Detector Characterization
with Mutual Information Coefficients

J. J. Oh (KGWG-NIMS)

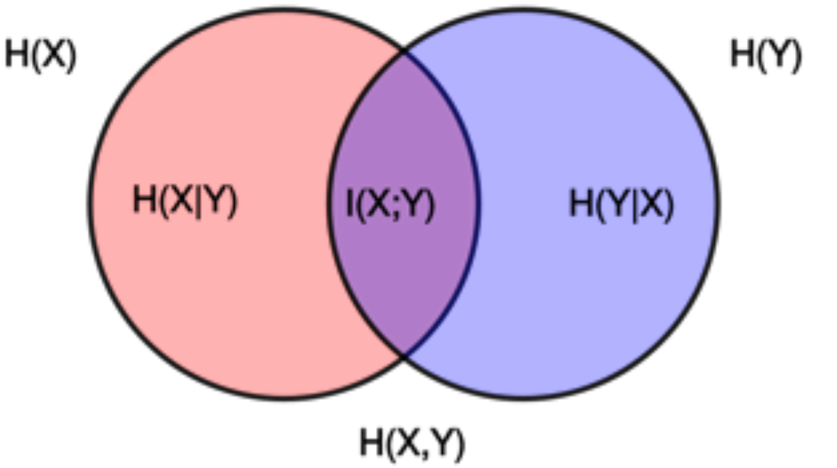
2014. 11. 4 (Tue)

KAGRA DetChar Teleconference

On behalf of
Sang Hoon Oh, Edwin J. Son, Young-Min Kim, Kyungmin Kim,
Lindy, Blackburn, Ruslan Vaulin, Florent Robinet, Kazuhiro Hayama

Objectives

- Mutual Information Coefficient (MIC) : nonlinear correlation measure widely used in *Information Theory* (Shannon-Weaver, 1949; Cover-Thomas, 1991)



measuring how much information shared between two random variables

- To get a correlation map using MIC between auxiliary channels of GW detectors

- To find and identify noise glitches in auxiliary channels of GW detectors

Methods

- **Pearson Correlation Coefficient (linear)**
  - a measure of linear correlation between two random variables defined by:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- **Mutual Information Coefficient: (non-linear)**
  - mutual information of two discrete random variables:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

where p(x,y) is the joint probability distribution function of X and Y, and p(x) and p(y) are the marginal probability distribution functions of X and Y.
- Intuitively, it measures the information that X and Y share: how much knowing one of these variables reduces uncertainty about the other.
- If both are independent variables, I(X;Y) = 0, no mutual information to share.

Codes

- In Scipy.stats module:

```
from scipy.stats import pearsonr
pearsonr(x,y)
```

which returns **`(pearsonr, 2-tailed p-value)`** between -1 and 1

- Interpretation:

r>= 0.70	Very strong positive relationship
0.40 ~ 0.69	Strong positive relationship
0.30 ~ 0.39	Moderate positive relationship
0.20 ~ 0.29	Weak positive relationship
0.01 ~ 0.19	No or negligible relationship
-0.01 ~ 0.19	No or negligible relationship
-0.20 ~ -0.29	Weak negative relationship
-0.30 ~ -0.39	Moderate negative relationship
-0.40 ~ -0.69	Strong negative relationship
-0.70 >=r	Very strong negative relationship

The p-value roughly indicates the probability of an uncorrelated system producing datasets that have a Pearson correlation at least as extreme as the one computed from these datasets. The p-values are not entirely reliable but are probably reasonable for datasets larger than 500 or so.

Codes and Data

- In Scikit.learn package:

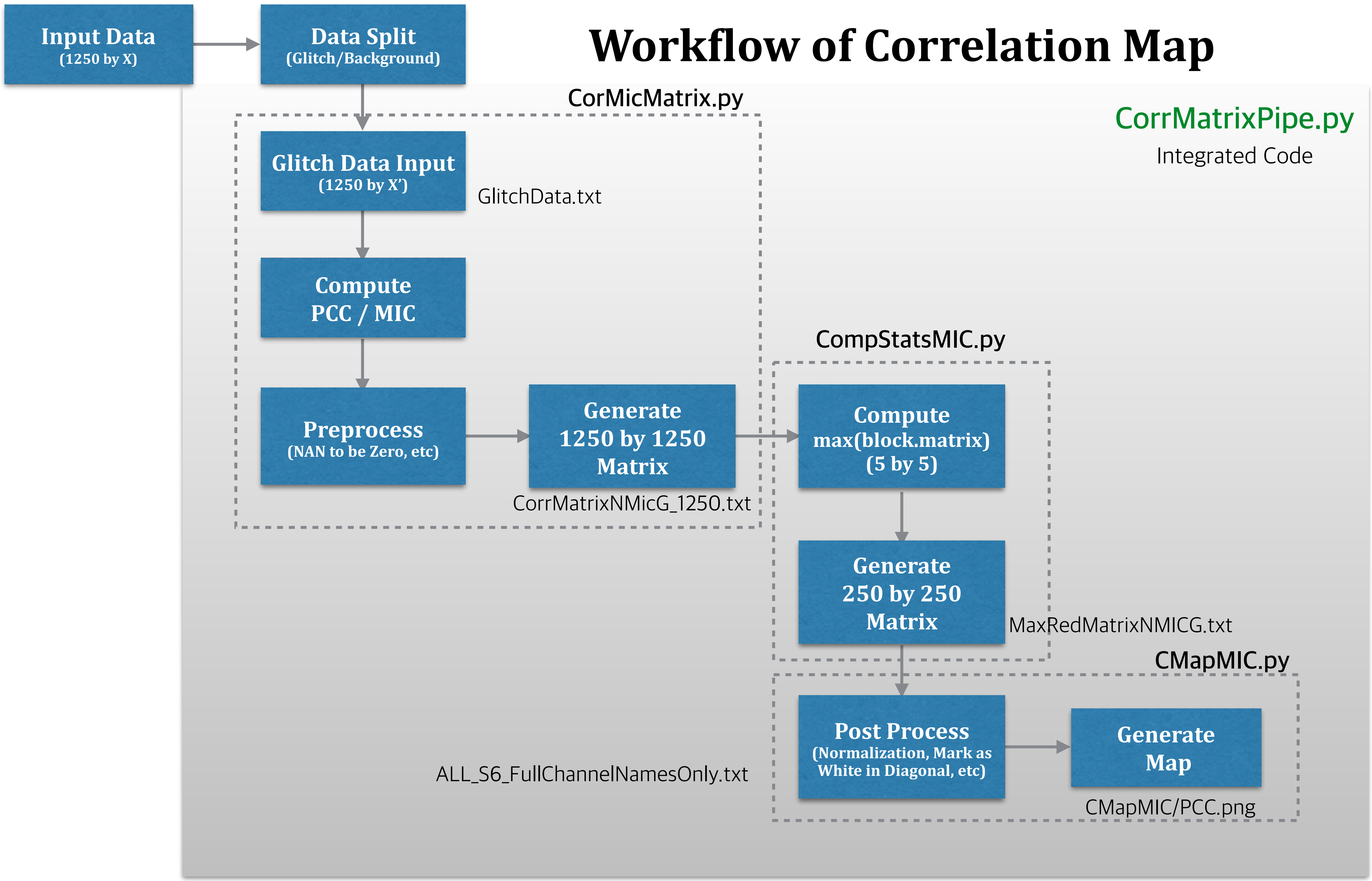
```
from sklearn.metrics.cluster import mutual_info_score
from sklearn.metrics.cluster import normalized_mutual_info_score

(normalized_)mutual_info_score(a,b)
```

- Code Requirements:
  - python 2.7 - https://www.python.org
  - matplotlib - http://matplotlib.org
  - scipy - http://www.scipy.org
  - numpy - http://www.numpy.org
  - scikit.learn - http://scikit-learn.org/
  - mpi4py (later) - http://mpi4py.scipy.org

- GitHub.com: https://github.com/chiewoo/AuxCode.git

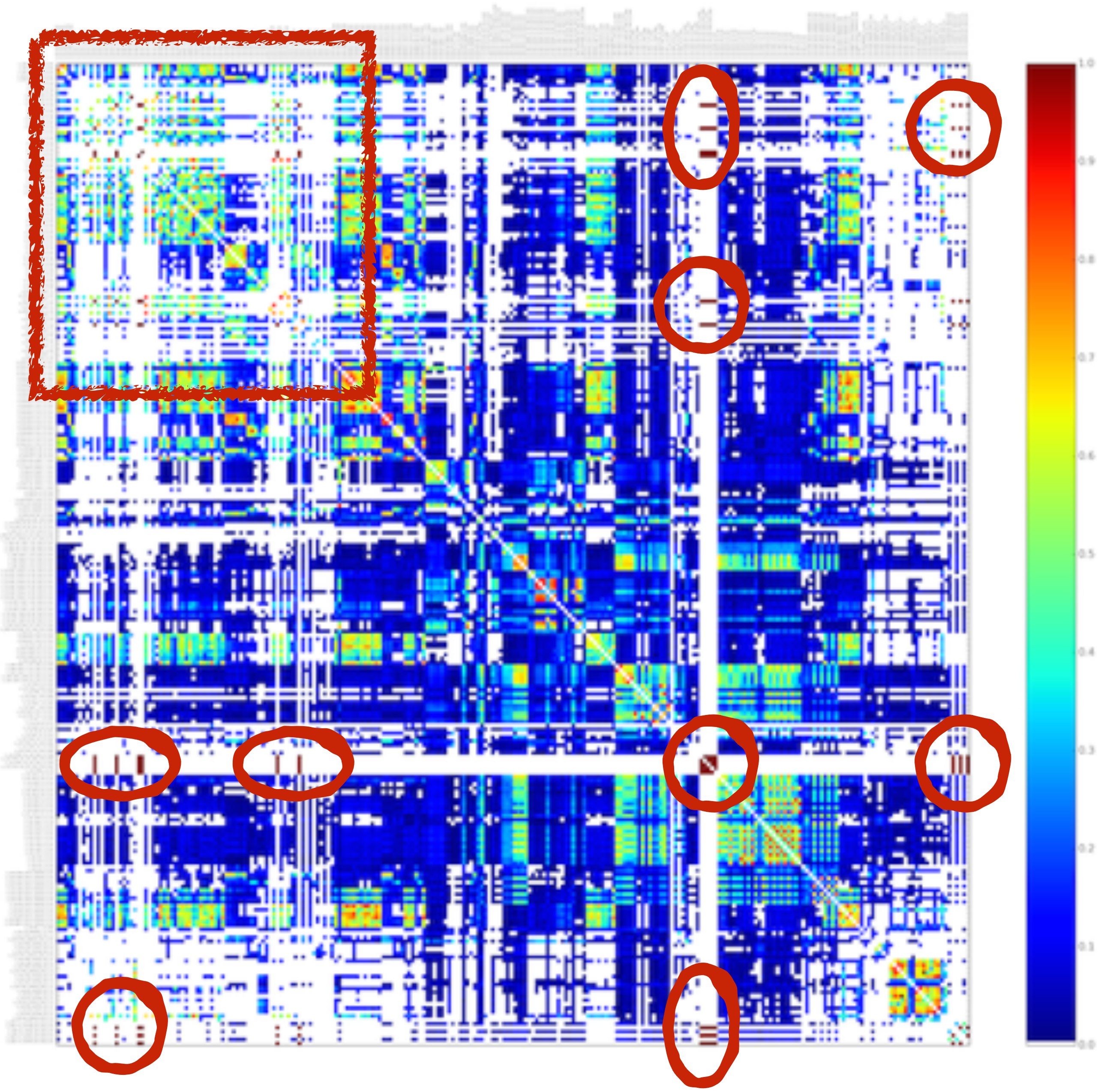
- Data
  - S6_week_959126400 Klein-Welle Triggered Data
    - Trigger threshold > 15
  - Data: ui04.sdfarm.kr: /data/ligo/home/john.oh/Pearsons/ALL_S6_full_100ms_Unorm_combined.ann
    - # of channel: 250
    - # of attribute: 5 {significance, deadtime, frequency, duration, number of points}
    - total: 1250 - {class 0(background) / class 1(glitch)}
  - Data Split:
    - We only use glitch-class data to find a channel-correlation that gives glitches
    - **Glitch data : (1250×2826)**
    - Background data: (1250×99869)

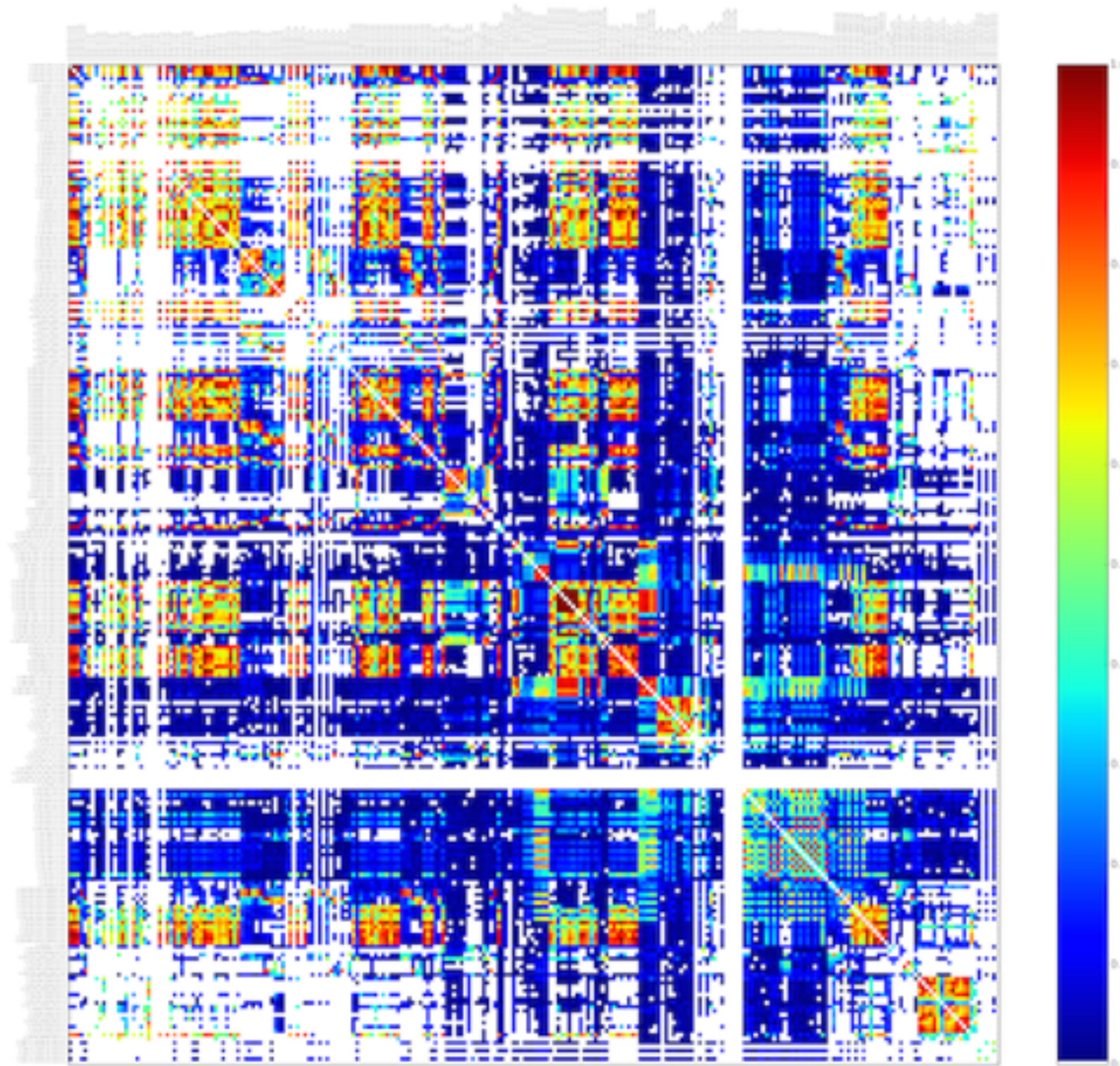
Workflow



Input Data
(1250 by X)
Data Split
(Glitch/Background)
Workflow of Correlation Map
CorMicMatrix.py
CorrMatrixPipe.py
Integrated Code
Glitch Data Input
(1250 by X')
GlitchData.txt
Compute
PCC / MIC
Preprocess
(NAN to be Zero, etc)
Generate
1250 by 1250
Matrix
CorrMatrixNMicG_1250.txt
CompStatsMIC.py
Compute
max(block.matrix)
(5 by 5)
Generate
250 by 250
Matrix
MaxRedMatrixNMICG.txt
CMapMIC.py
Post Process
(Normalization, Mark as
White in Diagonal, etc)
Generate
Map
ALL_S6_FullChannelNamesOnly.txt
CMapMIC/PCC.png

Analysis I



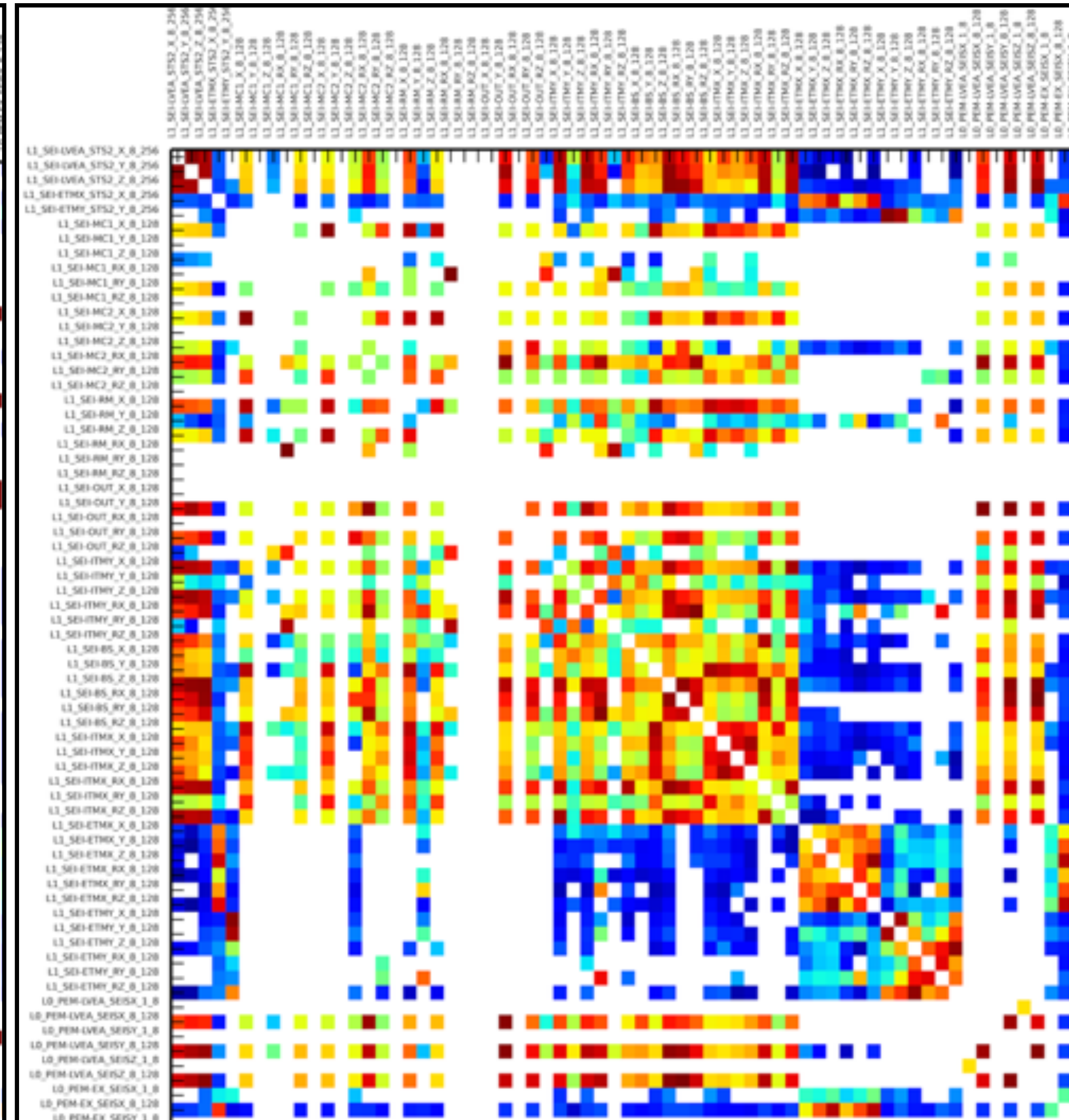


- Terminology: PCMap = Pearson's Correlation Map, MIMap = Mutual Information Map
- The correlation level in MIMap is lower than that in PCMap : Some ranges around 0.6~0.8 (orange, lightred) in PCMap haven been lowered to values 0.2~0.3 (cyan, blue). This is caused by the nonlinear strong-correlation points (dark red) that has not been appeared in PCMap.
- The circles and the box in MIMap are newly discovered correlations (presumedly, non-linear ones)
- There are lots of newly discovered spots in the whole map besides them.

Analysis II

Box - magnified

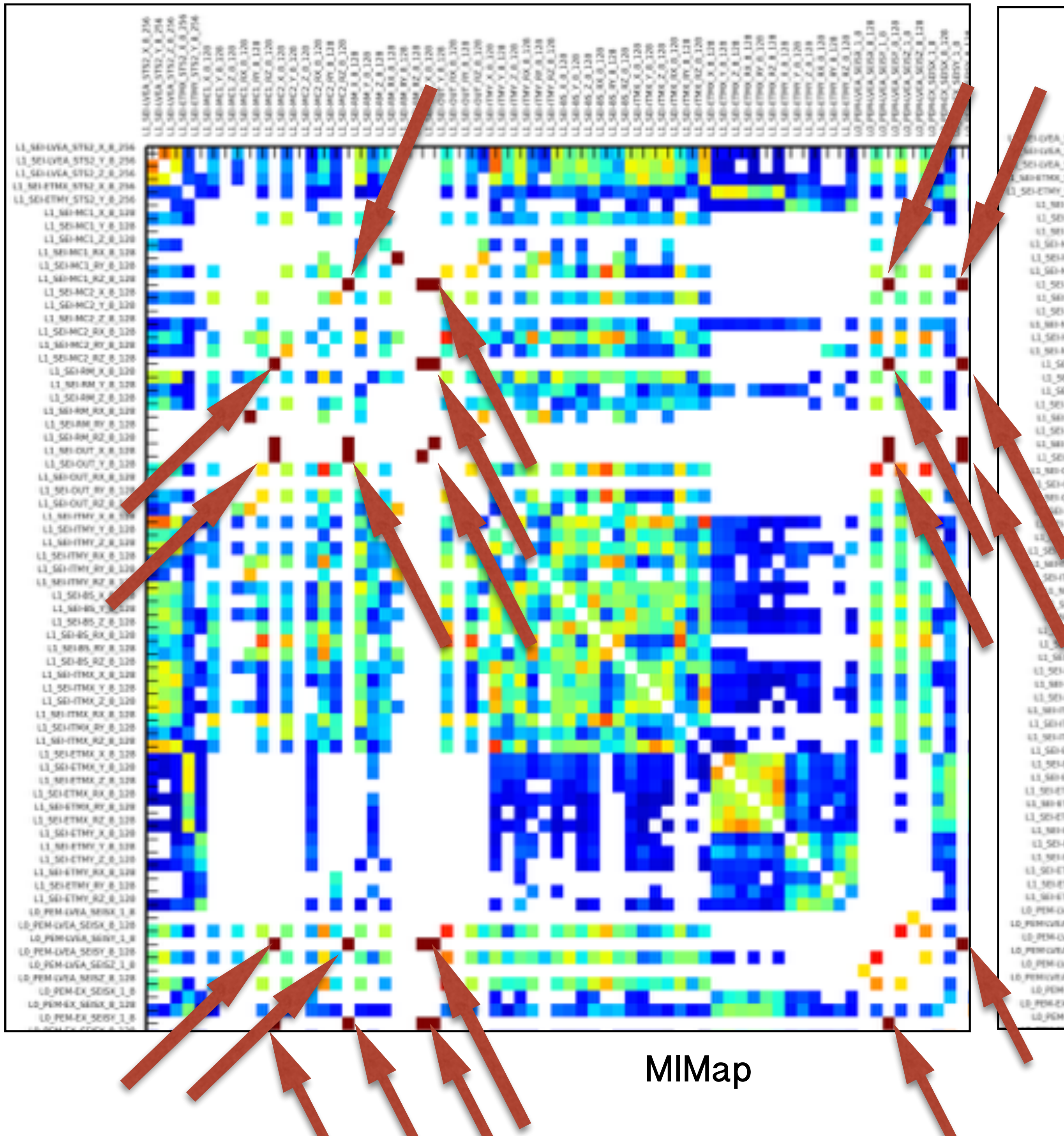
MIMap



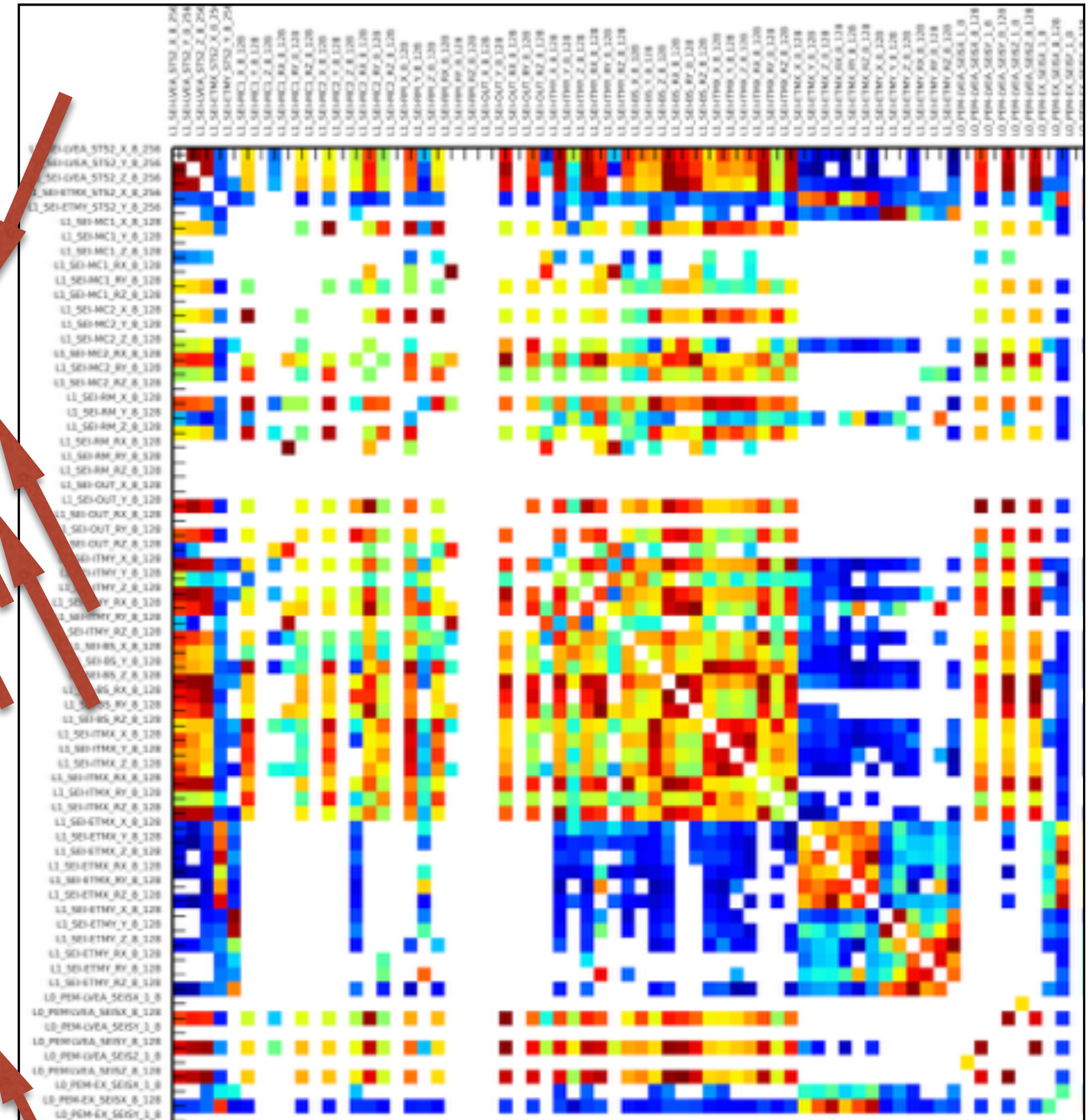
PCMap

Analysis III

newly detected strong correlations



MIMap



PCMap

- Many of these strong-correlated spots results from the “NaN” in computing PCC. This is because of 0/0 while computing PCC, which means “undefined value” with these data point. In other words, we cannot determine the correlation between two data points with PCC.
- The “NaN” is due to the sparsity of the original data – there are many zeroes in columns, which is originated from the Trigger threshold.
- Using MIC, this problem is resolved by returning very strong correlations.

Future Work

- Applying Correlation-threshold to select some interesting channels
- Confirm the data/attributes that are responsible for the correlation
- Generate Matrix Map for the lower trigger threshold (« 15) - helps finding channel correlation
- Get channel name information
- Study on other Trigger data[Omicron, etc]
- Compute correlation analysis between Auxchannels and GW channel
- Study up conversion data in the viewpoint of correlation coefficients

We can discuss on it more details during this Korea-Japan Workshop @ Toyama, December, 2014