

USER GUIDE



GUANIN

GUI-driven Analysis for Nanostring Interactive Normalization

Montoto-Louzao J, Gómez-Carballa A, Bello X, Martínón-Torres F, Salas A. [Paper details] [DOI]
Copyright (C) 2022. This program comes with absolutely no warranty, see 'license' for details.

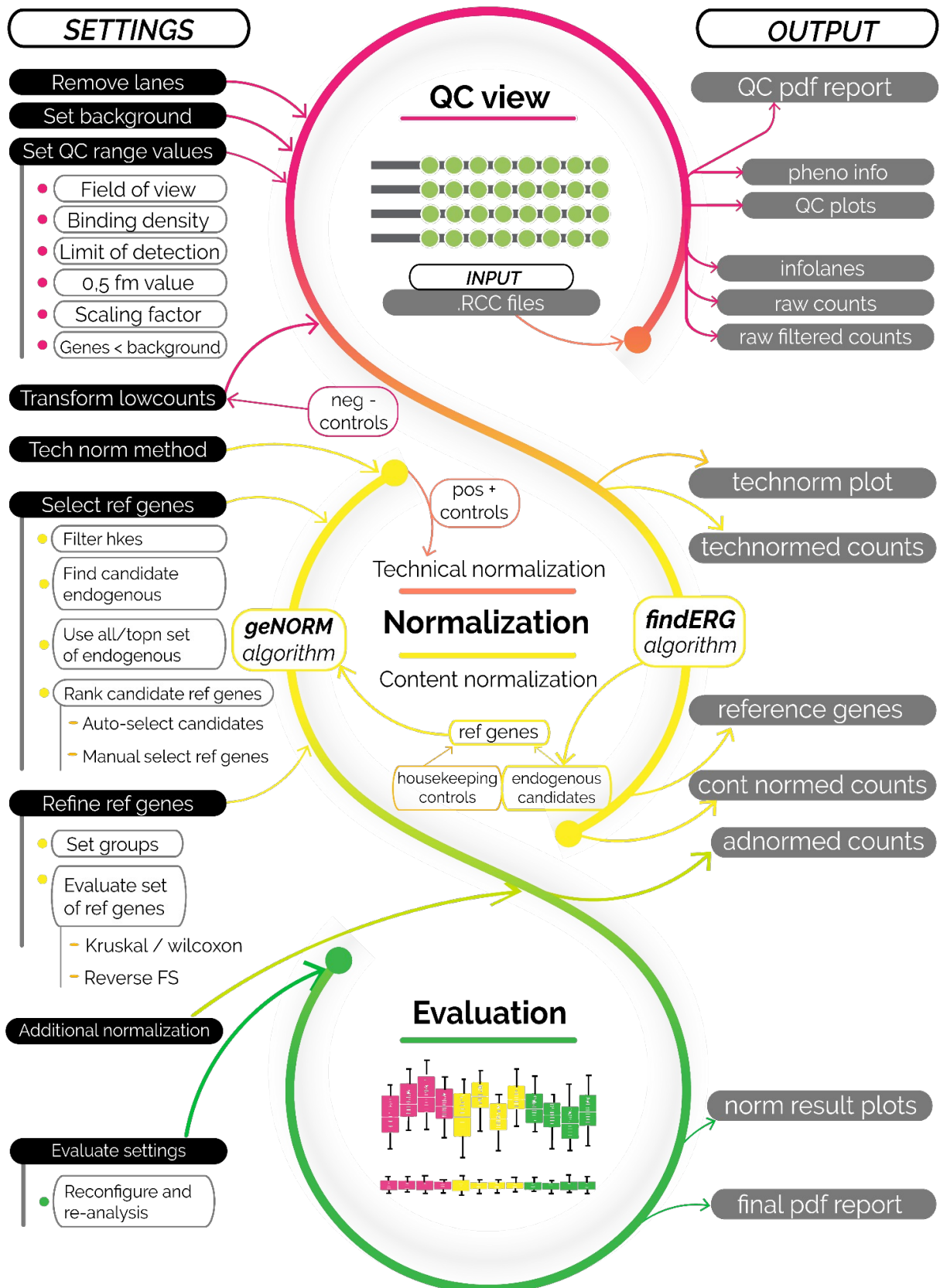


Based on GUANIN version 1.2.2: 04/07/2023

User guide version: 1.0: 05/07/2023

Please cite:

[Tal cual esto lo otro, info de la publicación]



WORKFLOW DESCRIPTION

LOADING DATA...

Input for normalization are raw .RCC files. No other Nanostring files need to be provided, neither preprocessing with nSolver or other software is required.

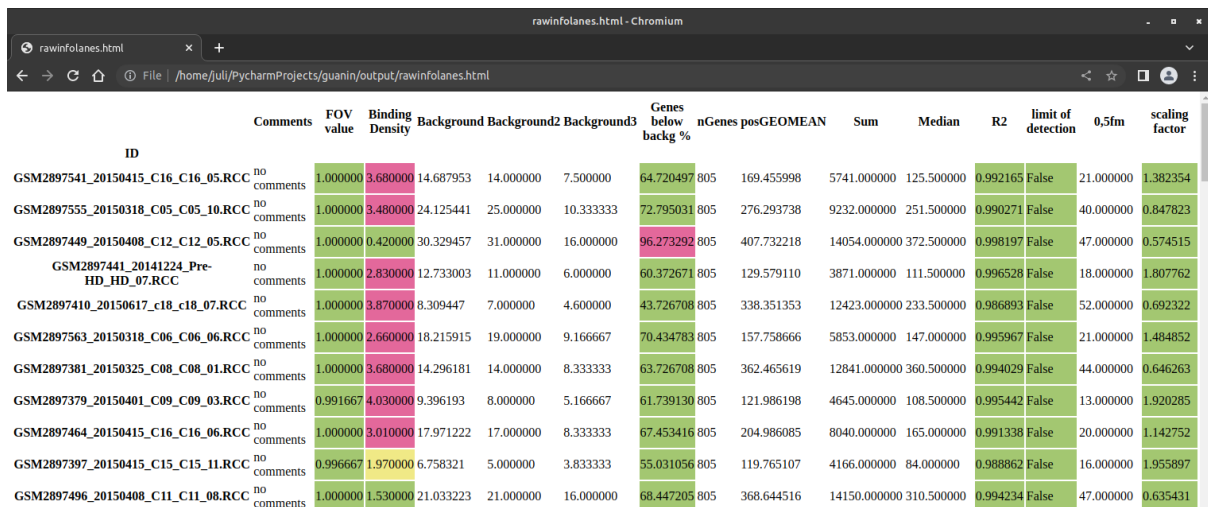
Sample info [condition/group] can be provided to refine content normalization. Although it's recommended, it's not mandatory. This sample info .csv file can be created using a text editor or spreadsheet software such as Excel. It should contain 2 columns: "SAMPLE" (that needs to match sample identification option (sample ID or file name) and "GROUP".

This is intended to represent the phenotype or the condition to study, and should not contain information about batches or replicates.

Both folder and groups .csv file are example dataset 1, so no further work is needed to start to get in touch with the program.

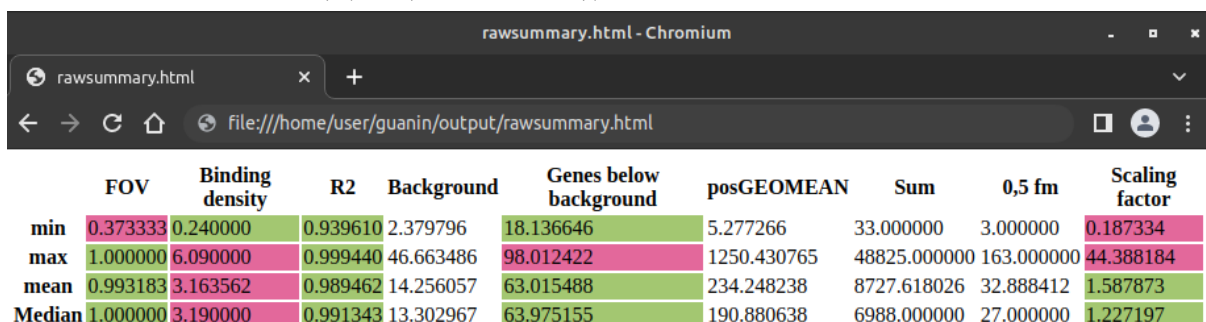
If "sample ID" field in the .RCC files have been declared and you trust this information (no duplicates, etc), "sample ID" is recommended to be used as sample identifier. Else, filename is default parameter as it is foolproof.

→OUTPUT: rawinfofiles.csv (sample information)



| ID | Comments | FOV value | Binding Density | Background | Background2 | Background3 | Genes below backg % | nGenes | posGEOMEAN | Sum | Median | R2 | limit of detection | 0,5fm | scaling factor |
|--------------------------------------|-------------|-----------|-----------------|------------|-------------|-------------|---------------------|--------|------------|--------------|------------|----------|--------------------|-----------|----------------|
| GSM2897541_20150415_C16_C16_05.RCC | no comments | 1.000000 | 3.680000 | 14.687953 | 14.000000 | 7.500000 | 64.720497 | 805 | 169.455998 | 5741.000000 | 125.500000 | 0.992165 | False | 21.000000 | 1.382354 |
| GSM2897555_20150318_C05_C05_10.RCC | no comments | 1.000000 | 3.480000 | 24.125441 | 25.000000 | 10.333333 | 72.795031 | 805 | 276.293738 | 9232.000000 | 251.500000 | 0.990271 | False | 40.000000 | 0.847823 |
| GSM2897449_20150408_C12_C12_05.RCC | no comments | 1.000000 | 0.420000 | 30.329457 | 31.000000 | 16.000000 | 96.273292 | 805 | 407.732218 | 14054.000000 | 372.500000 | 0.998197 | False | 47.000000 | 0.574515 |
| GSM2897441_20141224_Pre-HD_HD_07.RCC | no comments | 1.000000 | 2.830000 | 12.733003 | 11.000000 | 6.000000 | 60.372671 | 805 | 129.579110 | 3871.000000 | 111.500000 | 0.996528 | False | 18.000000 | 1.807762 |
| GSM2897410_20150617_c18_c18_07.RCC | no comments | 1.000000 | 3.870000 | 8.309447 | 7.000000 | 4.600000 | 43.726708 | 805 | 338.351353 | 12423.000000 | 233.500000 | 0.986893 | False | 52.000000 | 0.692322 |
| GSM2897563_20150318_C06_C06_06.RCC | no comments | 1.000000 | 2.660000 | 18.215915 | 19.000000 | 9.166667 | 70.434783 | 805 | 157.758666 | 5853.000000 | 147.000000 | 0.995967 | False | 21.000000 | 1.484852 |
| GSM2897381_20150325_C08_C08_01.RCC | no comments | 1.000000 | 3.680000 | 14.296181 | 14.000000 | 8.333333 | 63.726708 | 805 | 362.465619 | 12841.000000 | 360.500000 | 0.994029 | False | 44.000000 | 0.646263 |
| GSM2897379_20150401_C09_C09_03.RCC | no comments | 0.991667 | 4.030000 | 9.396193 | 8.000000 | 5.166667 | 61.739130 | 805 | 121.986198 | 4645.000000 | 108.500000 | 0.995442 | False | 13.000000 | 1.920285 |
| GSM2897464_20150415_C16_C16_06.RCC | no comments | 1.000000 | 3.010000 | 17.971222 | 17.000000 | 8.333333 | 67.453416 | 805 | 204.986085 | 8040.000000 | 165.000000 | 0.991338 | False | 20.000000 | 1.142752 |
| GSM2897397_20150415_C15_C15_11.RCC | no comments | 0.996667 | 1.970000 | 6.758321 | 5.000000 | 3.833333 | 55.031056 | 805 | 119.765107 | 4166.000000 | 84.000000 | 0.988862 | False | 16.000000 | 1.955897 |
| GSM2897496_20150408_C11_C11_08.RCC | no comments | 1.000000 | 1.530000 | 21.033223 | 21.000000 | 16.000000 | 68.447205 | 805 | 368.644516 | 14150.000000 | 310.500000 | 0.994234 | False | 47.000000 | 0.635431 |

→OUTPUT: rawsummary (samples summary).



| | FOV | Binding density | R2 | Background | Genes below background | posGEOMEAN | Sum | 0,5 fm | Scaling factor |
|--------|----------|-----------------|----------|------------|------------------------|-------------|--------------|------------|----------------|
| min | 0.373333 | 0.240000 | 0.939610 | 2.379796 | 18.136646 | 5.277266 | 33.000000 | 3.000000 | 0.187334 |
| max | 1.000000 | 6.090000 | 0.999440 | 46.663486 | 98.012422 | 1250.430765 | 48825.000000 | 163.000000 | 44.388184 |
| mean | 0.993183 | 3.163562 | 0.989462 | 14.256057 | 63.015488 | 234.248238 | 8727.618026 | 32.888412 | 1.587873 |
| Median | 1.000000 | 3.190000 | 0.991343 | 13.302967 | 63.975155 | 190.880638 | 6988.000000 | 27.000000 | 1.227197 |

PRELIMINARY QC INSPECTION...

- Background determination:

Negative controls are included in nanostring panels in order to set a threshold of non-expressed genes: the background. It can be more or less restrictive, depending on the characteristics of the experiment.

Default background is calculated as the mean of the negative controls + twice the standard deviation. Max of negative controls or mean of negative controls can be used. Also, Guanin implements a new method of selection alternative negative controls, useful in the case there is a problem with predefined negative controls (i.e: they are expressed). In this case, an alternative background is calculated from low-expressed genes among the endogenous. Additionally, background can be set manually.

- Background correction:

Once background is set, there are several options to handle values below background (low counts):

- a Assimilate to background: Sets all values $<$ background as equal to background.
- b Subtract background: Sets all values as value - background, assigning 0 value to genes expressed equally or lower than background level. (default)
- c Skip: Ignores background correction by not performing any correction.

- Sample inspection:

Samples with QC abnormalities can be a) flagged or b) removed from the analysis. QC flag values can be set:

- a % of genes below background: A big amount of genes being expressed below background can relate problems with the sample. By default, Guanin flags samples that have more than 80% of their genes less expressed than the background. Lower % values can refine more strictly.

Samples with FOV, BD, linearity or scaling factor values below or above recommended levels can be related with errors:

- b Field of view: Default values: [0.75 - 1]
- c Binding density: Default values: [0.1 - 1.8]
- d Linearity: Default values: [0.75 - 1]
- e Scaling factor: Default values: [0.3 - 3]

Additionally, samples can be manually selected to remove from the analysis. SampleIDs or filenameIDs should not contain spaces, and files manually selected to remove are input separated by spaces ("sample1 sample2 sample3").

Default values for preliminary QC inspection need to be set for plotting and calculations. Then, parameters can be modified in order to re-run QC and refine QC thresholds.

→OUTPUT_files:

rawcounts.csv (raw matrix counts of endogenous genes)

rawfcounts.csv (raw matrix counts of filtered samples, endogenous genes)

dfhkecounts.csv (raw matrix counts of housekeeping genes)

posnegcounts.csv (raw matrix counts of positive and negative controls)

→OUTPUT_reports: QC inspection pdf in output/reports (ejemplo adjunto)

Summary infolanes (filtered lanes info)

QCflags.txt (info about what samples have been flagged/discarded and why)

TECHNICAL NORMALIZATION

In order to perform technical normalization replicates with known concentration are used.

These positive controls are used to calculate a lane-specific scaling factor that can be derived from:

- a posgeomean of positive controls (default)
- b summation of positive controls
- c median of positive controls

Note: *Although Nanostring nCounter performs first background correction and after that technical normalization, other tools that throw better normalization results apply background correction over technically normalized data, Guanin has shown to obtain better normalization results with this procedure too. For this reason, although for the user this is a conceptually posterior process, QC inspection and technical normalization are performed together.

→OUTPUT_files:

tnormcounts.csv (matrix counts after technical normalization)

CONTENT NORMALIZATION

Choosing appropriate housekeeping genes is crucial for normalization. That's why content normalization can be performed using:

- a Default panel housekeeping genes (filtered or not)
- b Default housekeeping + best endogenous candidate reference genes
- c All endogenous genes
- d Manual selection of genes

As housekeeping genes are supposed to have stable high expression on every sample, it is recommended to discard any of them if it is lowly expressed at any sample. Default value for exclusion is set as 50, but higher values are encouraged.

Including most promising endogenous genes that can be used as housekeeping is a Guanin unique feature, that uses findERG (ERgene) algorithm. It finds among endogenous genes the most stably and high expressed among all lanes.

As a standard panel includes 12 housekeeping genes, a number of endogenous candidates is encouraged to be included between 4 and 12 (default 6). This endogenous candidates enter with housekeeping in a evaluation pool and, depending on the results of the final reference genes selection, can be reasonable to re-run including more or less (if housekeeping are bad and all endogenous are chosen over them, for example, more endogenous could be included).

For this evaluation of candidate reference genes, geNorm algorithm is used, retrieving a ranked list of best candidate reference genes, and calculating the optimal number to use. Indeed, a brand new content normalization approach is provided, using ponderated weights of every reference gene based on its ranking geNorm value.

In this way, several combination of parametrizations can be used and refined:

- a What genes to include in the pool as candidate ref genes?
 - a.i Housekeeping only
 - a.ii Housekeeping + n best endogenous (default)
- b Wich genes from the pool will be selected?
 - b.i geNorm n and gene names intelligent selection (default)
 - b.ii genorm n and gene names intelligent ponderated selection (best genes contribute more to normalization that bad genes) [exclusife feature]
 - b.iii Top n best from genorm ranking
- c Avoid geNorm calculations, use...
 - c.i All endogenous genes (useful when default housekeeping genes are bad)

- c.ii Top n most expressed genes (useful when default housekeeping genes are bad)
- c.iii Manual selection of reference genes

Once reference genes are chosen, Guanin allows to perform an [exclusive feature] additional filtering in order to ensure they are not differentially expressed between groups.

If groups are set, they can be flagged or filtered if Kruskal-Wallis or Wilcoxon's tests reveal that they are significantly differentially expressed among groups. In the case of Wilcoxon's tests, Guanin performs every pair of group comparisons (if groups > 2).

Additionally, a (only informative) reverse feature selection ranking can be shown in order to dilucidate what combination of reference genes are more significantly (and how much) revealing a relation with group predicting. Can be interesting to interpret wich genes are the ones the machine learning algorithm considers the most representative and discards them the last, and with what accuracy the algorithm can predict to wich groups belongs a sample from one or a group of reference genes. This accuracy should be close to $1/j$, where j is the number of groups declared in our experiment, and approximately could be warning that something wrong is happening if accuracy is closer to $2*(1/j)$ than to $1/j$.

In case of individual genes, this predictive ability will be spotted by Kruskal or Wilcoxon filtering, so no relevant results should be thrown. But it is in case of additive effect of a combination of genes that results in predictive vinculating effects with the output group when we can spot problems with our reference genes selection.

OUTPUT_files: refgenes.csv (count matrix of chosen reference genes)

rnormcounts.csv (content-normalized count matrix)

OUTPUT_reports: ranking_kruskal_wilcox.csv (p values of association of reference gene expression with groups. Values > 0.05 for all association are encouraged)

| ranking_kruskal_wilcox.html - Chromium | | | | | | | |
|--|----------|----------|----------|----------|----------|----------|----------|
| file:///home/user/output/ranking_kruskal_wilcox.html | | | | | | | |
| Kruskal p-value wilcox: Moderate/Severe wilcox: Moderate/Mild wilcox: Moderate/Control wilcox: Severe/Mild wilcox: Severe/Control wilcox: Mild/Control | | | | | | | |
| Genes | | | | | | | |
| S100A9 | 0.001021 | 0.271899 | 0.044164 | 0.005616 | 0.004906 | 0.000636 | 0.420113 |
| IFITM1 | 0.001500 | 0.498962 | 0.956622 | 0.000978 | 0.662956 | 0.000906 | 0.003559 |
| GAPDH | 0.001685 | 0.204894 | 0.141940 | 0.001551 | 0.043311 | 0.001791 | 0.072097 |
| EEF1G | 0.009353 | 0.017960 | 0.586491 | 0.644392 | 0.004906 | 0.000636 | 0.852404 |
| ABCF1 | 0.014935 | 0.108319 | 0.114705 | 0.391337 | 0.010013 | 0.283051 | 0.009195 |
| RPL19 | 0.018095 | 0.014248 | 0.785650 | 0.692379 | 0.007939 | 0.003415 | 0.420113 |
| ALAS1 | 0.025680 | 0.612090 | 0.355133 | 0.012223 | 0.165518 | 0.008415 | 0.106864 |
| TUBB | 0.058973 | 0.176296 | 0.744154 | 0.029559 | 0.284719 | 0.625585 | 0.013113 |
| PPIA | 0.071057 | 0.062979 | 0.663459 | 0.597843 | 0.029318 | 0.008415 | 0.619796 |
| HPRT1 | 0.071141 | 0.128190 | 0.785650 | 0.004578 | 0.191040 | 0.922258 | 0.153753 |
| G6PD | 0.078426 | 0.108319 | 0.586491 | 0.644392 | 0.019405 | 0.011170 | 0.756495 |
| SDHA | 0.095822 | 0.310494 | 0.663459 | 0.113532 | 0.191040 | 0.494525 | 0.013113 |
| POLR1B | 0.104936 | 0.204894 | 0.462764 | 0.210270 | 0.062574 | 0.625585 | 0.034980 |
| IL32 | 0.135682 | 0.062979 | 0.446359 | 0.029559 | 0.284719 | 0.922258 | 0.321062 |
| GUSB | 0.182524 | 0.672604 | 0.586491 | 0.040946 | 0.781511 | 0.171857 | 0.094038 |
| HLA-A | 0.182960 | 0.498962 | 0.624463 | 0.086457 | 0.250547 | 0.063709 | 0.214847 |
| TBP | 0.194641 | 0.075927 | 0.703389 | 0.509651 | 0.062574 | 0.171857 | 0.352236 |
| HLA-B | 0.252409 | 0.176296 | 0.703389 | 0.509651 | 0.074592 | 0.118420 | 0.321062 |
| CTSS | 0.272561 | 0.090969 | 0.384145 | 0.428794 | 0.191040 | 0.118420 | 0.950549 |
| POLR2A | 0.356808 | 0.398025 | 0.956622 | 0.468257 | 0.165518 | 0.063709 | 0.456750 |
| B2M | 0.479580 | 0.398025 | 0.956622 | 0.428794 | 0.250547 | 0.845252 | 0.136641 |
| FCGR3A/B | 0.536637 | 0.310494 | 0.624463 | 1.000000 | 0.191040 | 0.241567 | 0.576738 |
| OAZ1 | 0.549882 | 0.932647 | 0.210922 | 0.644392 | 0.219348 | 0.558185 | 0.576738 |
| PTPRC_all | 0.775771 | 0.554113 | 1.000000 | 0.428794 | 0.552297 | 0.845252 | 0.385261 |

metrics_reverse_feature_selection.csv (predictability of groups by association of several reference genes. Values similar to 1/groups are encouraged. High accuracy prediction with few combination of genes may indicate bad combination of reference genes

| | feature_idx | cv_scores | avg_score | feature_names | ci_bound | std_dev | std_err |
|----|--|----------------------------|-----------|--|----------|----------|----------|
| 17 | (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16) | [0.36904762 0.36666667] | 0.367857 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'TBP', 'SDHA', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS', 'HLA-B', 'PTPRC_all', 'HLA-A') | 0.005122 | 0.001190 | 0.001190 |
| 16 | (0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'TBP', 'SDHA', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS', 'HLA-B', 'PTPRC_all', 'HLA-A') | 0.158788 | 0.036905 | 0.036905 |
| 15 | (0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'TBP', 'SDHA', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS', 'HLA-B', 'PTPRC_all') | 0.158788 | 0.036905 | 0.036905 |
| 14 | (0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'TBP', 'SDHA', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS', 'HLA-B') | 0.158788 | 0.036905 | 0.036905 |
| 13 | (0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'TBP', 'SDHA', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 12 | (0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'TBP', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 11 | (0, 1, 2, 3, 4, 5, 6, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'POLR2A', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 10 | (0, 1, 2, 3, 4, 5, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'HPRT1', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 9 | (0, 1, 2, 3, 4, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'TUBB', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 8 | (0, 1, 2, 3, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'POLR1B', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 7 | (0, 1, 2, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'PPIA', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 6 | (0, 1, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'GUSB', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 5 | (0, 10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('G6PD', 'IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 4 | (10, 11, 12, 13) | [0.44047619 0.36666667] | 0.403571 | ('IL32', 'B2M', 'FCGR3A/B', 'CTSS') | 0.158788 | 0.036905 | 0.036905 |
| 3 | (10, 11, 13) | [0.31071429 0.4375] | 0.374107 | ('IL32', 'B2M', 'CTSS') | 0.272757 | 0.063393 | 0.063393 |
| 2 | (11, 13) | [0.36666667 0.28333333] | 0.325000 | ('B2M', 'CTSS') | 0.179277 | 0.041667 | 0.041667 |
| 1 | (11,) | [0.33333333 0.325] | 0.329167 | ('B2M,') | 0.017928 | 0.004167 | 0.004167 |

ADDITIONAL NORMALIZATION...

Additional normalization can be useful in the case we need our data in an specific format, such as in a range to 0-1, etc. For that, it can be implemented:

- a quantile normalization
- b standarization

OUTPUT_files: adnormcounts.csv (count matrix of additionaly normalized data)

EVALUATION OF NORMALIZATION

In order to assess if normalization is offering reasonable results, two measures can be used that make use of relative log expression.

Relative log expression are useful for visualizing unwanted variation.

- a RLE plots, comparing pre-normalization and post-normalization. Narrow boxplots mean less unknown expression differences.
- b IQR, that can be used to numerically compare different normalization parametrizations that could suit our experiment.

OUTPUT_reports: norm_report.pdf (showing genorm results for reference genes chosen, RLE plots and IQR).

<https://i.imgur.com/TBTcTnm.png>

The screenshot displays the GUANIN: Nanostring Interactive Normalization software interface. The interface is divided into several panels:

- Loading data:** Select RCC files, Select groups file, Sample identifier.
- Technical normalization parameters:** Method for technical normalization: Use posgeomean.
- Content normalization parameters:** Filter housekeeping panel genes by min counts: 50, Include best endogenous as reference candidates: checked.
- Quality Control parameters:** Choose background: Alternative background (Filtered neg ctrls), Set manual background: 0, Background correction (low counts): Subtract background value, % of low counts for lane remove: 80, Min fov: 0.75, Max fov: 1.00, Min binding density: 0.10, Max binding density: 1.80, Min linearity: 0.75, Max linearity: 1.00, Min scaling factor: 0.30, Max scaling factor: 3.00.
- Normalization evaluation:** Run RLE plot, Run evaluation.
- Results:** A table showing genes and their normalized counts. The table includes columns for Genes, p-value, wilcox, and wilcox.

The results table shows a list of genes with their corresponding p-values and normalized counts. The genes listed are: S100A9, IFITM1, GAPDH, EEF1G, ABCF1, RPL19, ALAS1, POLR1B, HPRT1, TUBB, PPIA, SDHA, G6PD, IL32, GUSB, TBP, HLA-A, HLA-B, CTSS, POLR2A, B2M, FCGR3A/B, OAZ1, and PTPRC. The p-values are shown in the first column, and the normalized counts are shown in the second column.

D1: GSE183071 – Blood study of gene expression profiling on the nasal epithelium in COVID-19 severity.

This dataset consists on 54 samples including controls, mild, moderate and severe severity.

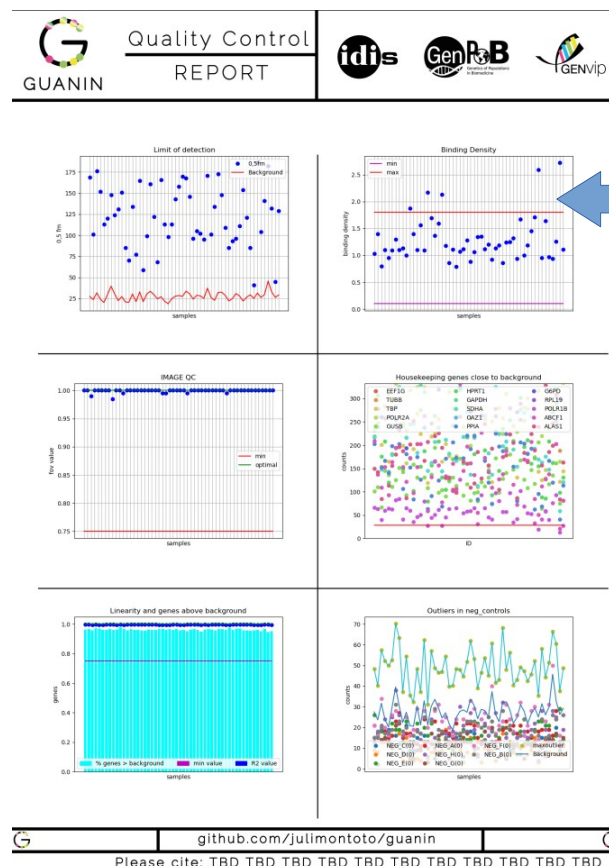
Raw data inspection may suggest there is a problem with binding density, while the rest of the samples are QC-ok.

| | FOV | Binding density | R2 | Background | Genes below background | posGEOMEAN | Sum | 0.5 fm | Scaling factor |
|--------|----------|-----------------|----------|------------|------------------------|-------------|--------------|------------|----------------|
| min | 0.984536 | 0.790000 | 0.994647 | 18.754359 | 13.980263 | 458.621941 | 13395.000000 | 41.000000 | 0.667838 |
| max | 1.000000 | 2.720000 | 0.999137 | 45.756282 | 32.565789 | 1726.146285 | 60421.000000 | 186.000000 | 2.513586 |
| mean | 0.999045 | 1.282963 | 0.996747 | 27.727611 | 22.947125 | 1152.785765 | 40192.481481 | 122.944444 | 1.087117 |
| Median | 0.990000 | 1.155000 | 0.997049 | 27.624700 | 23.273026 | 1165.965936 | 41032.500000 | 121.500000 | 0.988744 |

We can see that 5 samples (B0513, B0318, B0319, B0369, B0343) have high binding density value. We can choose to alter or not max binding density values to exclude this samples, as default option will be to discard them.

Rest of settings run by default, after QC filtering we can see after discarding 5 lanes all values are QC-ok.

At this point, we can view QC inspection plots and info about flagged/discarded lanes in the selected output folder:



After running default technical normalization, content normalization can be performed with few restrictions and no suggested endogenous reference genes.

We can see GUANIN filters POLR1B for having less than 50 counts (housekeeping genes are supposed to be expressed), but continues the analysis with the rest of the 14 genes, and after applying geNorm selection of the n reference genes, GUANIN has used 13 to perform content normalization.

Ref. genes selected (auto): ['EEF1G', 'OAZ1', 'POLR2A', 'G6PD', 'GAPDH', 'ALAS1', 'GUSB', 'TUBB', 'SDHA', 'ABCF1', 'HPRT1', 'TBP', 'PIA']

As we see, if we apply Wilcoxon filtering genes that can be associated to any of the conditions, only three valid housekeeping genes remain ['SDHA', 'TBP', 'ABCF10']. This is usually not enough for proper normalization, or even for genorm preprocess.

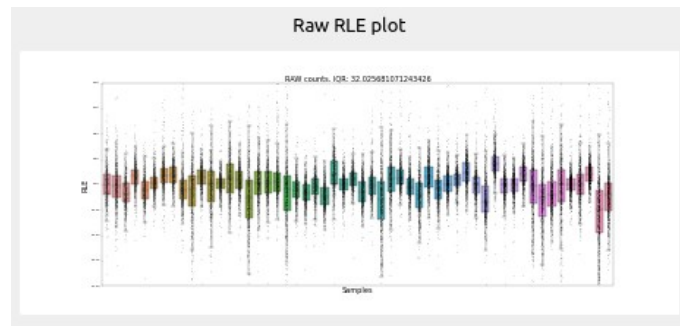
We can fix that selecting kruskal-wallis filter, which is less sensitive, but that could skew our results in the way of the related to a condition but selected anyways as reference genes.

In order to assess this common problematic, we can use ERgene to find best endogenous and include them in the pool that are validated by kruskal/wilcox and into genorm algorithm. Changing that parameters in guanin, including 6 most promising endogenous and wilcoxon filtering, we get a subselection of 5 genes, both endogenous and housekeeping, that can be used for content normalization: ['B2M', 'CTSS', 'PTPRC_all', 'SDHA', 'TBP'].

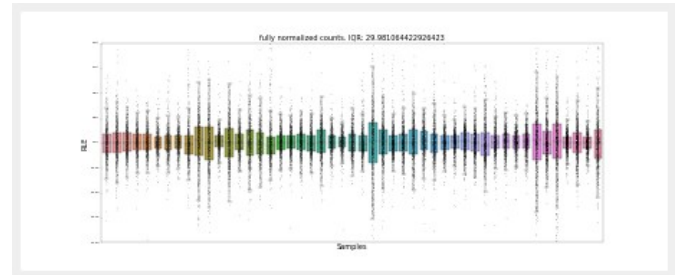
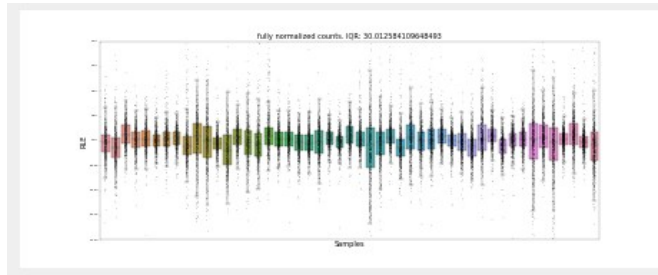
As we see, 2 out of 3 preselected housekeeping are included too, but GUANIN has found better company for them as reference genes for three endogenous genes.

When running evaluation, we expect to see narrower bars with means closer to the center. Nevertheless, this can also mean loss of biological variability (we want the experimental variability out, but not the biological one). So RLE plots are useful to have an idea of how the normalization behaves on different processes.

Note: for some methods that intend to remove all unwanted variation possible, RLE plots can be even narrower because of removing as variability as possible, with risk of removing biological variability. On the other hand, methods intended to target remove technical and biological variation preserve more general variation and have less cute RLE plots. So RLE plots are informative, but not definitive. Indeed, using all genes to content normalization is a method that is almost only used when others fail, as you can be losing significance on the results of the experiment, and usually, this method reports narrower RLE plots, because of removing “too much” variability.



Reference genes selected content normalization vs all genes normalization RLE plots



EXAMPLE DATASETS:

D2: GSE160208 – Gene expression in the brain of sporadic Creutzfeldt-Jakob disease patients (CJD), and normal controls (CT).

This dataset contains 47 samples from 2 groups disease and control.

For this dataset we see all samples are QC-ok, so we proceed with technical normalization.

Results on evaluation of reference genes (panel housekeeping and best endogenous), results in only 2 genes suitable to be reference genes for content normalization. As it is minimum required 3 genes, we may have several options:

- Lower the threshold of min counts for housekeeping, in case there is any low but stably expressed that can be rescued.
- Include more best endogenous as candidates (
- Introduce a selection of best bad genes (not very good idea as kruskal values of bad genes are very low)

Using first approach, from info in the QC report:

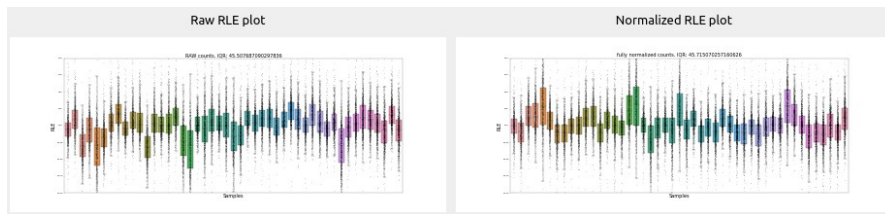


We can see there are a few genes behind 50 counts but above the background, so a threshold between 50 and the background should be a reasonable choice.

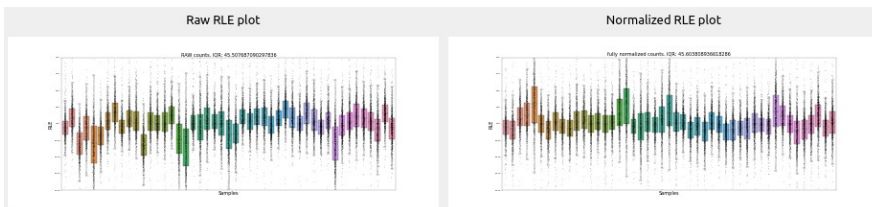
In the other hand, we can include 25 of the best endogenous genes instead of 6.

From this we get 1 more gene. We can decide to perform normalization with these 3 suitable genes, or proceed with alternative approaches like all expressed endogenous genes.

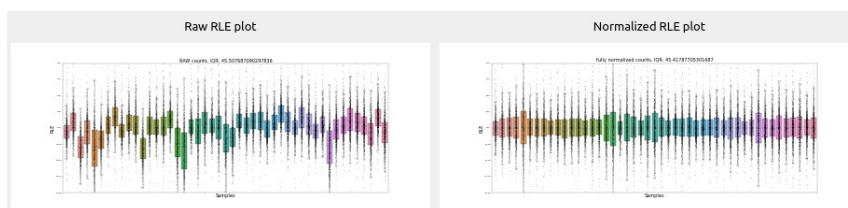
- Housekeeping normalization:



- Filtered and refined reference genes selected from housekeeping+ endogenous (3 genes) normalization



- All endogenous genes normalization:



The scientist can choose which normalization method better suits the experiment. In some special cases like this one, that maybe there is no “good” method for this data, differential expression can be analysed for several normalization approaches, and the scientist can choose in order of DE results.

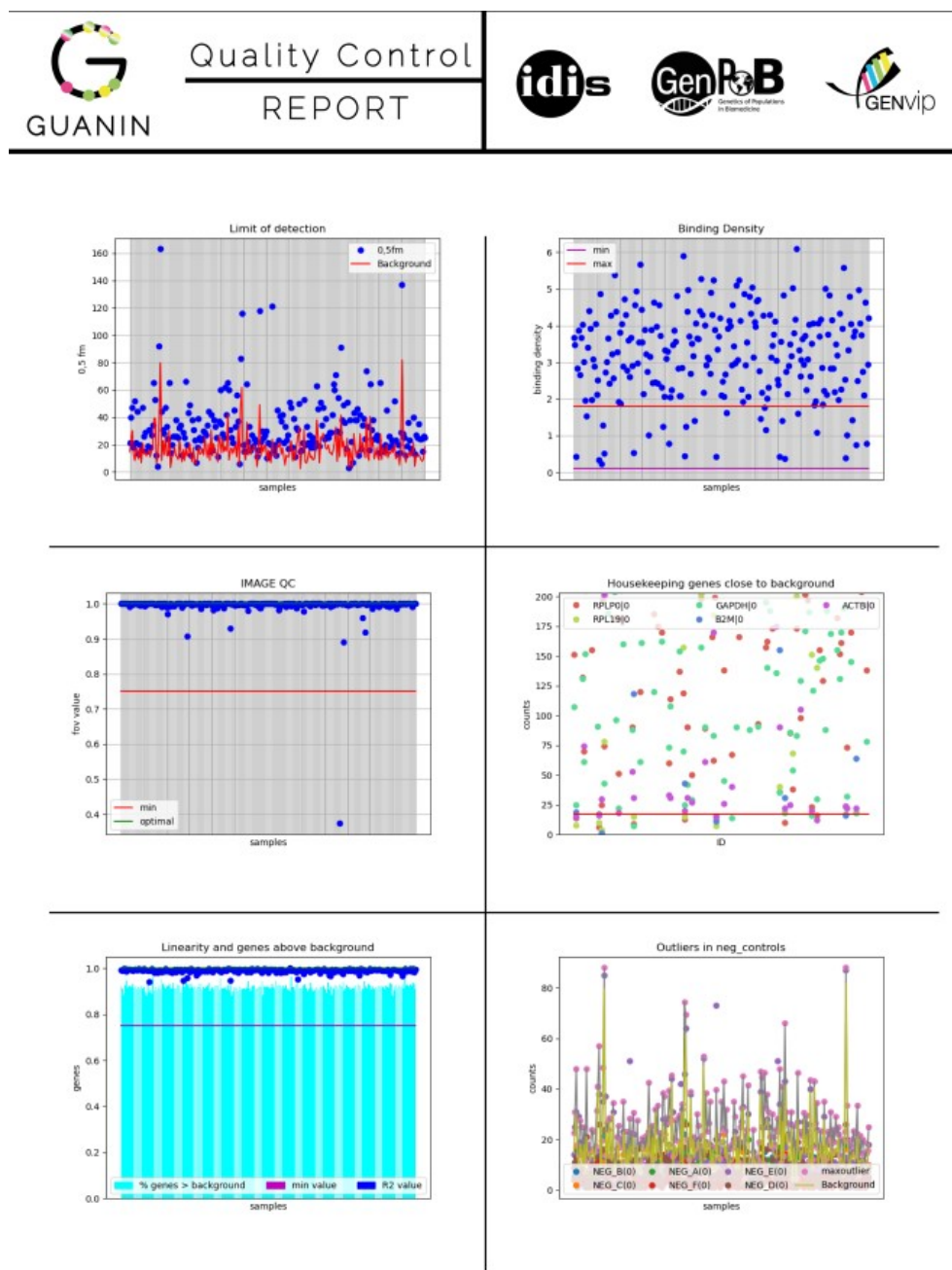
EXAMPLE DATASETS:

D3: GSE160208 – Gene expression in the brain of sporadic Creutzfeldt-Jakob disease patients (CJD), and normal controls (CT).

This dataset contains 233 samples from 2 groups disease and control.

This dataset, as many others found in databases, has a lot of QC problems that would be more difficult to adress with other normalization tools.

We can see in QC report that a lot of samples are above max binding density, some of them below limit of detection, one has very low fov value, there are several housekeeping genes close and below the background and there also are a lot of outliers in negative controls.



Discarding and repeating the experiment could be a reasonable choice. But if we would continue with the analysis we can tweak QC settings in order to allow some more than 14 out of the 233 to pass QC.

Probably the safest choice would be to set to 3 maximum binding density, as it is the biggest problem, hoping we get a good number of samples for both groups.

We also detect that around 200 samples have more than 50% of their genes with values below background, which may reveal a technical problem with the experiment.

Anyways as a “bad data” example, we can tweak:

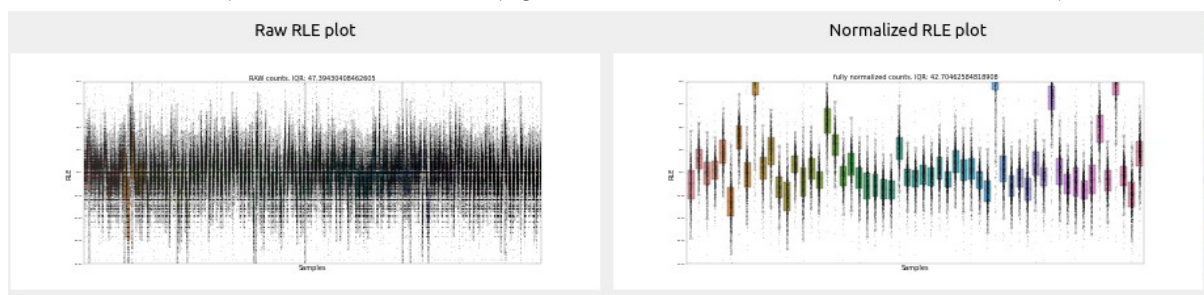
- Alternative background (as there are outliers in negative controls).
- 90% of low counts to remove (maybe there is very little expression in the panel, for some biological reason).
- 3 max fov.

With this, we have 57 samples Qc-“ok” in the analysis.

As some of housekeeping genes are not expressed, we will refine them with a selection of the most suitable endogenous (12 in this case, for example).

We can see only 5 housekeeping genes are suitable to be used as reference genes. Kruskal-wallis test discard none of the 17 preselected genes for being associated with the condition, so we let geNorm algorithm to choose how many and which genes are most suitable for the analysis. From these 17 preselected genes, it selects 13 as reference genes: 4 of the 5 housekeeping valid and 9 endogenous suitable to be used as reference genes.

Normalized RLE plot doesn't look very good either, which makes sense with bad QC data.



Note: Although modification of parameters and re-running blocks for interactive normalization should work, the combination of possibilities is untestable, so when you find the best normalization settings for your experiment we recommend a complete re-run of the full process.

CLI COMMANDS HELP:

- Argument: '-f', '--folder', *type=str*, *default=* pathlib.Path.cwd() / 'examples/d1_COV_GSE183071', *help=*'relative folder where RCC set is located.'
- Argument: '-minf', '--minfov', *type=float*, *default=*0.75, *help=*'set manually min fov for QC'
- Argument: '-maxf', '--maxfov', *type=float*, *default=*1, *help=*'set manually max fov for QC'
- Argument: '-minbd', '--minbd', *type=float*, *default=*0.1, *help=*'set manually min binding density for QC'
- Argument: '-maxbd', '--maxbd', *type=float*, *default=*1.8, *help=*'set manually max binding density for QC'
- Argument: '-minlin', '--minlin', *type=float*, *default=*0.75, *help=*'set manually min linearity for QC'
- Argument: '-maxlin', '--maxlin', *type=float*, *default=*1, *help=*'set manually max linearity for QC'
- Argument: '-minscf', '--minscalingfactor', *type=float*, *default=*0.3, *help=*'set manually min scaling factor for QC'
- Argument: '-maxscf', '--maxscalingfactor', *type=float*, *default=*3, *help=*'set manually max scaling factor for QC'
- Argument: '-swbrrq', '--showbrowserrawqc', *type=bool*, *default=*False, *help=*'pops up infolanes and qc summary'
- Argument: '-swbrq', '--showbrowserqc', *type=bool*, *default=*False, *help=*'pops up infolanes and qc summary')
- Argument: '-swbrcn', '--showbrowsercnorm', *type=bool*, *default=*False, *help=*'pops up infolanes and qc summary')
- Argument: '-lc', '--lowcounts', *type=str*, *default=*'sustract', *choices=*['skip', 'asim', 'sustract'], *help=*'what to do with counts below background?')
- Argument: '-mi', '--modeid', *type=str*, *default=*'filename', *choices=*['sampleID', 'filename', 'id+filename'], *help=*'choose sample identifier. sampleID: optimal if assigned in rccs. filenames: easier to be unique. id+filename: care with group assignment coherence')
- Argument: '-mv', '--modeview', *type=str*, *default=*'view', *choices=*['justrun', 'view'], *help=*'choose if plot graphs or just run calculations')
- Argument: '-tnm', '--tecnormeth', *type=str*, *default=*'posgeomean', *choices=*['posgeomean', 'Sum', 'Median', 'regression'], *help=*'choose method for technical normalization')
- Argument: '-reg', '--refendgenes', *type=str*, *default=*'endhkes', *choices=*['hkes', 'endhkes'], *help=*'choose refgenes, housekeeping, or hkes and endogenous')
- Argument: '-re', '--remove', *type=str*, *nargs=*'+', *default=*None, *help=*'lanes to be removed from the analysis')
- Argument: '-bg', '--background', *type=str*, *default=*'Background', *choices=*['Background', 'Background2', 'Background3', 'Backgroundalt'], *help=*'choose background: b1=meancneg+(2*std), b2=maxcneg, b3=meancneg, balt=')
- Argument: '-pbb', '--pbelowbackground', *type=int*, *default=*85, *help=*'if more than %bb genes are below background, sample gets removed from analysis')
- Argument: '-mbg', '--manualbackground', *type=float*, *default=*None, *help=*'set manually background')
- Argument: '-crg', '--chooserefgenes', *type=str*, *nargs=*'+', *default=*None, *help=*'list of strings like. choose manually reference genes to use over decided-by-program ones')
- Argument: '-fgv', '--filtergroupvariation', *type=str*, *default=*'filterkrus', *choices=*['filterkrus', 'filterwilcox', 'flagkrus', 'flagwilcox', 'nofilter'], *help=*'filter or flag preselected ref genes by significant group-driven differences? needs groups to be declared')
- Argument: '-fsn', '--featureselectionneighbors', *type=float*, *default=*4, *help=*'number of neighbors for feature selection analysis of refgenes. Recommended 3-6')
- Argument: '-g', '--groups', *type=str*, *default=*'yes', *choices=*['yes', 'no'], *help=*'defining groups for kruskal/wilcox/fs analysis?')
- Argument: '-ne', '--numend', *type=int*, *default=*6, *help=*'number of endogenous to find by ERgene to include in analysis to check viability as refgenes')
- Argument: '-ar', '--autorename', *type=str*, *default=*'off', *choices=*['on', 'off'], *help=*'turn on when sample IDs are not unique, be careful on sample identification detail')
- Argument: '-cn', '--contnorm', *type=str*, *default=*'refgenes', *choices=*['ponderaterefgenes', 'refgenes', 'all', 'topn'])
- Argument: '-an', '--adnormalization', *type=str*, *default=*'no', *choices=*['no', 'standardization'],

'quantile'], *help*='perform additional normalization? standarization and quantile normalization available')

- Argument: '-tn', '--topngenestocontnorm', *type*=int, *default*=100, *help*='set n genes to compute for calculating norm factor from top n expressed endogenous genes')
- Argument: '-mch', '--mincounthkes', *type*=int, *default*=80, *help*='set n min counts to filter hkes candidate as refgenes')
- Argument: '-nrg', '--nrefgenes', *type*=int, *default*=None, *help*='set n refgenes to use, overwriting geNorm calculation')
- Argument: '-lr', '--laneremover', *type*=str, *default*='yes', *choices*=['yes', 'no'], *help*='option to perform analysis with all lanes if set to no')
- Argument: '-grn', '--groupsinnormgenes', *type*=str, *default*='no', *choices*=['yes', 'no'], *help*='want groups to be specified in last column of rnormgenes dataframe?')
- Argument: '-lo', '--logarizedoutput', *type*=str, *default*='10', *choices*=['2', '10', 'no'], *help*='want normed output to be logarized? in what logbase?')
- Argument: '-le', '--logarizeforeval', *type*=str, *default*='10', *choices*=['2', '10', 'no'], *help*='logarithm base for RLE calculations')
- Argument: '-gf', '--groupsfile', *type*=str, *default*='examples/groups_d1_COV_GSE183071.csv', *help*='enter file name where groups are defined')
- Argument: '-st', '--start_time', *type*=float, *default* = time.time())
- Argument: '-cs', '--current_state', *type*=str, *default*='Ready')
- Argument: '-ftl', '--firsttransformlowcounts', *type*=bool, *default*=True)
- Argument: '-of', '--outputfolder', *type*=str, *default*= tempfile.gettempdir() + '/guanin_output')
- Argument: '-sll', '--showlastlog', *type*=bool, *default* = False)