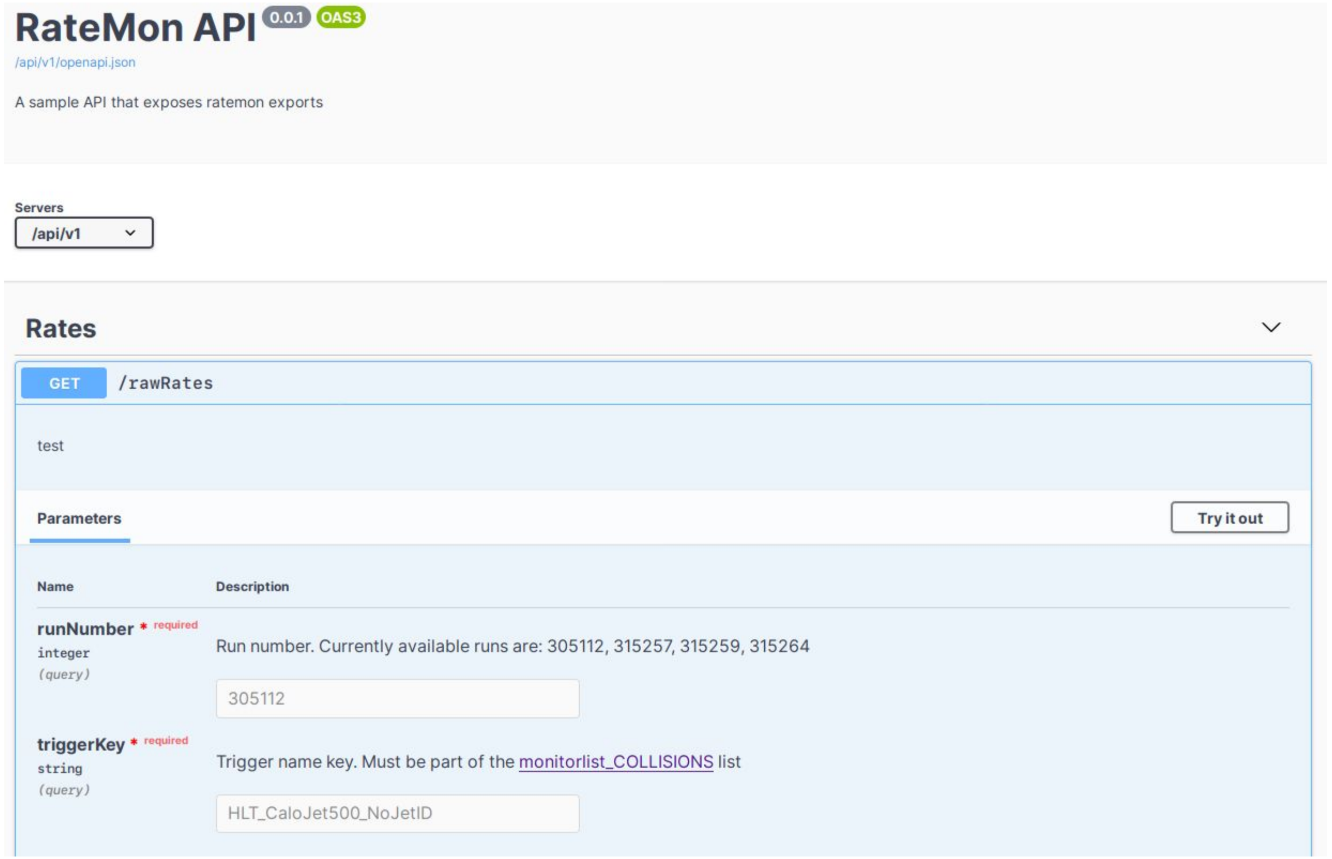
- Very old (first twiki revision in 2011, last updated 2018)
- Runs only on LXPLUS, needs `cx_Oracle` and access to `cms_omds_lb` oracle db
- Fits are ROOT objects, serialised with pickle
- Output are static renders of the plots
- Configuration is a python class
- No packaging, no CI/CD
- Must be restarted manually
- **Python2**
  Upgrade in progress but current machines at P5 still need Python2
- **Data needs to be plugged into an OMS panel** (WBM imported the PNGs) and readable by its plot engine (HighCharts)
  Push the produced JSONs to the DB, then prepare an OMS panel using these data to plot the trigger rates [...]

RateMon API

Simple service exposing the JSON exports (rates, fits).

With a simple **GET request** you get the rates of the desired Fill, Run or specific LS in the selected Run

```
GET
http://brandeis.cern.ch:8081/api/v1/rawRates?runNumber=305112&triggerKey=HLT_CaloJet500_NoJetID
```


RateMon API 0.0.1 OAS3

/api/v1/openapi.json

A sample API that exposes ratemon exports

Servers: /api/v1

Rates

GET /rawRates

test

Parameters

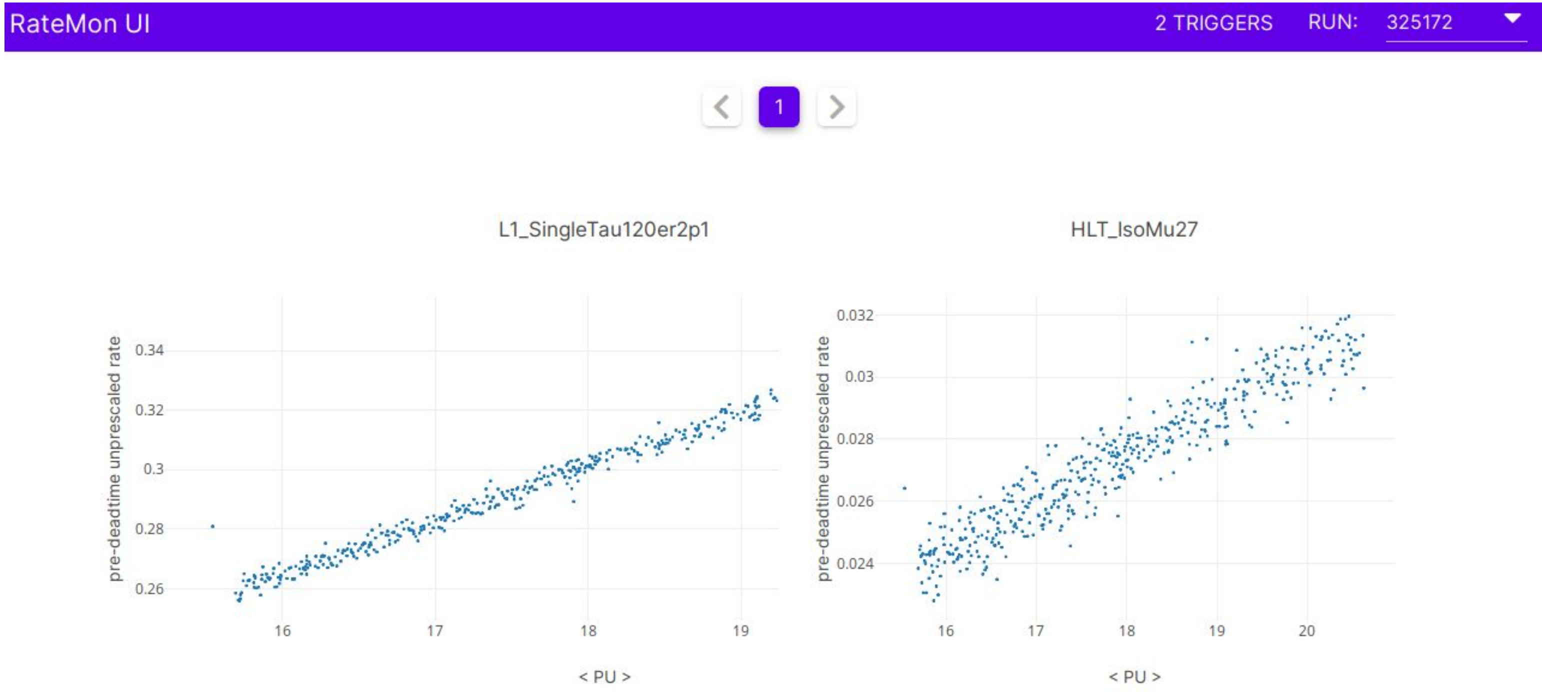
Name	Description
runNumber * required integer (query)	Run number. Currently available runs are: 305112, 315257, 315259, 315264 — 305112
triggerKey * required string (query)	Trigger name key. Must be part of the monitorlist_COLLISIONS list — HLT_CaloJet500_NoJetID

RateMon UI

Web app making use of the API.

Easy Trigger Selection

Interactive Plots

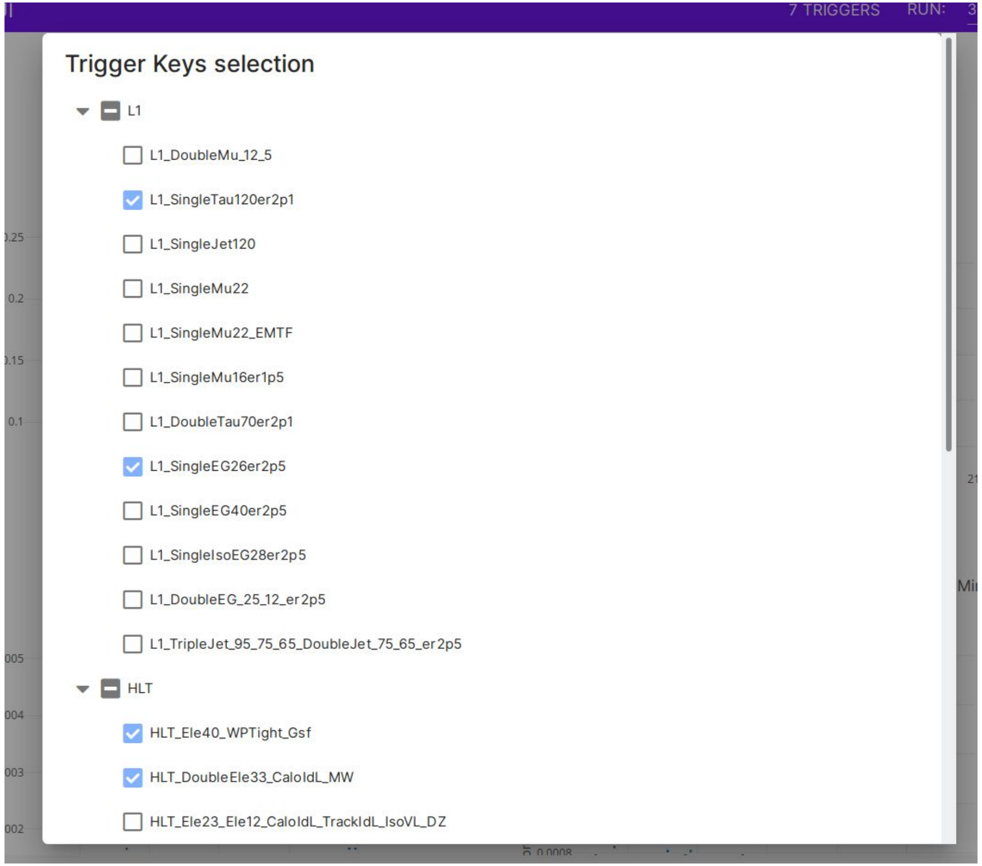


RateMon UI

Web app making use of the API.

Easy Trigger Selection

Interactive Plots



Trigger Keys selection

- [–] L1
  - [ ] L1_DoubleMu_12_5
  - [✓] L1_SingleTau120er2p1
  - [ ] L1_SingleJet120
  - [ ] L1_SingleMu22
  - [ ] L1_SingleMu22_EMTF
  - [ ] L1_SingleMu16er1p5
  - [ ] L1_DoubleTau70er2p1
  - [✓] L1_SingleEG26er2p5
  - [ ] L1_SingleEG40er2p5
  - [ ] L1_SingleIsoEG28er2p5
  - [ ] L1_DoubleEG_25_12_er2p5
  - [ ] L1_TripleJet_95_75_65_DoubleJet_75_65_er2p5
- [–] HLT
  - [✓] HLT_Ele40_WPTight_Gsf
  - [✓] HLT_DoubleEle33_CaloIdL_MW
  - [ ] HLT_Ele23_Ele12_CaloIdL_TrackIdL_IsoVL_DZ

RateMon UI

Current feature requests (thanks Sam Harper for the feedback!) :

- Export from UI
- Allow plotting Hz rather than Hz/bx
- Multiple fills in the same plot, Comparing runs feature
- Ability to plot fills, not only single runs
- y axis should show <Hz> on rates

Send yours at https://gitlab.cern.ch/avivace/ratemon-ui/issues !

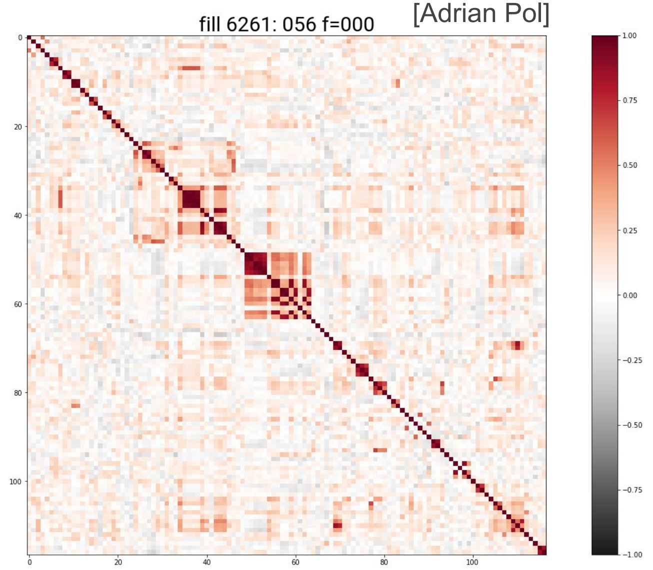
Anomaly Detection on Trigger Rates

- Fits and rates are dependent on PU and many other factors. Risk of bad fitting and a lot of false positive (RateMon alerts on >3σ)
- Dimensionality reduction needed: hundreds of rates and plots to look at!
- Exploit statistical relationships (possibly over time!) between trigger rates instead of focusing on absolute values
- Isolate and ignore (make the model "recognise" them) clusters of naturally related blocks of triggers (because tracking similar events)
- Find couples of triggers for which anomalies show different effects, thus breaking the normally shown correlation

E.g. Pearson Correlation Index:

$$-1 \leq \rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \leq +1$$

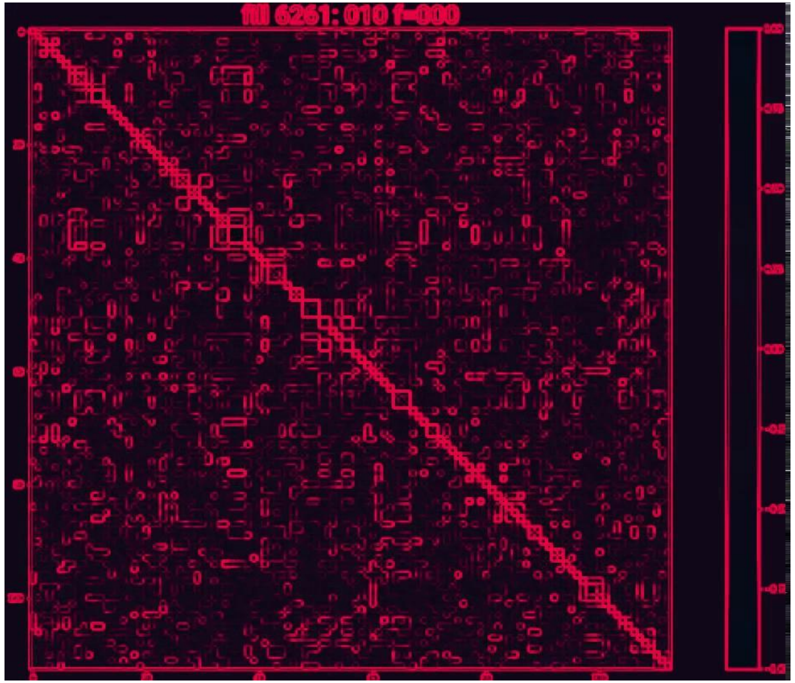
[Adrian Pol]

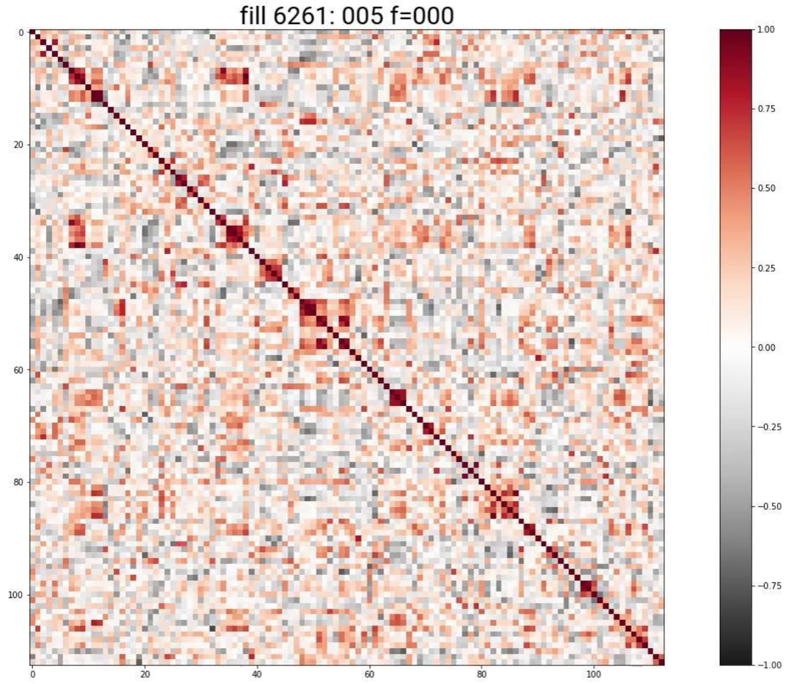


Anomaly Detection on Trigger Rates: next steps

- Possible model: **CVAE** (*...perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets*)
  [AutoEncoding Variational Bayes : arXiv:1312.6114]
- Good runs are easily obtainable
- **Problem**: labeled data
  Use the new Run Registry to find Runs/LS with reported problems on particular subdetectors (e.g. HCAL/ECAL)

Anomaly Detection on Trigger Rates: correlation matrixes





https://drive.google.com/drive/folders/1GxlMOirobY2r5PZie_PVXMw4dvp_uvz2?usp=sharing

[Adrian Pol]