CONSTANd : an efficient normalization method for relative quantification in small- and large-scale omics experiments in R BioConductor and Python

Reference:
Van Houtven Joris, Hooyberghs Jef, Laukens Kris, Valkenborg Dirk.- CONSTANd : an efficient normalization method for relative quantification in small- and large-scale omics experiments in R BioConductor and Python
Journal of proteome research - ISSN 1535-3893 - 20:4(2021), acs.jproteome.0c00977
Full text (Publisher's DOI): https://doi.org/10.1021/ACS.JPROTEOME.0C00977
To cite this reference: https://hdl.handle.net/10067/1762360151162165141

CONSTANd: an efficient normalization method for relative quantification in small and large scale omics experiments in R BioConductor and Python

Joris Van Houtven,[†,‡,¶] Jef Hooyberghs,[†,§] Kris Laukens,[‖,⊥] and Dirk Valkenborg[*,‡,¶]

†*Flemish Institute for Technological Research (VITO), Boeretang 200, B-2400 Mol, Belgium*

‡*Universiteit Hasselt, Data Science Institute (DSI), Interuniversity Institute for Biostatistics and Statistical Bioinformatics (I-BioStat), Agoralaan, Diepenbeek, BE 3590*

¶*Universiteit Antwerpen, Centre for Proteomics, Groenenborgerlaan 171, Antwerpen, BE 2020*

§*Universiteit Hasselt, Data Science Institute (DSI), Theoretical Physics, Agoralaan, Diepenbeek, BE 3590*

‖*Universiteit Antwerpen, Biomedical Informatics Network Antwerp (Biomina), Middelheimlaan 1, Antwerpen, BE 2020*

⊥*Universiteit Antwerpen, Adrem Data Lab, Department of Computer Sciences, Middelheimlaan 1, Antwerpen, BE 2020*

E-mail: dirk.valkenborg@uhasselt.be

Abstract

For differential expression studies in all omics disciplines, data normalization is a crucial step that is often subject to a balance between speed and effectiveness. To keep up with the data produced by high-throughput instruments, researchers require fast and easy-to-use, yet effective methods that fit into automated analysis pipelines. The CONSTANd normalization method meets these criteria, so we have made its source code available for R/BioConductor and Python. We briefly review the method and demonstrate how it can be used in different omics contexts, for experiments of any scale. Widespread adoption across omics disciplines would ease data integration in multi-omics experiments.

Keywords: normalization, transcriptomics, quantitative, proteomics, mass spectrometry, data-driven, multi-omics, workflow, quality control

Introduction

Due to the use of modern, high-throughput instruments in most omics disciplines, the data analysis has become one of the most labour-intensive steps, requiring a high level of expertise and effectively creating a bottlenck[1 2 3]. Typically, omics – and specifically in this paper: transcriptomics and proteomics – data analysis can be split into two parts: characterization and quantification. Although a variety of characterization software exists, this step is rather monolithic and usually highly automated. There is also no shortage of quantification software applications[4], although here the user typically has more freedom to choose the optimal steps involved.

Although the quantification workflow can differ significantly between omics disciplines, it always involves a normalization step – responsible for removing systematic errors (bias) due to experimental variation – which can make or break the analysis. In differential expression studies – where one is interested in differences between groups of biological samples with different phenotypes – it is sufficient to do a relative quantification between

the biological samples or conditions, as opposed to revealing absolute abundances. One can take either a simpler, purely data-driven normalization approach, or a more complex approach based on statistical models as f.i. proposed by Oberg et al.[5] and Hill et al.[6] et al. for proteomics. Although based on a very rigorous statistical framework, a model-based approach can be computationally intensive and in the proteomics case even become computationally infeasible as the number of features runs in the thousands[5]. On the other hand, data-driven methods are sometimes limited in efficacy and become unsuitable when combining data from multiple runs, despite their popularity and widespread use (e.g., quantile normalization). As such, choosing an appropriate normalization method involves a balance between speed and effectiveness, though often one resorts to data-driven approaches due to the ease of implementation – at least, if the method is publicly available. Fortunately, some methods like NOMAD[7] (only for specific proteomics experiments) and median sweeping[8] (generally applicable) seem to be both fast and effective, or at least strike an excellent balance. A method we developed, CONSTANd[9] seems to be on-par with normalization by median sweeping, DE-Seq2 (in transcriptomics[10]) and linear mixed models, but comes with few caveats or additional processing steps (e.g. no multiple stages of normalization, nor data transformations). It had already been implemented in QCQuan[11], an online workflow for proteomics, and is also available as a standalone version on the same website (`qcquan.net/constand`).

CONSTANd has been updated since its initial publication and to make the method fully integratable in researchers' in-house workflows, we have now made it available as an R BioConductor package[12] and for Python via GitHub[13]. In this technical note, we briefly review the updated methodology and when to use it, in a generalized context (i.e., not tied to a particular omics discipline). Then, we demonstrate how to use the R BioConductor package on both a small-scale proteomics data set from a single instrument run, as well as a larger transcriptomics data set spanning multiple instruments, labs, and wet lab protocols. To do this, we also review the recently introduced concept of 'identically processed

subsets' (IPSs)[10], again in a generalized context.

Materials & methods

IPS

Although CONSTANd normalization has only been shown to work on transcriptomics and proteomics data, it is conjectured to work on any type of quantitative omics data set as long as it is applied to each IPS separately. An identically processed subset (IPS) is a collection of quantification values (usually in a matrix) w.r.t. some omics features (e.g., genes or proteins) that belong to a subset of biological samples which have been identically processed in the wet lab and by the instrument, i.e., exposed to exactly the same (significant) sources of variability. An extreme example is multiplexing, where samples are labeled and then pooled for simultaneous processing. This guarantees that only the biological systematic effects (which we exclude from aforementioned sources) can be restricted to a particular part of the IPS' quantification matrix (e.g., all genes related to cell division in a sample of cancerous cells). All other remaining systematic effects in an IPS are due to experimental biases which affect only entire samples (columns; e.g., library size variations in transcriptomics) or features (rows; e.g. ionization efficiency in mass spectrometry-based omics).

CONSTANd v1.1

The CONSTANd method (updated since the initial publication of v1.0, see further) removes these remaining experimental biases within each IPS and simultaneously removes any bias affecting entire IPSs, by imposing two constraints on each IPS' $M \times N$ matrix of quantification values:

- the mean of all observed values in each row is 1;

- the mean of all observed values in each column is 1;

and missing values are ignored. These constraints are theoretically impossible to accomplish in a finite number of operations while keeping ratio's between quantification values meaningful. However, there exists an iterative proportional fitting procedure (IPFP)[14] to get a matrix arbitrarily close to satisfying both constraints while conserving the quantity $\frac{x_{r_1c_1}x_{r_2c_2}}{x_{r_2c_1}x_{r_1c_2}}$, where x_{ij} is the value at position i, j. The meaningfulness of this conserved quantity is not immediately obvious, but the RAS[15] implementation of the IPFP is actually very intuitive. It is sometimes called 'matrix raking' and is used in CONSTANd as follows:

1. Divide each of the M rows by its mean.

2. Divide each of the N columns by its mean.

3. Repeat until all such means are close to 1, within some given precision.

One could loosely state that the column operations more or less correspond to a sample size correction, and the row operations to a standardization. If the row and column operations are collected in diagonal matrices R and S, respectively, the normalized matrix K can be written in terms of the original matrix A as $K = RAS$.

Note how CONSTANd sets the average quantification value to 1 in each row and column, independently of the matrix size $M \times N$. This particular choice constitutes the update we have made to the algorithm since its inception (v1.0), when these targets used to be 1 and $\frac{M}{N}$, respectively, and average value $\frac{1}{N}$. However, the matrix size independence of (updated) version 1.1 allows comparisons between measurements of the same feature in a differently sized matrix (but with the same balance). For instance, if one conducted two instrument runs of condition A versus B, one can now compare the results from a 3vs3 run with those of a 4vs4 run (but not, e.g., with a 4vs2 run) without making further adjustments.

Properties

The procedure described above has a couple of noteworthy effects on the measurements, leading to proprties which we list here. First, as mentioned earlier, CONSTANd 'normalizes' the data, in the strict sense that it removes unwanted biases, while keeping values meaningfully comparable within the same feature and even across IPSs, but not across features. The latter is because secondly, the method entails a mild form of standardization: the observations in each feature are centered around the mean value 1, but this is done by a single multiplication (division) instead of substracting the mean and dividing by the standard deviation like in z-scores. Moreover we divide by the mean instead of the standard deviation, which somewhat resembles a variance stabilizing transformation[16]; dividing by a measure of the magnitude brings the variances 'on the same scale' and this 'captured variability' is stored in the row multipliers R, which can be of use in visualizations like the MA plot (see further). As for CONSTANd's effect on log_2-transformed fold changes: this can be interpreted roughly as an up or down shift, as shown in Figure S2. Lastly, there are two additional advantages to this 'standardization'. One is that the distribution of values is rendered more symmetrical (see Figure S3), which removes the need for a log-transformation. The other is that one can look at an isolated value in the normalized matrix and immediately judge whether it represents an up- (> 1) or down-regulated (< 1) feature measurement. Lastly, Figure S3 shows how the normalization effect can be roughly interpreted as a shift in fold changes.

Use cases

As with all data-driven methods, CONSTANd relies on the entirety of all measurements to act as an 'internal reference' and uses them to compute a central tendency. Therefore, the data to be normalized must adhere to the following assumptions, which may all be verified in an MA plot (e.g. Figure 3a) of the raw data:

1. Most features are not differentially expressed (e.g., housekeeping proteins). MA-plot: the observations form a single 'cloud' with a dense center and less dense edges. It is assumed that up-down shifts of the cloud are not due to biological causes and can safely be removed from the data.

2. The amount of downregulated features is roughly equal to the amount of upregulated ones. MA-plot: the cloud of observations exhibits a bilateral symmetry about some axis (which is usually completely horizontal, but it may be somewhat inclined).

3. The magnitude of any systematic bias is directly proportional to the intensity of the measurement. The axis of bilateral symmetry is a straight line (which may be inclined), i.e., the moving average of M-values form an (approximately) straight line. No banana-like lines allowed.

An example of such a 'good' MA plot can be found in Figure 3a, and a 'bad' one in Figure S1.

CONSTANd works on raw intensities (not log-transformed) and assume variability and systematic effects to work on a multiplicative scale (which justifies the multiplictive matrix raking approach). The authors believe that this is a widespread assumption and to the best of our knowledge there exists no method which makes a different assumption.

One may use CONSTANd to combine relative measurements from all kinds of different samples, as long as they can be split into IPSs which all have the same balance between the same biological conditions, although the size may vary. In principle, one could even combine a run with experimental design balance 2vs2vs2 samples of conditions A,B,C together with a run with balance 3vs3 samples of conditions A,C if one drops the B samples from the first run (and for this case also IPS) before normalization. To combine any number and type of instrument run quantification matrices:

1. Split each matrix into IPSs (often, a matrix corresponds to an IPS).

2. Make sure their experimental design balances are identical and drop (parts of) IPSs if necessary.

3. Normalize each IPS separately from the others.

4. Merge the resulting matrices (inner join by features).

After the use of CONSTANd, one may proceed with differential expression testing[17] as usual, calculating log fold changes of features by taking the $\log_2$ of a ratio of normalized intensity values. As CONSTANd is a data-driven method (not based on statistical models) there is no formal statistical estimation of 'effect sizes'. The fold changes are the intended outcome variables

Data sets

In our demonstrations, we use two publicly available data sets.

The 'Spike-in' data from PXD000001 is obtained in the `RforProteomics` vignette. It is a rather small (about 36×10^3 quantification values) TMT-labeled, single-run proteomics data set entailing 6 different conditions (only 1 sample each) consisting of a background and 4 spike-in proteins: "[...] (ENO) at 10:5:2.5:1:2.5:10, [...] (BSA) at 1:2.5:5:10:5:1, [...] (PHO) at 2:2:2:2:1:1 and [...] (CYT) at 1:1:1:1:1:2"[18]. All things considered, this data set is a single IPS – any systematic differences in the background (technical replicate) are unlikely to be of significance.

The 'ABRF' data can be obtained by downloading the supplementary file with counts from GSE48035 (GEO database). It is a large (about 1.2×10^6, excluding sites M,N and conditions C,D) transcriptomics data set that entails four biological conditions 'brain' (A) and 'tumor' (B), and then a 3:1 (C) and 1:3 (D) mixture of A and B. They have been measured at different sites (L,R,V,W,M,N), sometimes using two or more different protocols ('RiboDepletion', 'PolyA', and others), so there are multiple IPSs. In the demonstration, we will exclude mixed conditions C and D (not present at every site) and disregard site M

and N (used other wetlab procedures), to illustrate how to properly subset a rather complex data set.

Results

We have made CONSTANd available for `R4` as a `BioConductor` package named `CONSTANd`, and for `Python3` via github: `https://github.com/PDiracDelta/CONSTANd-py`. Here, we use `R` to demonstrate the use of CONSTANd, although a `Python3` script is also available in the Supporting Information. The scripts to load and clean the data can be found in the Supporting Information.

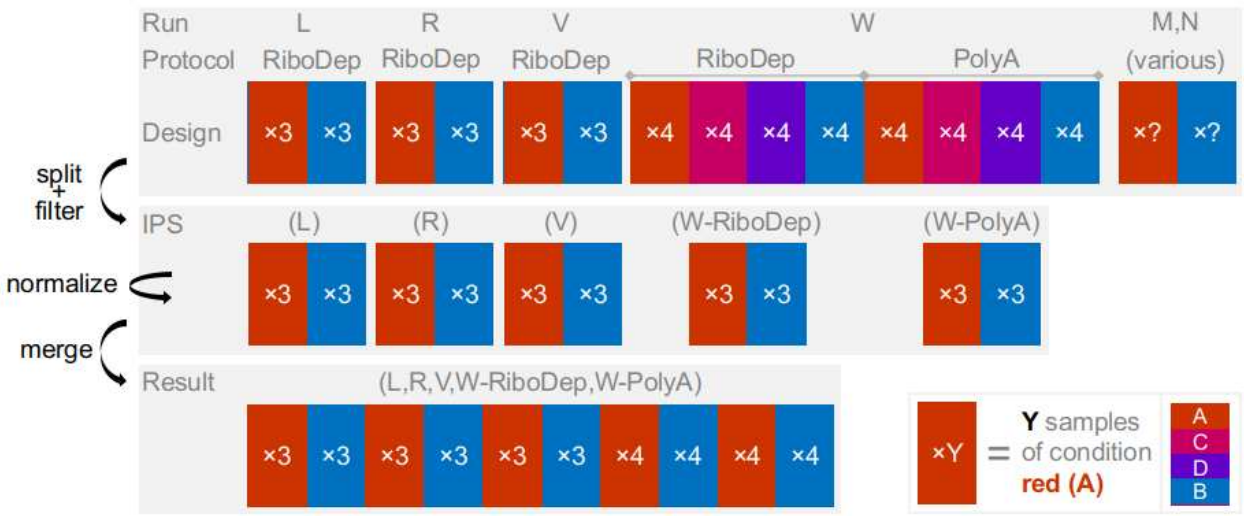


Figure 1: The complex ABRF data set is first split and filtered into IPSs. Samples that cannot form an IPS with a balance identical to that of other IPSs are discarded (filtered out). After normalizing each IPS individually, they may be merged into one large matrix of values, comparable within each feature (row). For the intricacies of how to do this in R: see the `ABRF.R` file in the Supporting Information.

A simple data set like the Spike-in can be cleaned and then normalized using `R` code similar to that in Figure 2a. A single, multiplexed LC-MS run in proteomics is usually its own IPS, so no data splitting is necessary. Since CONSTANd returns the `normalized_data`, as well as the attained precision and the vectors R and S (row and column multipliers), it is good practice to save its output as a separate object. With the help of PCA plots, we

a) Spike-in

```
x <- exprs(qnt)  # qnt is an MSnSet object
colnames(x) # TMT 6-plex reporter labels:
# [1] "126" "127" "128" "129" "130" "131"
# PSM-level normalization
result <- CONSTANd(x)
exprs(qnt) <- result$normalized_data
```

b) ABRF

```
## split into IPSs
colnames(colData) # [1] "sample" "protocol" "site" "tissue"
dim(data) # all in one big matrix: [1] 30595    62
data.ips <- remove_irrelevant_samples(data, colData)
dim(data) # dropped 28 samples: [1] 30595    34
data.ips <- split_by_site(data, colData)
data.ips <- split_site_W_by_protocol(data.ips, colData)
# filter out low count cases per site, too
data.ips <- lapply(data.ips, function(d)
  d[rowMedians(as.matrix(d)) > 0,])

## PSM-level normalization
result.ips <- lapply(data.ips, CONSTANd)
data.ips.norm <- lapply(result.ips, function(r)
  r$normalized_data)
## re-combine data
data.norm <- merge_and_restore_column_order(data.ips.norm)
```

c) MA plot trick

```
# get CONSTANd row multipliers of features in merge result
R <- result.ips$L$R[rownames(data.norm)]
L.brain <- rowMeans_of_site_L_Brain_samples(data, colData)
L.tumor <- rowMeans_of_site_L_Tumor_samples(data, colData)
# regular MA plot; looks bad as M and A are correlated
MAplot(rowMeans(data[,L.brain]), rowMeans(data[,L.tumor]))
# trick: use R vector from CONSTANd (K = R*A*S) to re-scale
plot((log2(L.brain/R)+log2(L.tumor/R))/2,
     log2(L.brain/L.tumor))
```

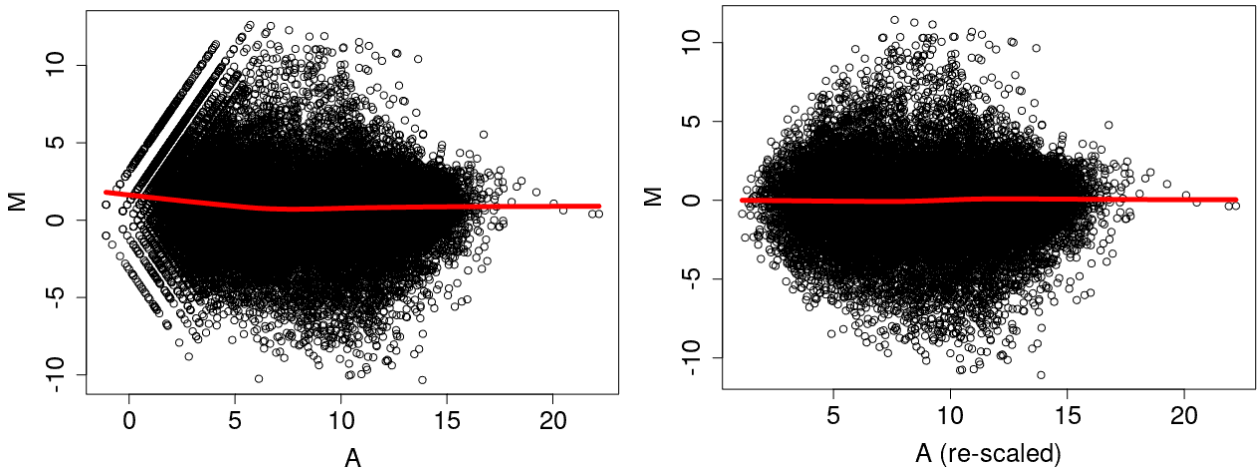
Figure 2: a) A single IPS can be easily extracted from, for instance, an MSnSet object and immediately normalized. b) A more complex experimental design should be split into multiple IPSs, which may be re-combined after normalization. c) After normalization, one can still produce a meaningful MA-like plot by exploiting the feature magnitude information in the vector R which contains the row multipliers used in CONSTANd. Note that some features have been removed during the merge of IPSs after normalization.

can see in Figure S5 that the normalization has maximized the PC1 and PC2 distances between dissimilar samples.

A more complex experimental design should be split into multiple IPSs, which may be re-combined after normalization. In the ABRF transcriptomics data, we are interested in the AvsB comparison. Figure 1 and Figure 2b show how we first drop all samples from sites M and N, because some of the samples there have been prepared with protocols other than RiboDepletion and PolyA, and the remaining samples cannot constitute a balanced IPS anymore. We also drop samples from conditions C,D (not relevant for us) from site W. We do so before the normalization because their presence changes the design balance to AvsBvsCvsD and thus would in principle affect the normalization outcome (although this is an exceptional case, see Supporting Information). Then, we split the samples from site W into two IPSs, one for each protocol. The skilled person can recognize the splitting of the data into IPSs as an equivalent to adding fixed 'batch' effects to a linear model. After independent normalization, the IPSs are merged back together (here, by inner join on gene name) and may be compared. The PCA plots in Figure 4 show how the remaining

variability in the normalized data is dominated by the biological conditions, suggesting the procedure was successful. A similar conclusion can be drawn from the heatmaps of correlations between samples in Figure S4, which suggests CONSTANd greatly improves separation between the conditions.

Unfortunately, a regular MA plot of normalized data can look unusual (see Figure S6) as M and A values are correlated after setting all mean row marginals to 1. This can be remedied by using information on the feature magnitudes stored in the R vector from the CONSTANd output to construct a pseudo-MA plot. One could either adapt the A values by dividing by R (as in Figure 3b), or by replacing them with the ordering in R (as in Figure S7).



(a) Regular MA plot of raw data, which seems to adhere to the three assumptions.

(b) Pseudo-MA plot of normalized data.

Figure 3: MA and pseudo-MA plot of Tumor versus Brain from site L of the ABRF study. After normalization, there seems to be less variability (5.14) in the M values than before (5.87), and the artifacts (diagonal lines) in the low count region have been abolished.

Discussion

We have shown that using CONSTANd can be as simple as a one- or two-line coding endeavour, although in more complex study designs the data sets have to be partitioned and afterwards re-combined according to the experimental setup. The concept of the

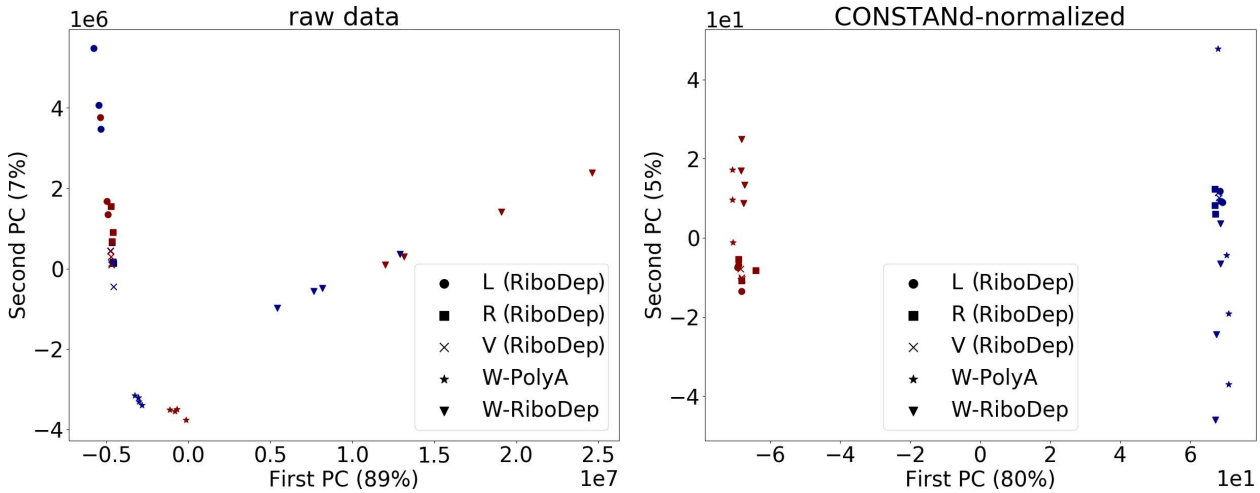


Figure 4: While the PCA only partially and only weakly groups raw (left) data samples according to all influential factors, CONSTANd (right) succeeds at separating the samples according to their biological condition (red: brain (A), blue: tumor (B)) on PC1.

IPSs provides an intuitive way of doing this: if two samples are in the same partition, they should have been identically processed. In principle, no additional steps are required to visualize or further use the normalized data, although we have demonstrated a trick to improve the interpretative potential of the MA plot, which is otherwise limited. Scripts that demonstrate the use of CONSTANd in both R and Python are available in the Supporting Information.Although CONSTANd has only been shown to work on transcriptomics and proteomics data, it is conjectured to work on any type of quantitative data sets as long as it adheres to the usual assumptions for data-driven methods, mentioned in the section on use cases. By making the method available not only as an online tool but also as an open source project, the authors hope to ease adoption of the method in researchers' customized workflows. The method should be widely tested in other omics disciplines, as it is designed for both horizontal and vertical integrative omics[19]. This versatile applicability can increase the uniformity and comparability of analytic pipelines in multi-omics studies, leading to more comparable measures and thus better integration of results.

Supporting Information

The following supporting information is available free of charge at ACS website `http://pubs.acs.org`

1. Why it would be acceptable to compare an AvsBvsCvsD IPS with an AvsB IPS in the ABRF study.
2. How to use CONSTANd in Python.
3. Example of MA plot that violates all assumptions Figure S1.
4. Correlation plot of $\log_2$ fold changes Figure S2.
5. Boxplots before and after CONSTANd normalization Figure S3.
6. Correlation heatmaps of ABRF samples before and after CONSTANd normalization Figure S4.
7. PCA plots of Spike-in data, before and after normalization Figure S5.
8. Regular MA plot of CONSTANd-normalized ABRF data Figure S6.
9. Rank-based MA plot of raw and normalized ABRF data Figure S7.
10. Text files with Python and R-code for processing data sets (`ABRF.py.txt`, `ABRF.R.txt`, `Spikein.R.txt`): `scripts.zip`.

References

(1) Wetie, A. G. N.; Shipp, D. A.; Darie, C. C. *Advancements of Mass Spectrometry in Biomedical Research*; Springer, 2014; pp 581–593.

(2) Ltd., C. The genomic analysis bottleneck - is it limiting your lab's ability to scale? 2020; https://blog.congenica.com/the-genomic-analysis-bottleneck-is-it-limiting-your-labs-ability-to-scale.

(3) Laurenson, S. Scream If You Wanna Go FASTA: Breaking the Bottleneck in Genomic Data Analysis. 2019; https://www.technologynetworks.com/informatics/articles/scream-if-you-wanna-go-fasta-breaking-the-bottleneck-in-genomic-data-analysis-3283

(4) O'Rourke, M. B.; Town, S. E.; Dalla, P. V.; Bicknell, F.; Koh Belic, N.; Violi, J. P.; Steele, J. R.; Padula, M. P. What is Normalization? The Strategies Employed in Top-Down and Bottom-Up Proteome Analysis Workflows. *Proteomes* **2019**, *7*, 29.

(5) Oberg, A. L.; Mahoney, D. W.; Eckel-Passow, J. E.; Malone, C. J.; Wolfinger, R. D.; Hill, E. G.; Cooper, L. T.; Onuma, O. K.; Spiro, C.; Therneau, T. M.; et al., Statistical analysis of relative labeled mass spectrometry data from complex samples using ANOVA. *Journal of Proteome Research* **2008**, *7*, 225–233.

(6) Hill, E. G.; Schwacke, J. H.; Comte-Walters, S.; Slate, E. H.; Oberg, A. L.; Eckel-Passow, J. E.; Therneau, T. M.; Schey, K. L. A statistical model for iTRAQ data analysis. *Journal of Proteome Research* **2008**, *7*, 3091–3101.

(7) Murie, C.; Sandri, B.; Sandberg, A.-S.; Griffin, T. J.; Lehtiö, J.; Wendt, C.; Larsson, O. Normalization of mass spectrometry data (NOMAD). *Advances in biological regulation* **2018**, *67*, 128–133.

(8) Herbrich, S. M.; Cole, R. N.; West Jr, K. P.; Schulze, K.; Yager, J. D.; Groopman, J. D.; Christian, P.; Wu, L.; O'Meally, R. N.; May, D. H. et al. Statistical inference from multiple iTRAQ experiments without using common reference standards. *Journal of Proteome Research* **2013**, *12*, 594–604.

(9) Maes, E.; Hadiwikarta, W. W.; Mertens, I.; Baggerman, G.; Hooyberghs, J.; Valkenborg, D. CONSTANd : A Normalization Method for Isobaric Labeled Spectra by Constrained Optimization. *Molecular & Cellular Proteomics* **2016**, *15*, 2779–2790.

(10) Van Houtven, J.; Cuypers, B.; Meysman, P.; Hooyberghs, J.; Laukens, K.; Valkenborg, D.

(11) Van Houtven, J.; Agten, A.; Boonen, K.; Baggerman, G.; Hooyberghs, J.; Laukens, K.; Valkenborg, D. Qcquan: A web tool for the automated assessment of protein expression and data quality of labeled mass spectrometry experiments. *Journal of proteome research* **2019**, *18*, 2221–2227.

(12) Van Houtven, J.; Valkenborg, D. CONSTANd. BioConductor package, 2021; `https://doi.org/doi:10.18129/B9.bioc.CONSTANd`.

(13) Van Houtven, J. CONSTANd for Python. online, 2020; `https://github.com/PDiracDelta/CONSTANd-py`.

(14) Deming, W. E.; Stephan, F. F. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics* **1940**, *11*, 427–444.

(15) Stone, R.; Champernowne, D. G.; Meade, J. E. The precision of national income estimates. *The Review of Economic Studies* **1942**, *9*, 111–125.

(16) Durbin, B. P.; Hardin, J. S.; Hawkins, D. M.; Rocke, D. M. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics* **2002**, *18*, S105–S110.

(17) Dunkler, D.; Sánchez-Cabo, F.; Heinze, G. *Bioinformatics for Omics Data*; Springer, 2011; pp 113–131.

(18) Gatto, L.; Christoforou, A. Using R and Bioconductor for proteomics data analysis. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* **2014**, *1844*, 42–51.

(19) Wu, C.; Zhou, F.; Ren, J.; Li, X.; Jiang, Y.; Ma, S. A selective review of multi-level omics data integration using variable selection. *High-throughput* **2019**, *8*, 4.

For TOC Only