NonNA: a non-stationary noise analysis tool

for noise hunters and commissioners

Francesco Di Renzo,
Physics Department of Pisa University

LIGO-Virgo Collaboration Meeting

18-21 March 2019

Lake Geneva, Wisconsin

VIRGO

IN SUPREMÆ DIGNITATIS 1343
UNIVERSITÀ DI PISA

INFN
Istituto Nazionale di Fisica Nucleare

NonNA tools: project overview

Original project by Gabriele Vajente (~2015):

- https://dcc.ligo.org/LIGO-G1500230

Updated project (2018-): Python scripts based on **virgotools**.

Data Analysis web area:

- https://scientists.virgo-gw.eu/DataAnalysis/NonNA

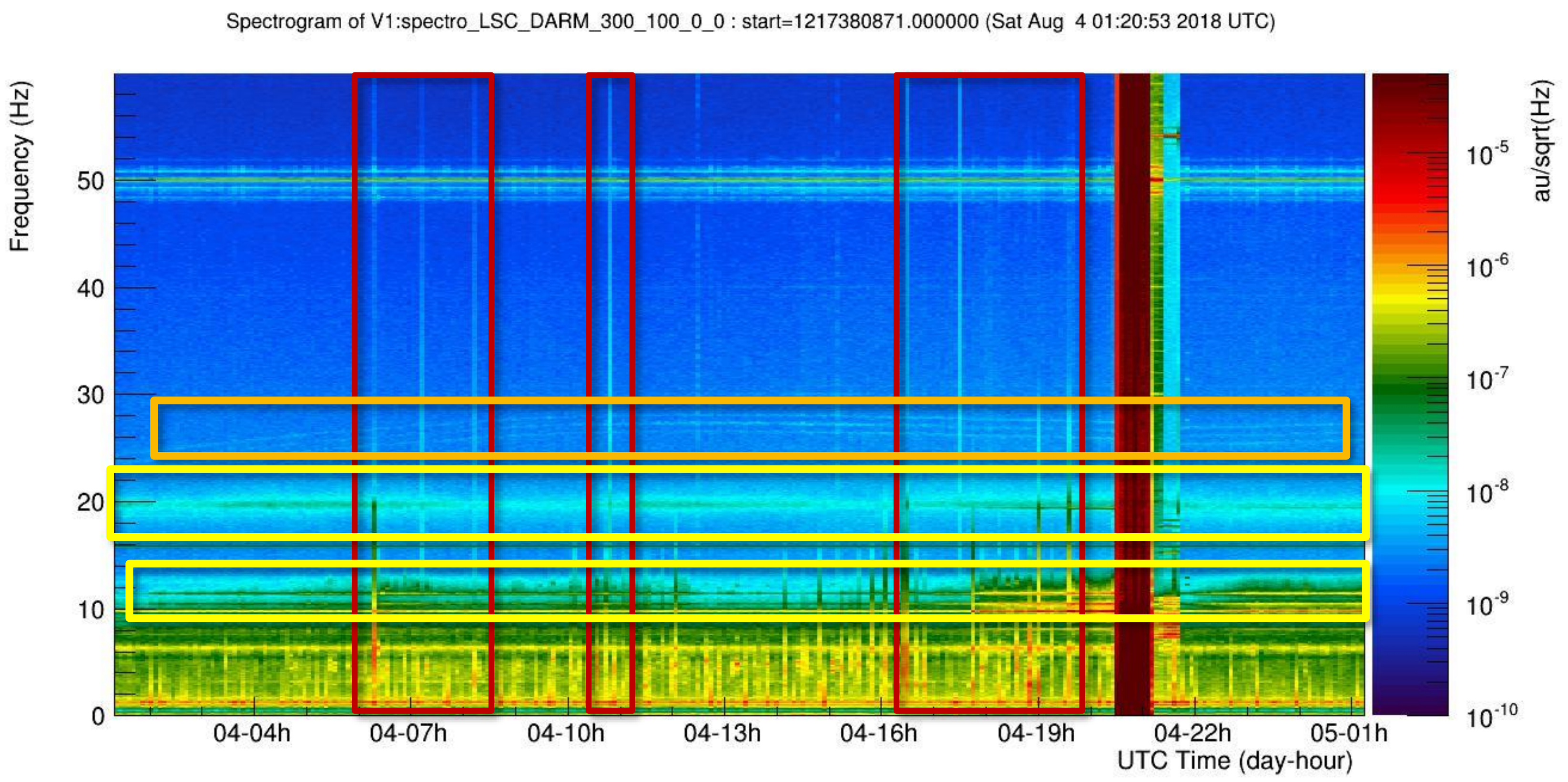
Previous presentations at Virgo Env and Detchar meetings:

- https://tds.virgo-gw.eu/?content=3&r=14414
- https://tds.ego-gw.it/?content=3&r=14614
- https://tds.virgo-gw.eu/?content=3&r=14806
- https://tds.virgo-gw.eu/?content=3&r=15319

Understanding the noise for more GW detections and better PEs

- Most of the detection and parameter estimation analysis pipelines rely on the assumption that the **detectors noises** are: [1]
  - Gaussian distributed,
  - **Stationary** and
  - Independent in each detector.
- Improper noise modelling may lead to incorrectly estimate **detection significance** and to systematic errors in the GW source **properties estimates**.
- Especially during commissioning phases, noises are likely to have non-Gaussian components and to be non-stationary.

Non-stationary noise in GW detectors: example from Virgo C10 data



Glitches: short duration “bursts” of excess power. Typical time scales $\lesssim$ 1 sec.

Slow non-stationarities (spectral noise):

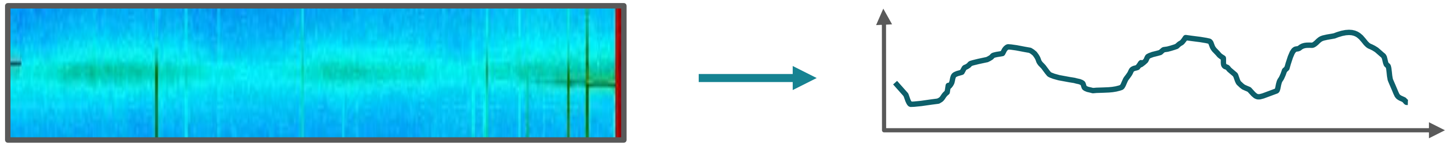
- Amplitude non-stationarities: bumps, “longer glitches” ($\gtrsim$ 1 sec),
- Frequency non stationarities: drifting/wandering lines

Data pre-processing for slow non-stationarities

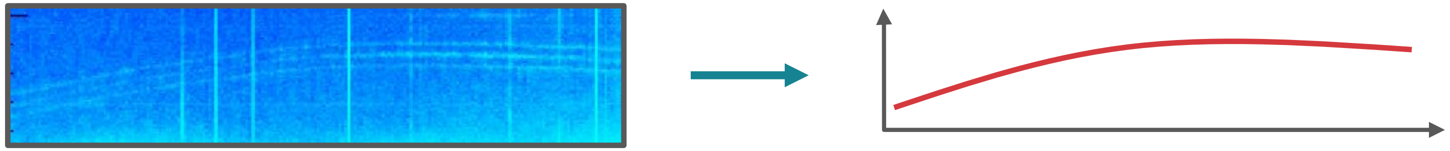
- **Band-limited Root Mean Square** (BRMS) of the power spectral density of the "noisy" channel: [2]

$$BRMS(t; [f_1, f_2]) = \sqrt{\int_{f_1}^{f_2} S_n(t, f)\, df}$$

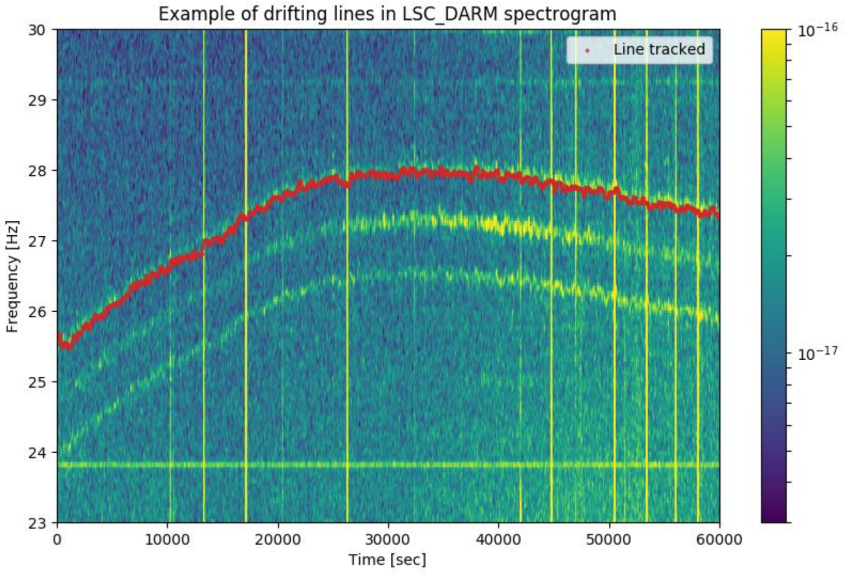
where $S_n(t, f)$, the noise power spectral density, can be estimated by means of some fft based method.

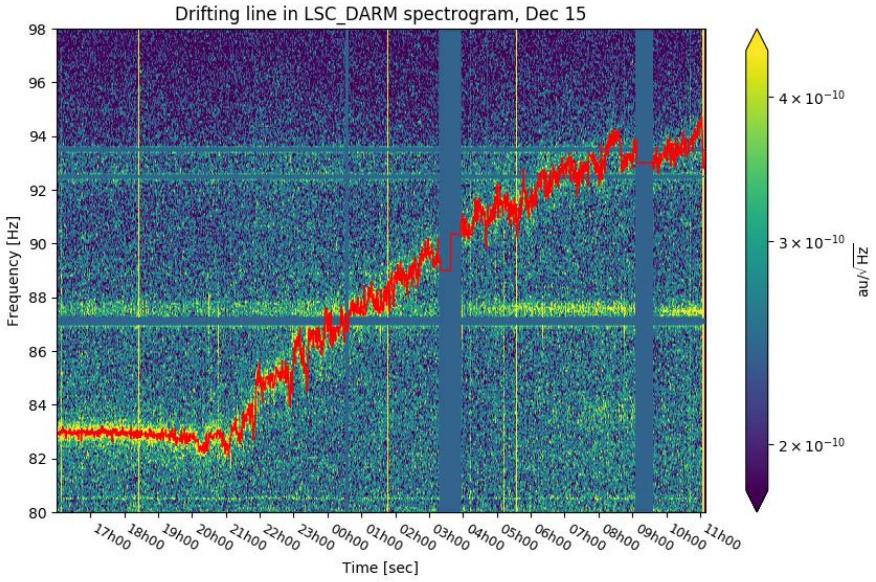


- **Line tracking**: extract from $S_n(t, f)$ the time series of the frequency maxima corresponding to the wandering line: continue to the next page.



NonNA Line Tracker tool





NonNA Line Tracker:

Inputs: high rate target channel (e.g. DARM, Hrec), duration (up to 5-7 days of data), frequency band where to look for the line.

Outputs: frequency maxima time series.

Notes: depending on the "noise foreground", it needs additional fine toning parameters: median normalization, masks.

Cross-correlation analysis

The detector and its environment are continuously monitored by $\mathcal{O}$(10k) **auxiliary sensors** (~40 MB/s flux of data): photodiodes, seismometers, magnetometers, etc.

The idea is that some of these channels may "**witness**" the noisy behaviour of the detector.

Pearson cross-correlation coefficient: measures the similarity, in the time domain, between two time series:

$$r_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}\frac{(x_i-\bar{x})(y_i-\bar{y})}{s_x s_y}$$

where $\bar{x} = \frac{1}{N}\sum_i x_i$ is the sample mean and $s_x{}^2 = \frac{1}{N-1}\sum_i (x_i-\bar{x})^2$ the sample variance.

NonNA cross-correlation tool

Overview: with a “brute force approach”, the tool takes as arguments a target channel and a list of auxiliary channels; it computes their Pearson correlation coeff. and produces a summary html page and log file with their ranking.

Target: DARM or Hrec BRMS, BNS range, frequencies of a wandering line, etc.

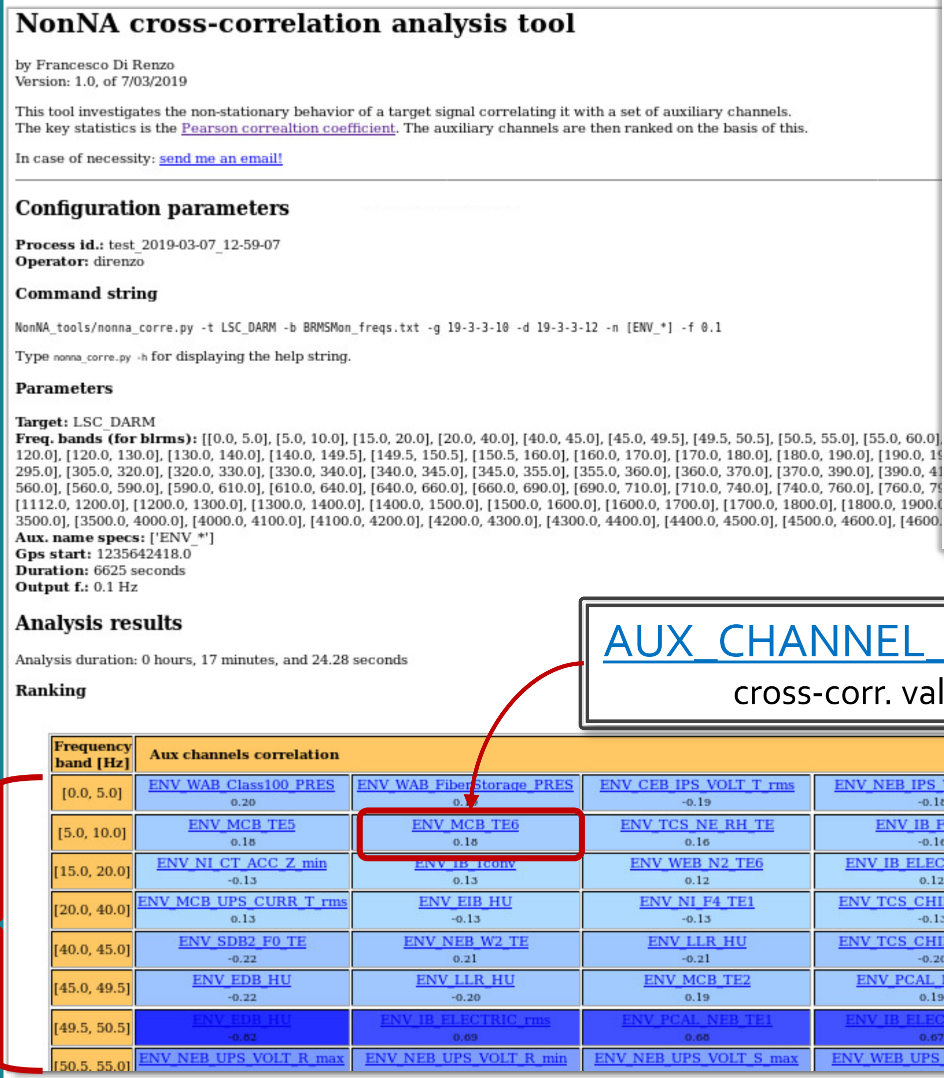
Aux. channels: standard Detchar channels, all channels from trend frame, ENV_*, LSC_*_rms, etc.

Typical set up: $\mathcal{O}$(10k) seconds of data, $\mathcal{O}$ (10k) auxiliary channels, 0.1 Hz output frequency.

Execution time (extreme case): 40 minutes analysis for 40k channels for 1 day, and 15 min for the plots.

Command string:
```
nonna_corre.py -t LSC_DARM
               -b BRMSMon_freqs.txt
               -g 19-3-3-10  -d 19-3-3-12
               -n [ENV_*]
               -f 0.1
```

NonNA results html output page



NonNA cross-correlation analysis tool

by Francesco Di Renzo
Version: 1.0, of 7/03/2019

This tool investigates the non-stationary behavior of a target signal correlating it with a set of auxiliary channels.
The key statistics is the Pearson correaltion coefficient. The auxiliary channels are then ranked on the basis of this.

In case of necessity: send me an email!

Configuration parameters

Process id.: test_2019-03-07_12-59-07
Operator: direnzo

Command string

```
NonNA_tools/nonna_corre.py -t LSC_DARM -b BRMSMon_freqs.txt -g 19-3-3-10 -d 19-3-3-12 -n [ENV_*] -f 0.1
```

Type `nonna_corre.py -h` for displaying the help string.

Parameters

Target: LSC_DARM
Freq. bands (for blrms): [[0.0, 5.0], [5.0, 10.0], [15.0, 20.0], [20.0, 40.0], [40.0, 45.0], [45.0, 49.5], [49.5, 50.5], [50.5, 55.0], [55.0, 60.0],
120.0], [120.0, 130.0], [130.0, 140.0], [140.0, 149.5], [149.5, 150.5], [150.5, 160.0], [160.0, 170.0], [170.0, 180.0], [180.0, 190.0], [190.0, 1
295.0], [305.0, 320.0], [320.0, 330.0], [330.0, 340.0], [340.0, 345.0], [345.0, 355.0], [355.0, 360.0], [360.0, 370.0], [370.0, 390.0], [390.0, 4
560.0], [560.0, 590.0], [590.0, 610.0], [610.0, 640.0], [640.0, 660.0], [660.0, 690.0], [690.0, 710.0], [710.0, 740.0], [740.0, 760.0], [760.0, 7
[1112.0, 1200.0], [1200.0, 1300.0], [1300.0, 1400.0], [1400.0, 1500.0], [1500.0, 1600.0], [1600.0, 1700.0], [1700.0, 1800.0], [1800.0, 1900.
3500.0], [3500.0, 4000.0], [4000.0, 4100.0], [4100.0, 4200.0], [4200.0, 4300.0], [4300.0, 4400.0], [4400.0, 4500.0], [4500.0, 4600.0], [4600.
Aux. name specs: ['ENV_*']
Gps start: 1235642418.0
Duration: 6625 seconds
Output f.: 0.1 Hz

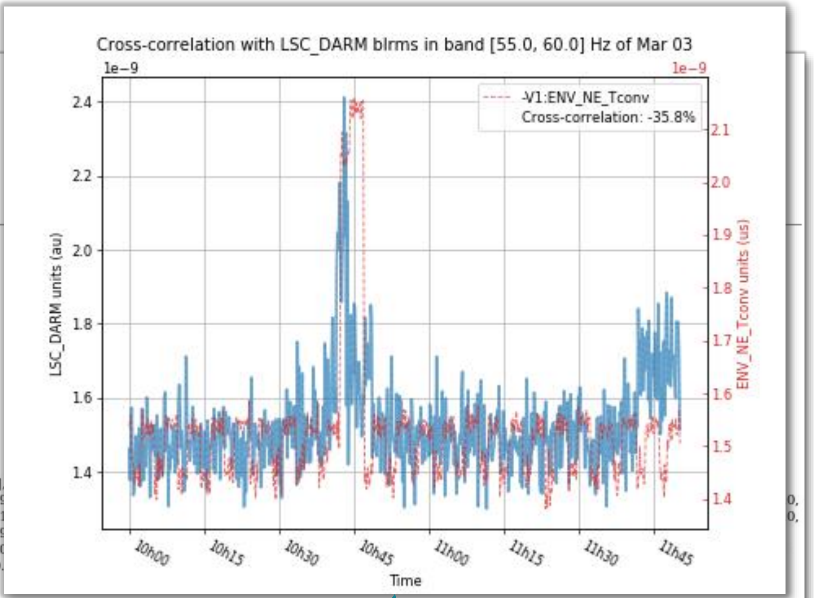
Analysis results

Analysis duration: 0 hours, 17 minutes, and 24.28 seconds

Ranking

Channel ranking

Frequency band [Hz]	Aux channels correlation					
[0.0, 5.0]	ENV_WAB_Class100_PRES 0.20	ENV_WAB_FiberStorage_PRES 0.1[illegible]	ENV_CEB_IPS_VOLT_T_rms -0.19	ENV_NEB_IPS_VOLT_T_rms -0.18	ENV_WAB_FiberLab_PRES 0.18	ENV_MCB_PRES 0.18	ENV_NEB_IPS_VOLT_R_rm
[5.0, 10.0]	ENV_MCB_TE5 0.18	ENV_MCB_TE6 0.18	ENV_TCS_NE_RH_TE 0.16	ENV_IB_F0_TE2 -0.16	ENV_NE_MIR_COIL_UL_TE 0.16	ENV_TCS_NI_CO2laser_TE 0.15	ENV_DT_F4_TE2 -0.15
[15.0, 20.0]	ENV_NI_CT_ACC_Z_min -0.13	ENV_IB_Tconv 0.13	ENV_WEB_N2_TE6 0.12	ENV_IB_ELECTRIC_max 0.12	ENV_IB_ELECTRIC_mean 0.12	ENV_IB_ELECTRIC_min 0.12	ENV_BS_F0_TE1 0.11
[20.0, 40.0]	ENV_MCB_UPS_CURR_T_rms 0.13	ENV_EIB_HU -0.13	ENV_NI_F4_TE1 -0.13	ENV_TCS_CHILROOM_TE -0.13	ENV_CEB_SEIS_W_rms 0.12	ENV_CEB_SEIS_W_max 0.12	ENV_LLR_HU -0.12
[40.0, 45.0]	ENV_SDB2_F0_TE -0.22	ENV_NEB_W2_TE 0.21	ENV_LLR_HU -0.21	ENV_TCS_CHILROOM_TE -0.20	ENV_EDB_HU -0.19	ENV_NEB_W1_TE7 0.16	ENV_NEB_W1_TE8 0.16
[45.0, 49.5]	ENV_EDB_HU -0.22	ENV_LLR_HU -0.20	ENV_MCB_TE2 0.19	ENV_PCAL_NEB_TE1 0.19	ENV_MCB_TE1 0.19	ENV_WAB_FiberStorage_HU -0.19	ENV_WAB_Class100_HU -0.18
[49.5, 50.5]	ENV_EDB_HU -0.82	ENV_IB_ELECTRIC_rms 0.69	ENV_PCAL_NEB_TE1 0.68	ENV_IB_ELECTRIC_max 0.67	ENV_IB_ELECTRIC_mean 0.67	ENV_IB_ELECTRIC_min 0.66	ENV_WAB_Class100_HU -0.66
[50.5, 55.0]	ENV_NEB_UPS_VOLT_R_max	ENV_NEB_UPS_VOLT_R_min	ENV_NEB_UPS_VOLT_S_max	ENV_WEB_UPS_VOLT_T_min	ENV_WEB_UPS_VOLT_R_min	ENV_WEB_UPS_VOLT_T_max	ENV_WEB_UPS_VOLT_R_ma



AUX_CHANNEL_NAME
cross-corr. value

BLRMS frequency bands

Cross-correlation extended: regression analysis

Motivations/ideas:

- Many channels can contribute to target non-stationarities at the same time;
- Usually, the channels are interdependent: redundant information, feedback mechanisms, cascade effects;
- Do the channels themselves respond to underlying noise processes?

Regression analysis: model the target (y) as a linear combination of the aux. channels (x_n):

$$\hat{y}_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_n x_{ni} \equiv \mathrm{X}_i \boldsymbol{\beta}$$

$e_i = y_i - \hat{y}_i$ is the **residual** difference between the estimate $\hat{y}_i$ and the target y_i .

Make a model → Evaluate the coefficients ($\boldsymbol{\beta}$) according to some criterion → Assess the goodness of the model according to some criterion: R^2, R^2_{adj}, AIC, BIC → Inference on the results: identify the likely noise source(s)

Ordinary Least Square solution

Under the Classical Linear Model (CLM) assumptions

- $\mathrm{X}_i \equiv (x_{1i}, x_{2i}, \dots, x_{ni})$ is full rank (independent aux channels)
- $E[e_i] = 0, \quad E\left[e_i^2\right] = \sigma^2 \text{ and } E[e_i e_j] = 0$

the **Gauss-Markov theorem** says that the **Ordinary Least Squares** (**OLS**) estimator $\widehat{\boldsymbol{\beta}}$ of the regression coefficients is **BLUE**: [3]

- **B**est (minimum variance, according to the Cramèr-Rao lower bound [4])
- **L**inear function of y
- **U**nbiased ($E\left[\widehat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}$)
- **E**stimator of $\boldsymbol{\beta}$

If the e_i's are also normally distributed, $\widehat{\boldsymbol{\beta}}$ becomes **efficient**, and reliable $\boldsymbol{t}$ **and $\boldsymbol{F}$ tests** can be carried out to asses channels and models significances.

However:

- Often CLM assumptions don't hold: correlated auxes, homoscedasticity, etc.;
- It could be preferable to have a smaller variance in change of a biased estimate.

Principal component regression

Intermediate step: perform a Principal Component Analysis (PCA) of the auxiliary channels, then regress the target onto these PCs:

$$X^T X = V \Lambda V^T$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, and λ_i is the variance of the i-th principal component.

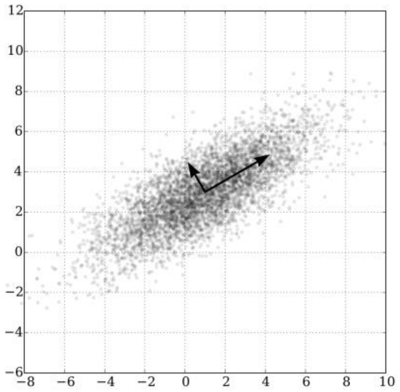


Image credit Wikipedia

Pros:

- Since the zero-energy PCs are automatically omitted, OLS optimal solution is recovered: **multicollinearity problem fixed**;
- **Dimensionality reduction**: keeping only a number (p) of the PCs introduces a bias (hard shrinkage) but reduces the variance of the estimate:

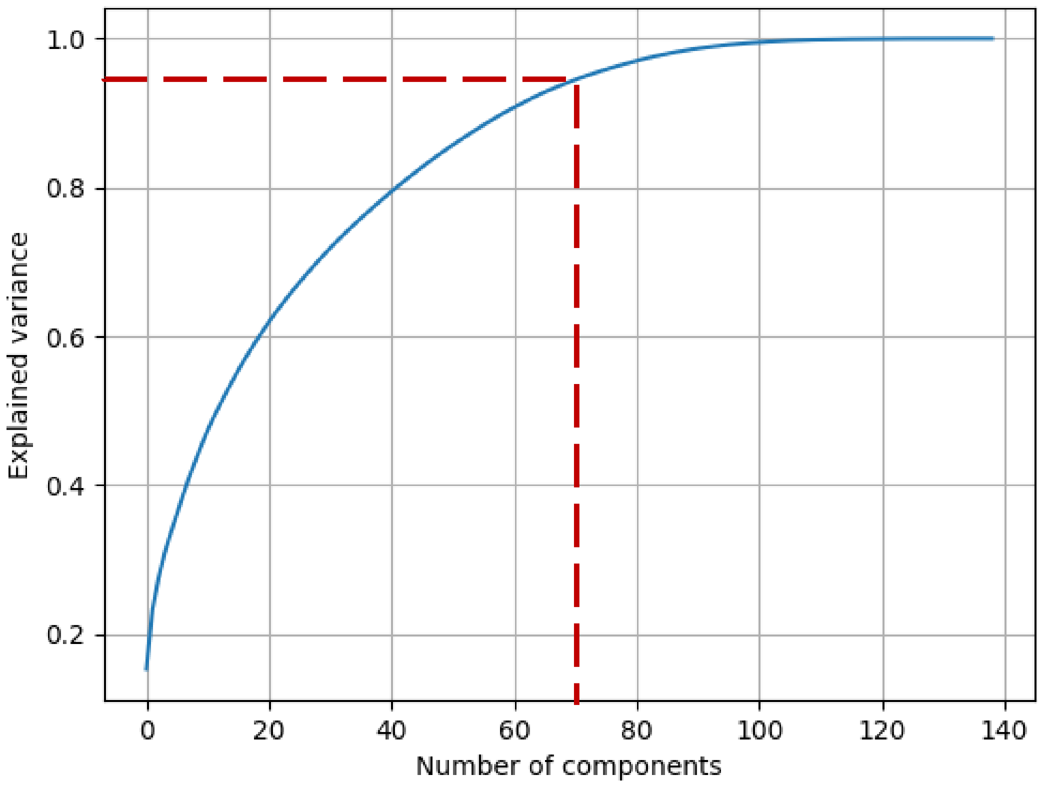
Reconstruction error (bias): $\left\| x_i - \hat{x}_i^{(p)} \right\|^2 = \sum_p^n \lambda_i$

Variance reduction: $\mathrm{Cov}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) - \mathrm{Cov}(\widehat{\boldsymbol{\beta}}_{\mathrm{PCA}}^{(p)}) \sim \sigma^2 \sum_p^n \frac{1}{\lambda_i}$

- Step towards understanding the underlying **data generating processes** (DGP).

Correlated auxiliary channels and explained variance

Normalized cumulative sum of principal values



Example: all ENV_*_rms channels (137) on 3 hours of data at 0.1 Hz output frequency.

Keeping just half of the principal components allows to explain ~95% of the data variance.

Cons: what about the interpretability of these PCs?

Cons:

- PCs are "geometrical objects" not corresponding to any physical sensor or place in the detector. How can we interpret them?

  Possible solution:
  Exploiting Virgo channel names convention [5],

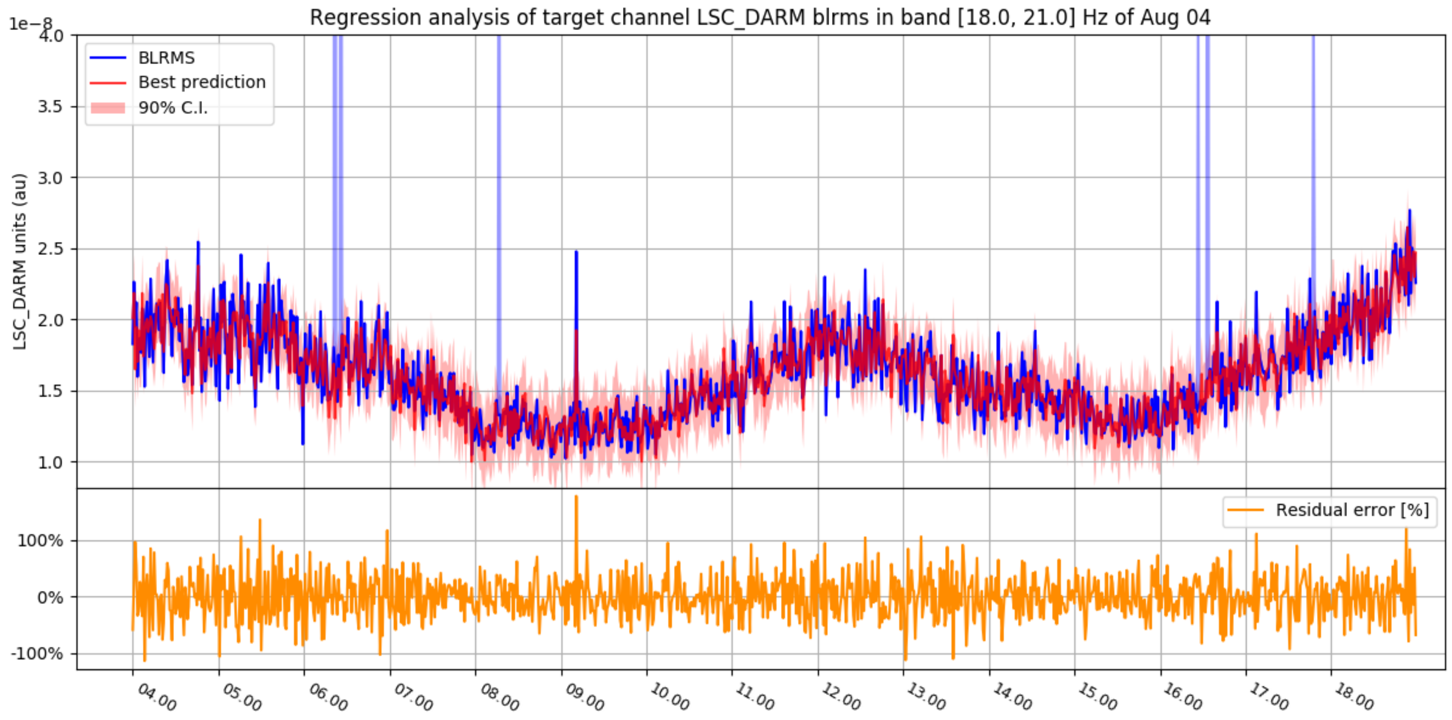
  V1:SUBSYSTEM_LOCATION_SENSOR_...

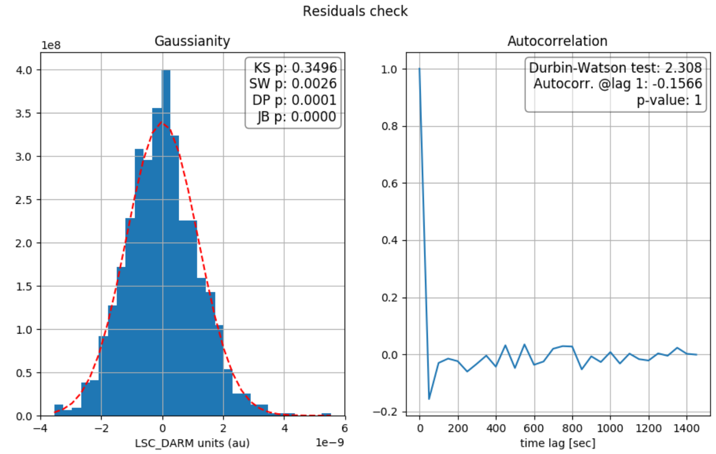
  we can produce, for every PC and its contribution to the regression, the histograms corresponding to which SUB, LOC and SENS are most contributing to it.

Some finer points:

- Aux channels principal values are "a priori" not related with the target. So, why removing smaller ones? [6]
  Possible solutions: *supervised PCR* [7], PLS regression.
- How to choose p?
  Possible solutions: fixing the explained variance (e.g. 95%) or by iteration, according to some criterion (R^2_{adj}, AIC, BIC), if n is not too big ($\lesssim 400$).

NonNA regression analysis example





Results:

54k seconds of data, 216 model params.

$R^2_{\text{adj}} \simeq 72\%$.

Many channels related with the pre-stabilized laser and the injection subsystem (based on their t-statistics).

Refer to the spectrogram on page 4

Conclusions/ discussion

Two tools for (slow) non-stationary noise investigation have been presented:

- Based on **time domain** cross-correlation analysis: Pearson correlation coefficient, and regression analysis;
- Fast results exploiting **multiprocessing** on Virgo farm computers;
- Correlation tool suitable for **1 vs. 1 comparison** in a "brute force" approach (but beware of correlation by chance);
- Regression tool meant for **explanatory purposes** but **suitable for prediction**: both interpolation and extrapolation. High dependency of the kind of non-stationarity, though;
- **PCR** introduced to fix multi-collinearity problem and to reduce the variance, can be used to dig deeper into the origin of the noise;
- Make the information from PCs more easily accessible by noise hunters and commissioners.

Bibliography

1. N. Christensen *et al. A guide to LIGO-Virgo detector noise and extraction of gravitational-wave signals* (draft) https://dcc.ligo.org/LIGO-P1900004, 2019.
2. G. Vajente. *Nonstatmoni technical description*. Virgo Internal note, VIR-0004A-08, https://tds.virgo-gw.eu/ql/?c=1958, 2008.
3. Robinson, G.K.*That BLUP is a Good Thing: The Estimation of Random Effects.* Statistical Science. **6** (1): 15–32, . (1991).
4. Cramèr, H. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1946.
5. B. Swinkels et al . *Channel naming conventions for adv signals*. Technical report, Virgo internal note. VIR-0233A-14, 2014.
6. I. T. Jolliffe. *A note on the use of principal components in regression*. Journal of the Royal Statistical Society. Series C (Applied Statistics), 31(3):300–303, 1982.
7. E. Bair, T. Hastie, P. Debashis, and R. Tibshirani. *Prediction by supervised principal components*. Journal of the American Statistical Association, 101(473):119–137, 2006.