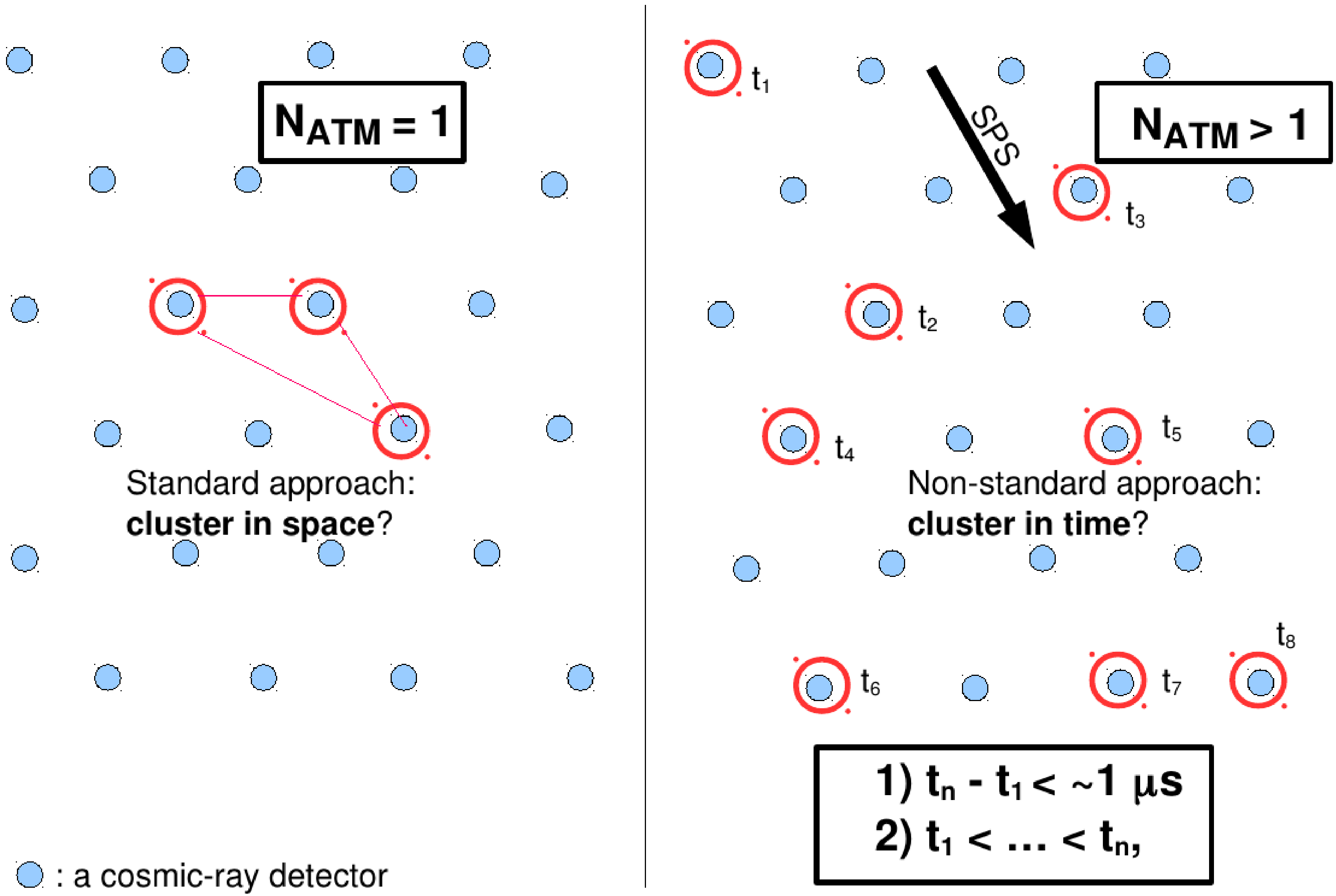
Time clustering analysis: a tool to search for unique signatures of cosmic-ray ensembles

Kévin Almeida Cheminant
Institute of Nuclear Physics PAS, Kraków

CREDO Anniversary Symposium
30^{th}-31^{st} August, 2017
INP PAS, Cracow, Poland

New trigger – chance for unique signature



1st step: collecting and sorting data

What is the data we are collecting?

- For **each** detector connected to the ***CREDO network***:
  - *Timestamps* of recorded events.
  - *ID* of the detector.

  → data is collected **every 24 hours**.

- Events are sorted by increasing timestamps

Station id

2206704	1409529600	40105068	1
2206664	1409529600	40105086	1
2206662	1409529600	46636506	1
2206704	1409529600	47116597	1
2206664	1409529600	47116626	1
509	1409529600	51999850	1
2206662	1409529600	52500763	1
2206703	1409529600	63566495	1
2206662	1409529600	63767873	1

Station status

Timestamp (sec) Timestamp (nano-sec)

2^{nd} *step: time clustering algorithm*

I. For a fixed time interval (*e.g* ***1 sec***):

1. Event **binning** (independent bins) according to the time interval value.

Bin #1 = 7 events

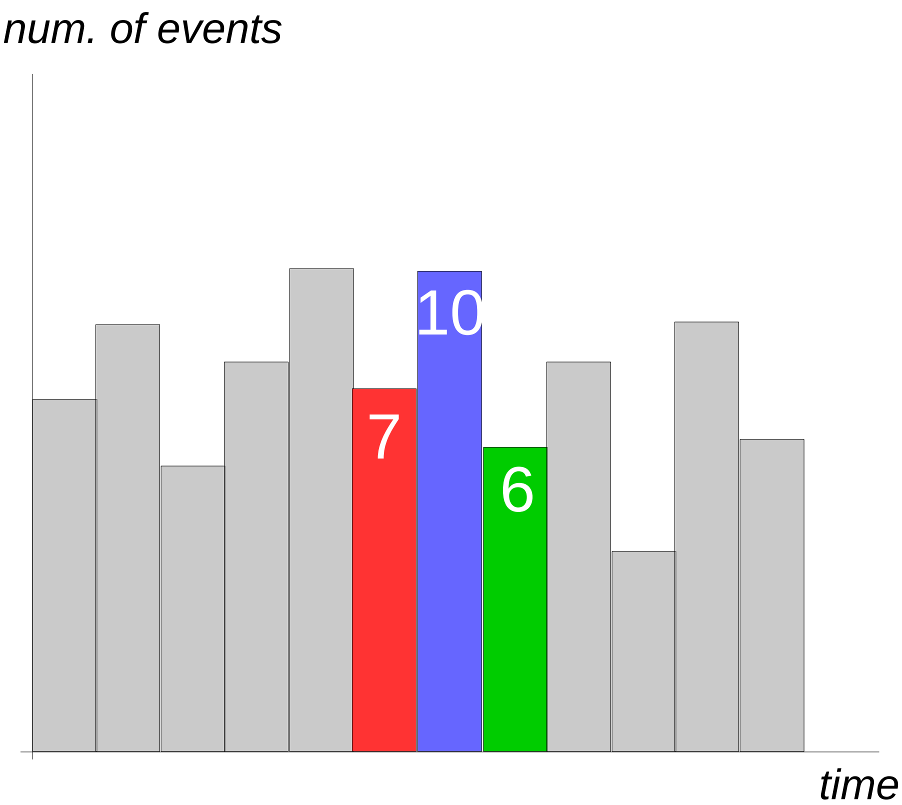
2001	1409529600	217049975	1
501	1409529600	219996563	1
2206202	1409529600	220070763	1
13004	1409529600	231053887	1
2206662	1409529600	265092268	1
9	1409529601	46801302	1
2206119	1409529601	47441737	1
2206664	1409529601	281220536	1
2206119	1409529601	300426257	1
1008	1409529601	307228336	1
7201	1409529601	313531023	1
2206202	1409529601	316860028	1
201	1409529601	334307610	1
2206664	1409529601	613546422	1
2206704	1409529601	614602412	1
2206662	1409529601	672626733	1
7401	1409529602	17613273	1
2206704	1409529602	61444256	1
2206664	1409529602	61444278	1
2206704	1409529602	65807888	1
2206423	1409529602	88937619	1
2206662	1409529602	99130276	1
2206703	1409529602	115914891	1

Bin #2 = 10 events

Bin #3 = 6 events

2^{nd} *step: time clustering algorithm*

I. For a fixed time interval (*e.g* ***1 sec***):



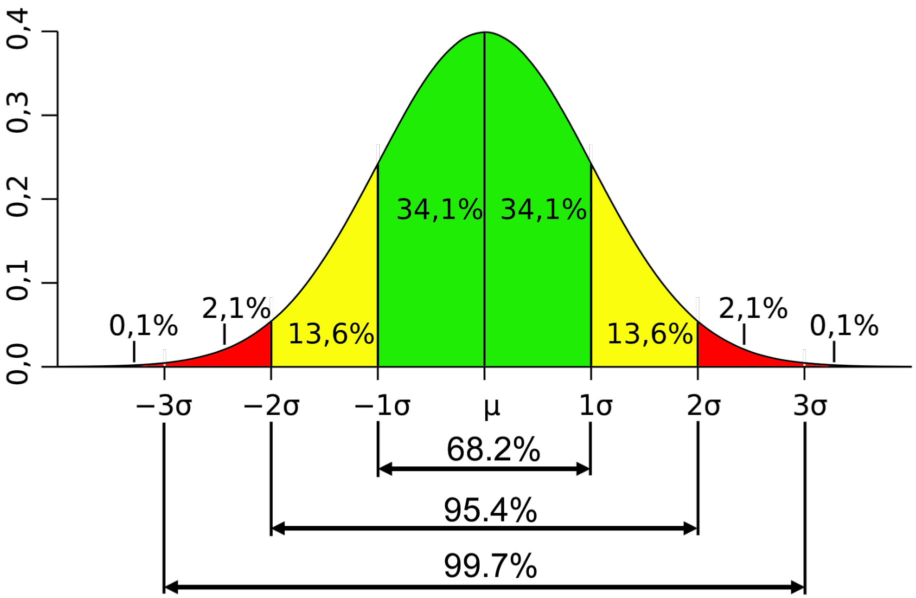
1. Event **binning** (independent bins) according to the time interval value.

2. For **each bin,** calculate local average **μ** (and standard deviation **σ**) taking a fixed number of bin (*e.g.* ***1***) before and after the bin considered (*e.g. blue):*

e.g: ***μ*** = ***7.6***

2^{nd} *step: time clustering algorithm*

I. For a fixed time interval (*e.g* ***1 sec***):



https://kanbanize.com/blog/normal-gaussian-distribution-over-cycle-time/

1. Event **binning** (independent bins) according to the time interval value.

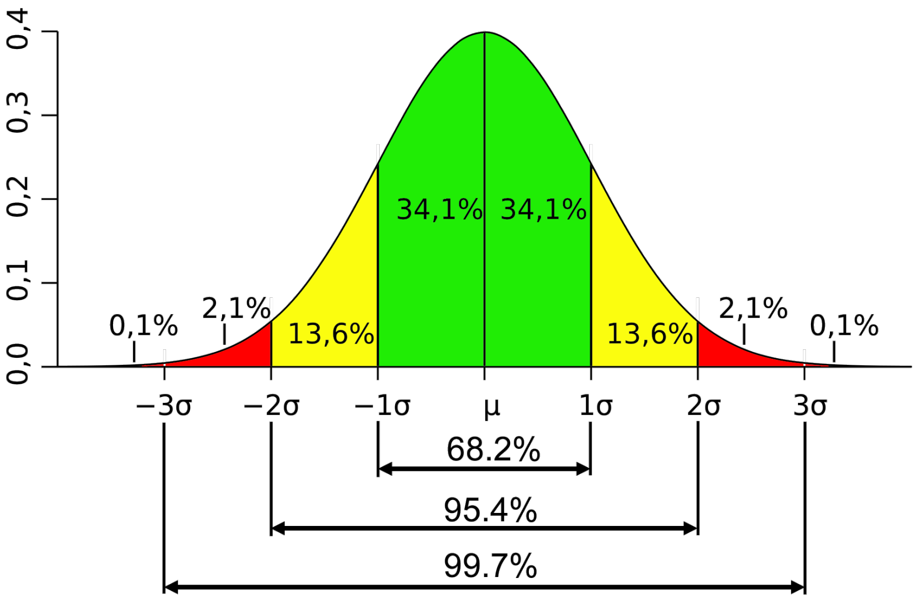
2. For **each bin,** calculate local average **μ** (and standard deviation **σ**) taking a fixed number of bin (*e.g.* ***1***) before and after the bin considered (*e.g. blue*):

   e.g: ***μ = 7.6***

3. Assuming a **normal distribution**, we get: *Z-score* $Z = \dfrac{X - \mu}{\sigma}$ and the p-value of the bin.

2^{nd} *step: time clustering algorithm*

I. For a fixed time interval (*e.g* ***1 sec***):



https://kanbanize.com/blog/normal-gaussian-distribution-over-cycle-time/

1. Event **binning** (independent bins) according to the time interval value.

2. For **each bin,** calculate local average **μ** (and standard deviation **σ**) taking a fixed number of bin (*e.g.* ***1***) before and after the bin considered (*e.g. blue*):

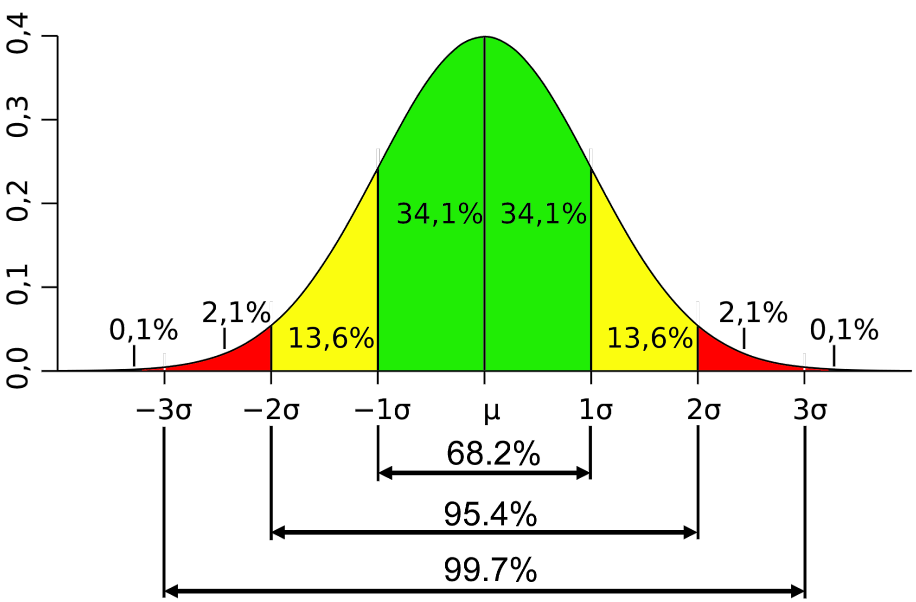
   e.g: ***μ* = *7.6***

3. Assuming a **normal distribution**, we get: *Z-score* $Z = \dfrac{X - \mu}{\sigma}$ and the p-value of the bin.

4. Sort bins by **decreasing** Z-scores.

2^{nd} step: time clustering algorithm

I. For a fixed time interval (*e.g* ***1 sec***):



https://kanbanize.com/blog/normal-gaussian-distribution-over-cycle-time/

1. Event **binning** (independent bins) according to the time interval value.

2. For **each bin,** calculate local average **μ** (and standard deviation **σ**) taking a fixed number of bin (*e.g.* ***1***) before and after the bin considered (*e.g. blue*):

   e.g: ***μ*** = ***7.6***

3. Assuming a **normal distribution**, we get: *Z-score* $Z = \frac{X - \mu}{\sigma}$ and the p-value of the bin.

4. Sort bins by **decreasing** Z-scores.

II. Try **other time windows** (litterature: possibilities up to **5 minutes** time-correlated events).

3^{rd} step: output files – log file

```
Performing time clustering analysis for day 2014-9-1
Type of analysis: independent bins - 100 bins on each side of analized cluster
Number of saved top clusters: 5
Extracting data...
```

Summary of ***fixed parameters***

```
****** Analysis for t = 1 sec *******
        Cluster #0 - Number of events: 139
        Local average = 100.24 - Standard Deviation = 10.789
        Sigma = 3.59255 - p-value = 0.000327464
-------------------------
        Cluster #1 - Number of events: 137
        Local average = 101.55 - Standard Deviation = 10.8674
        Sigma = 3.26206 - p-value = 0.00110605
-------------------------
        Cluster #2 - Number of events: 133
        Local average = 92.08 - Standard Deviation = 9.92256
        Sigma = 4.12394 - p-value = 3.72454e-05
-------------------------
        Cluster #3 - Number of events: 132
        Local average = 103.445 - Standard Deviation = 10.5129
        Sigma = 2.71618 - p-value = 0.00660393
-------------------------
        Cluster #4 - Number of events: 132
        Local average = 97.82 - Standard Deviation = 11.5686
        Sigma = 2.95456 - p-value = 0.00313113
-------------------------
Sum of sigmas = 16.6493
```

Results for a ***particular time window***

Sum of sigmas used to select the ***most interesting time window***

```
****** Analysis for t = 2 sec *******
        Cluster #0 - Number of events: 249
        Local average = 201.005 - Standard Deviation = 17.7206
        Sigma = 2.70843 - p-value = 0.00676027
-------------------------
        Cluster #1 - Number of events: 246
        Local average = 207.15 - Standard Deviation = 14.8308
        Sigma = 2.61955 - p-value = 0.0088047
-------------------------
        Cluster #2 - Number of events: 242
        Local average = 205.97 - Standard Deviation = 14.5955
        Sigma = 2.46858 - p-value = 0.0135652
-------------------------
        Cluster #3 - Number of events: 242
        Local average = 205.95 - Standard Deviation = 15.3034
        Sigma = 2.35568 - p-value = 0.0184887
```

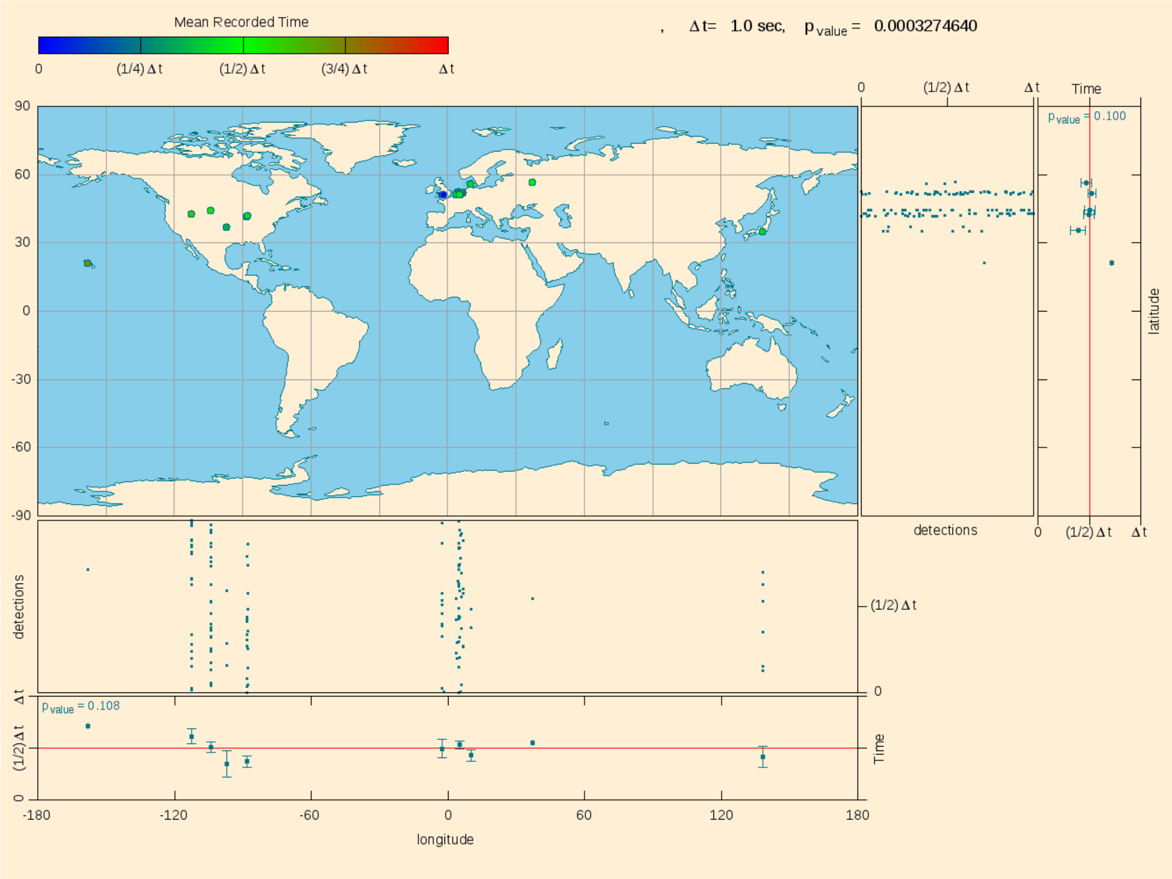
4th step: example of application

Cluster file → **Zooniverse – citizen science project***

```
1          0.000327464
1409529600        3.59
100.24  139
305        58804.075572595
2206119 58804.075866497
8003       58804.080310691
13201      58804.082681783
2206662 58804.089460474
13102      58804.099172147
2206662 58804.101807169
2206119 58804.109883122
2206664 58804.116747533
2206703 58804.117602896
2005       58804.117996883
2206704 58804.126996242
2206119 58804.152438426
2206423 58804.198436665
2206664 58804.202340671
2206703 58804.212477937
505        58804.216749126
```

Parameters

Events



*** *Dark Universe Welcome:*** looking for patterns in time arrivals of *possibly* correlated events. Involves **non-scientists.**

Conclusion and outlook

- Simple algorithm to detect excesses of events
  → only **timestamps** and **location** of detection needed

- Can be improved by taking into account the status of the stations and possible external impacts on event counts.

- **More stations = better statistics**

- Already operating for map classification (see *“CREDO monitor: the simplest tool for real data analysis”* talk by O. Sushchov*)*

CREDO

THE QUEST FOR UNEXPECTED