GUANIN: GUi-driven Analyzer for NanoString Interactive Normalization

Julián Montoto-Louzao, Alba Camino-Mera, María J Martín, and Antonio Salas

Genetics, Vaccines, Infections and Pediatrics Research Group (GENVIP) - Instituto de Investigación Sanitaria - Hospital Clínico Universitario de Santiago, USC, 15782, Santiago de Compostela, Spain
INCIFOR, Facultade de Medicina - GenPoB - Instituto de Investigación Sanitaria - Hospital Clínico Universitario de Santiago, USC, 15782, Santiago de Compostela, Spain
Grupo de Arquitectura de Computadores, Centro de Investigación CITIC, Universidade da Coruña, 15071 A Coruña, Spain
Correspondence: antonio.salas@usc.es

DOI: *https://doi.org/10.17979/spudc.000024.09*

Abstract: Most tools for NanoString data normalization, aside from the default NanoString nCounter software, are R packages that focus on technical normalization but lack configurable parameters. However, content normalization is the most sensitive, experiment-specific, and relevant step to preprocess NanoString data. Currently, this step requires the use of multiple tools and deep management and understanding of all the normalization process by the researcher. To simplify this crucial step, we have developed GUANIN, a complete normalization tool that offers a wide variety of options to introduce, filter, choose, and evaluate reference genes for content normalization. GUANIN allows, among other features, introducing reference genes from an endogenous subset, a useful approach that addresses the problems associated with the selection of housekeeping genes only. GUANIN allows specific and straightforward normalization approach for each experiment, using a wide variety of parameters with suggested adjustments. GUANIN outperforms other available methods in terms of normalization, especially when comparison groups are defined beforehand, and allows the researcher to comprehensively interact with the preprocessing process without programming knowledge.

1 Introduction

NanoString Geiss et al. (2008) is a molecular barcoding platform for quantification of direct nucleic acid hybridization of RNA in tissue samples. It reports actual counts of sequences of interest through image analysis. As no amplification is needed, it avoids potential bias introduced by reverse transcription, striking a balance between the limitations of RNA-seq and microarrays. Due to its robust performance, NanoString is mainly used in experiments involving low quality samples, and/or tissue samples for the identification of nucleic acid presence, where proper quality control (QC) and normalization are crucial to maintain the accuracy of the experiments Gagnon-Bartsch and Speed (2012).

The NanoString nCounter platform nSolver offers a Graphical User Interface (GUI) addressing background correction, positive control (technical) normalization, and housekeeping normalization. However, nSolver lacks a comprehensive configuration for content normalization or the evaluation of normalization results, among other features.

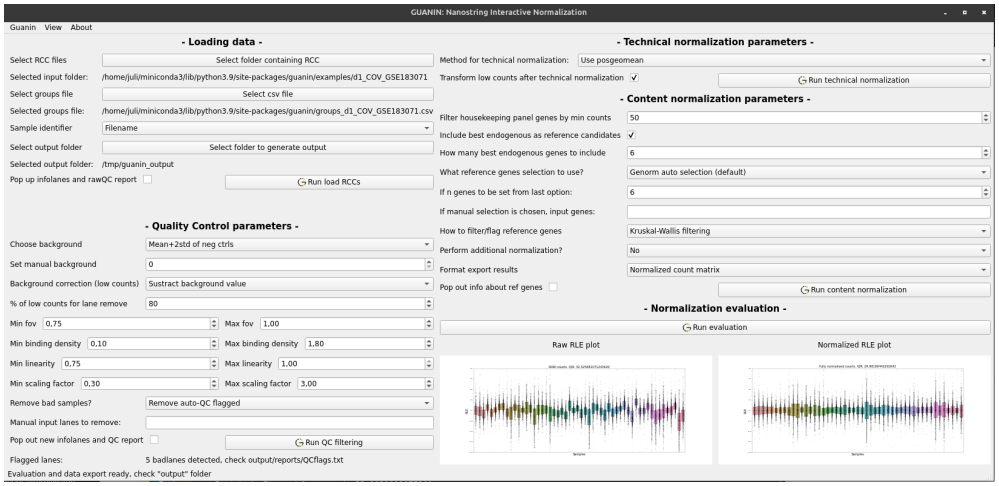

Figure 1: Main screen of GUANIN

Apart from nCounter NanoString native platform, there are several R packages that address the preprocessing of NanoString data, such as NACHO Canouil et al. (2020), NanoStringNorm Waggott et al. (2012), NanoStringDiff Wang et al. (2016), RUV-III Molania et al. (2019) or RCRNorm Jia et al. (2019). Most of them focus their robustness on a single issue, such as technical normalization, visualization, or differential expression. Therefore, a comprehensive normalization analysis would involve the use of several of these packages, which would require in-depth knowledge of the field, including data management and migration from one package to another, and where data handling problems usually arise.

Furthermore, previous approaches mainly focus on complete technical normalization (experiment-wise and sample-wise variability), while content normalization (gene-wise variability) approach is limited, usually offering selection among the housekeeping genes only.

To offer a comprehensive, easy to use, interactive pipeline for NanoString preprocessing we created GUANIN, a flexible and adaptable tool that allows the users to adjust the normalization process to the characteristics of their experiments.

GUANIN is implemented in Python and includes two user interfaces: a Command-Line Interface (CLI) and a very easy-to-use GUI. It is available through the official Python PyPI repository (*https://pypi.org/project/guanin*) and it can be installed with a single command (`pip install guanin`) in Linux, MacOS and Windows systems.

2 GUANIN overview

GUANIN enables users to easily detect and manage difficulties for normalization and QC problems within the experiment through wide flexible input of data, flexible QC, complete technical normalization, improved content normalization with an integration of several new and well-known methods, exhaustive evaluation of reference genes and easy evaluation through in-built plots and pdf reports.

Figure 1 shows the initial screen of the GUI of GUANIN. As can be seen in the figure, expert users can enter a large amount of parameters to fully adapt the normalization process to their experiments, but default values are also provided to help novice users to use the tool. The GUANIN GUI was built using PyQt6 Computing (2023), one of the most popular libraries for the development of Python graphical applications.

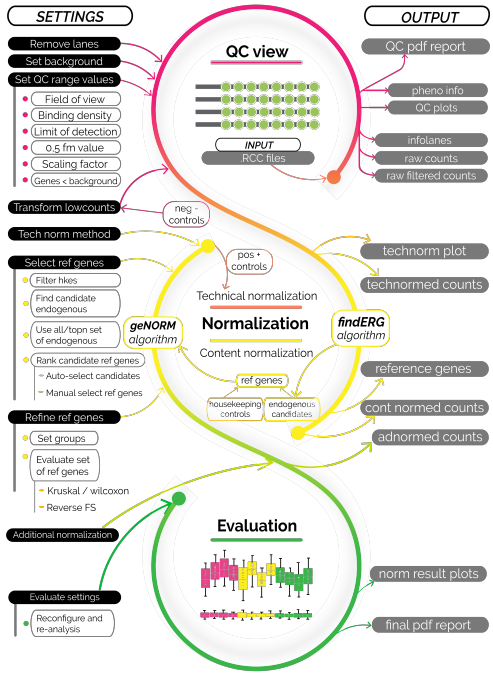

Figure 2: Scheme of GUANIN full workflow

3 GUANIN workflow

Figure 2 shows GUANIN workflow. It starts with load and inspection of input data (RCC files), continues with technical normalization (assessing experimental variations), background correction and content normalization (assessing biological variability). Additionally, it offers the possibility of performing additional normalization, formatting output data and evaluating the normalization process.

3.1 Step 1: Loading RCC files and generating a QC report

Right after loading input files, the first analysis reports the inherent information from the experiment needed to perform adequate normalization through a QC report. From there, QC parameters such as background, lanes to remove, or QC acceptance ranges can be recursively modified until an optimal QC status is achieved, allowing the normalization step to begin.

3.2 Step 2: Normalization (main step)

In contrast to the nSolver pipeline, GUANIN's default workflow performs technical normalization before background correction, as it has shown improved normalization results Lin et al. (2016). Several methods of background calculation and technical normalization are offered.

For content normalization, a set of reference genes needs to be chosen. In addition to default housekeeping genes, we introduce a new approach to select candidate reference genes among the endogenous ones, as it is a common issue that housekeeping genes are not suitable for the experiment. We utilized ERGene Zeng et al. (2020), a Python library for screening endogenous reference genes. The candidate reference genes, including n selected endogenous genes and panel housekeeping genes, are evaluated using a geNORM-based algorithm Vandesompele

	nSolver	NanoStringNorm	NACHO	RCRnorm	RUV-III	Guanin
Platform	GUI	R package	R package	R package	R package	GUI
Visualization	None	None	Shinyapp	None	None	QC pdf report
Background	Manual/auto	Manual/auto	Auto	Auto	Auto	Manual / 4 auto
Tech. norm.	Geomean normfactor	Geomean / sum normfactor	Geomean normfactor and glm	Regression model	Remove unwanted variation algorithm	Geomean/sum/med normfactor
Remove outliers	Manual/flagged	Manual	By parameter	None	Manual	Manual/auto
Housekeeping filtering	Manual	Manual	Manual	None	None	Throrough
Reference genes available	Default / manual housekeeping	Default / manual housekeeping	Default / manual housekeeping	Regression model housekeeping	None	FindERG endogenous + housekeeping
Reference genes selection	Manual	Manual	Manual	Manual	None	geNorm selection + filter and rank
Additional normalization	None	Quantile and z-score	None	None	None	Quantile and standarization
Unwanted variation removed	Low	Low	Medium	Medium	High	High
Evaluation	None	None	None	None	None	Mean IQR and RLE plots
User friendlyness	Easy	Medium	Medium	Hard	Hard	Easy

Figure 3: Comparative on usability amongst main NanoString normalizing tools and GUANIN

et al. (2002); Zhong (2019) to select n genes for driven content normalization. Indeed, the candidate reference genes are filtered or flagged by a three-way group-driven differential expression analysis among groups, employing the Kruskal-Wallis Kruskal and Wallis (1952), Wilcoxon rank sum test Wilcoxon (1945), and a reverse sequential feature-selection method Gelsema and Kanal (2014) that considers the combined effect of several genes. Alternatively, manual selection of reference genes is also available to the researcher. Once content normalization is performed, additional normalization options such as standardization or quantile normalization are available.

3.3 Step 3: Evaluation of normalization results

The normalization results are evaluated through computation of the interquartile range and graphical analysis using Relative Log Expression (RLE) plots Gandolfo and Speed (2018), which compare the raw data with the normalization results.

An example of RLE plots can be seen at the bottom right corner of Figure 1. It corresponds to the processing of the dataset GSE183071 available at the GEO database.

For more detailed information see the GUANIN User Guide at *https://github.com/julimontoto/guanin/blob/master/GUANIN_userguide_1.3.pdf* .

4 Results

The main objective of GUANIN is to provide a flexible and adaptable parametrization to allow the users to adjust the normalization process to the characteristics of their experiment. As experienced users of NanoString data normalization, we have implemented into GUANIN the next features to offer a better user experience and wide experiment compatibility:

- GUANIN can preprocess miRNA and RNAs experiments by default.
- Wide compatibility with different editions of RCC format, column names, and gene identifiers.
- Optional visualization of results for every step.
- Wide range of background choices, including brand new approaches relevant to specific experiments.
- Configurable and visual QC.

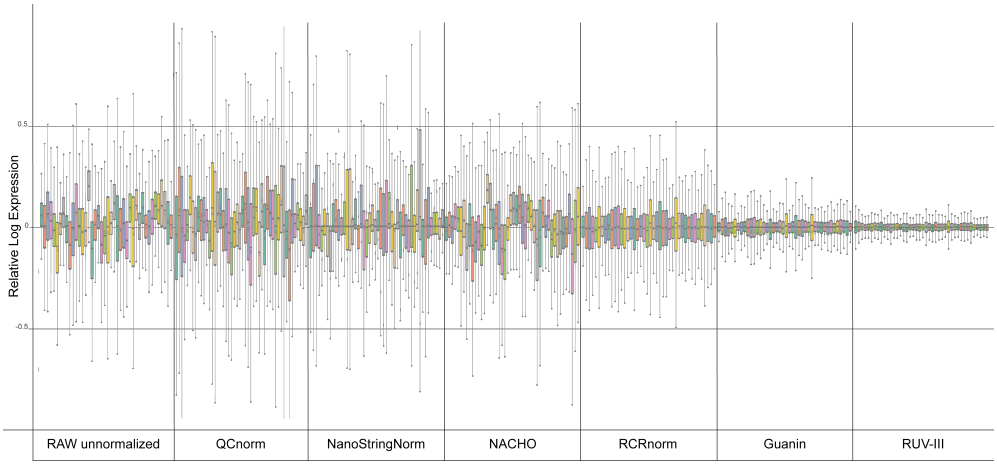

Figure 4: Comparison of RLE plots from normalized output data for the dataset GSE183071 and for several NanoString normalizing solutions

- Possibility to perform technical normalization after or before background correction.
- Adaptable content normalization, which includes algorithms to select, ponderate, and validate any number of endogenous genes.
- GUANIN incorporates a thorough evaluation of candidate reference genes, including the Kruskal-Wallis test, Wilcoxon test, and a machine learning reverse feature selection procedure to asses to assess the combined effect of multiple genes on the condition under study.
- PDF reports with main plots for QC and normalization results.
- In-built evaluation with RLE plots in the main window.

A comparison of usability with respect to other NanoString data normalization tools can be found in Figure 3.

In order to evaluate GUANIN, we have examined three studies, including one in-house dataset of a COVID-19 study (GSE183071) and two published datasets (GSE160208, GSE108395) that can assess several standard casuistic issues when analyzing NanoString data. All of them are available at the GEO database (*https://www.ncbi.nlm.nih.gov/geo/*). The User's Guide of GUANIN (*https://github.com/julimontoto/guanin/blob/master/GUANIN_userguide_1.3.pdf*) details how the normalization process of each of these dataset is performed and the results obtained.

While other software packages may be unable to address specific problems during data handling and normalization, GUANIN maintains good RLE plots and provides accurate normalized data for all the datasets. Figure 4 shows a comparison of the RLE plots for normalized output data for the dataset GSE183071 amongst main NanoString normalization tools. While RUV-III shows better RLE plots that could indicate a better normalization, it is less configurable, flexible to input data, adaptable to preprocessing problems, and user-friendly compared to GUANIN. Besides, it is important to note that centered and narrow RLE plots are not always indicative of a better normalization, as some relevant variability can be lost too. Additionally, GUANIN implements additional features like the geNorm selection algorithm, which is not possible to use with other approaches, making it a comprehensive tool for the normalization process.

5 Conclusions and future work

In a sequential interactive framework, it is necessary to properly address preprocessing step of every specific experiment properly. GUANIN provides the ability to conduct thorough analysis and adapt preprocessing to each experiment without an in-deep knowledge of any programming language, which is a useful advantage for health researchers.

Assuming NanoString pre-built-in housekeeping genes will work ideally as reference genes for an experiment is naive, especially when dealing with diverse tissues and metabolic processes. Therefore, including endogenous genes as candidate reference genes proves to be a great option. The geNorm intelligent evaluation and selection process implemented in GUANIN includes at least three endogenous genes in best 6-gene selection to be used as reference genes.

While no other tool provides as wide interactive parametrization as GUANIN does, NACHO can be useful for a smooth alternative visualization, and RUV-III offers excellent results in removing unwanted variation but it requires to include technical replicates, which are not typically available in most NanoString experimental designs or in public repository datasets. RUVSeq Risso et al. (2014) has been also probed to offer good normalization results on NanoString data Hafemeister and Satija (2019), although it was designed for RNA-Seq analysis.

Additionally to its flexibility, GUANIN generally exhibits better RLE plots than any previous NanoString normalizing tool, especially when groups are given.

Furthermore, GUANIN's results are particularly promising when the analysis can be refined; which is frequently the case on exploratory or confirmatory studies, main objective of NanoString experiments. It can address issues such as poor housekeeping performance, poor negative control, low general expression, and suboptimal experiment design. This is common, as most NanoString panels are preset with default housekeeping genes for a tissue or with a selection of endogenous genes that does not have to match specifically our experiment if panel is not custom (and even if it is custom these problems are usual). Because of this, poor QC is often encountered.

As for future work, although we introduced a regularized linear binomial regression model option for technical normalization, a single scaling factor does not effectively normalize both lowly and highly expressed genes Bhattacharya et al. (2021), thus we are working on improving the method and/or applying it to content normalization, as results are not as good as expected.

Finally, we think that GUANIN's excellent results, combined with its wide flexibility and easy-to-use interface, make it the best preprocessing tool for clinical scientists seeking a fast, reliable, and comprehensive method to preprocess their data and obtain visual reports of the results.

It can be also a useful tool for experienced scientists with programming experience, as it allows for an easy transition from RCCs to evaluated normalized data and provides intermediate data of the processes that can be easily accessed, facilitating the introduction of custom pipelines if desired.

Data Availability

GUANIN is open software distributed under the GPL v3 license. Source code, documentation and case studies are available at https://github.com/julimontoto/guanin.

Acknowledgements

CITIC is funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS).

Bibliography

A. Bhattacharya, A. M. Hamilton, H. Furberg, E. Pietzak, M. P. Purdue, M. A. Troester, K. A. Hoadley, and M. I. Love. An approach for normalization and quality control for NanoString RNA expression data. *Briefings in bioinformatics*, 22(3):bbaa163, 2021.

M. Canouil, G. A. Bouland, A. Bonnefond, P. Froguel, L. M. t. Hart, and R. C. Slieker. NACHO: an R package for quality control of NanoString nCounter data. *Bioinformatics*, 36(3):970–971, 2020.

R. Computing. What is pyqt. *https://www.riverbankcomputing.com/software/pyqt/*, 2023.

J. A. Gagnon-Bartsch and T. P. Speed. Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, 13(3):539–552, 2012.

L. C. Gandolfo and T. P. Speed. RLE plots: Visualizing unwanted variation in high dimensional data. *PloS one*, 13(2):e0191629, 2018.

G. K. Geiss, R. E. Bumgarner, B. Birditt, T. Dahl, N. Dowidar, D. L. Dunaway, H. P. Fell, S. Ferree, R. D. George, T. Grogan, et al. Direct multiplexed measurement of gene expression with color-coded probe pairs. *Nature biotechnology*, 26(3):317–325, 2008.

E. S. Gelsema and L. N. Kanal. *Pattern Recognition in Practice IV: multiple paradigms, comparative studies and hybrid systems*. Elsevier, 2014.

C. Hafemeister and R. Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome biology*, 20(1):296, 2019.

G. Jia, X. Wang, Q. Li, W. Lu, X. Tang, I. Wistuba, and Y. Xie. RCRnorm: an integrated system of random-coefficient hierarchical regression models for normalizing NanoString nCounter data. *The annals of applied statistics*, 13(3):1617, 2019.

W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

Y. Lin, K. Golovnina, Z.-X. Chen, H. N. Lee, Y. L. S. Negron, H. Sultana, B. Oliver, and S. T. Harbison. Comparison of normalization and differential expression analyses using RNA-Seq data from 726 individual Drosophila melanogaster. *BMC genomics*, 17(28), 2016.

R. Molania, J. A. Gagnon-Bartsch, A. Dobrovic, and T. P. Speed. A new normalization for Nanostring nCounter gene expression data. *Nucleic Acids Research*, 47(12):6073–6083, 2019.

D. Risso, J. Ngai, T. P. Speed, and S. Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature biotechnology*, 32(9):896–902, 2014.

J. Vandesompele, K. De Preter, F. Pattyn, B. Poppe, N. Van Roy, A. De Paepe, and F. Speleman. Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. *Genome biology*, 3(research0034.1), 2002.

D. Waggott, K. Chu, S. Yin, B. G. Wouters, F.-F. Liu, and P. C. Boutros. NanoStringNorm: an extensible R package for the pre-processing of NanoString mRNA and miRNA data. *Bioinformatics*, 28(11):1546–1548, 2012.

H. Wang, C. Horbinski, H. Wu, Y. Liu, S. Sheng, J. Liu, H. Weiss, A. J. Stromberg, and C. Wang. NanoStringDiff: a novel statistical method for differential expression analysis based on NanoString nCounter data. *Nucleic acids research*, 44(20):e151, 2016.

F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Z. Zeng, Y. Xiong, W. Guo, and H. Du. ERgene: Python library for screening endogenous reference genes. *Scientific Reports*, 10(18557), 2020.

S. Zhong. ctrlGene: assess the stability of candidate housekeeping genes. R Package, version 1.0.1, 2019.