Ingrid M. Lönnstedt* and Sven Nelander*

FC1000: normalized gene expression changes of systematically perturbed human cells

https://doi.org/10.1515/sagmb-2016-0072

Abstract: The systematic study of transcriptional responses to genetic and chemical perturbations in human cells is still in its early stages. The largest available dataset to date is the newly released L1000 compendium. With its 1.3 million gene expression profiles of treated human cells it offers many opportunities for biomedical data mining, but also data normalization challenges of new dimensions. We developed a novel and practical approach to obtain accurate estimates of fold change response profiles from L1000, based on the RUV (Remove Unwanted Variation) statistical framework. Extending RUV to a big data setting, we propose an estimation procedure, in which an underlying RUV model is tuned by feedback through dataset specific statistical measures, reflecting *p*-value distributions and internal gene knockdown controls. Applying these metrics – termed evaluation endpoints – to disjoint data splits and integrating the results to select an optimal normalization, the procedure reduces bias and noise in the L1000 data, which in turn broadens the potential of this resource for pharmacological and functional genomic analyses. Our pipeline and normalization results are distributed as an R package (nelanderlab.org/FC1000.html).

Keywords: gene expression; normalization; *p*-value inflation; remove unwanted variation.

1 Introduction

The systematic exploration of transcriptional responses in living cells is an increasingly important tool to characterize bioactive compounds. Presently, the largest body of data available, L1000, is generated as part of the NIH-supported Library of Integrated Network Based Cellular Signatures (LINCS) project. The publicly available L1000 database currently includes 1.3 million gene transcript expression profiles, resulting from treatment effects of drug like compounds, gene knockdowns and other treatments in 77 cultured human cancer and noncancerous cell lines. The gene expression profiles are derived from Luminex bead arrays (Peck et al., 2006) limited to 978 carefully selected gene transcripts, which are stated to be minimally redundant and widely expressed in different cellular contexts. Conceptually, an expression experiment this scope offers enormous possibilities to explore how different cell types respond to various interventions. Increasing with the size of data is, however, also the amount of bias and unwanted variation needing attention. The focus of the current paper is to present a strategy to remove unwanted variation in the estimates of expression changes in big datasets in general, as well as to provide normalized fold change estimates from the L1000 data.

1.1 Structure of the L1000 dataset

The L1000 gene expression data is the result of an extensive set of experiments in which the 77 different cell lines have been treated with different molecular perturbations in 384 well plates. Following perturbation, each well is used to generate one Luminex expression profile (*array*) with expression levels of each of 978 transcripts. The perturbations used include 19,013 different small molecular compounds (*drugs*), 4308 unique

***Corresponding authors: Ingrid M. Lönnstedt,** Department of Immunology, Genetics and Pathology, Uppsala University, 75185 Uppsala, Sweden; Science for Life Laboratory, S-751 85 Uppsala, Sweden; and Bioinformatics Division, Walter and Eliza Hall Institute, Melbourne, Victoria 3052, Australia, e-mail: ingrid.lonnstedt@gmail.com; and **Sven Nelander,** Department of Immunology, Genetics and Pathology, Uppsala University, 75185 Uppsala, Sweden; and Science for Life Laboratory, S-751 85 Uppsala, Sweden, e-mail: sven.nelander@igp.uu.se

genes studied by *shRNA* knockdowns, 3097 unique genes perturbed by Open Reading Frame (*ORF*) based overexpression, and a limited set of <50 proteins such as growth factors. Furthermore, the experimental design involves multiple doses and time-points of many compounds. Different cell lines, perturbations, doses and time-points appear at very different frequencies. For instance, the most studied cell line (VCAP) has 187,488 experiments, whereas the two least studied cell lines have fewer than 100 experiments. Furthermore, the use of technical replicates varies. The most replicated experiment (VCAP cells treated by Vorinostat at 10 μM at 72 h) has been analyzed 164 times, whereas the majority of shRNAs and drugs have only 4–6 replicates at any given dose or time-point. Technical replicates are evenly distributed over several 384 well plates, and each plate holds many different perturbations. Since 24 h was the best represented time-point, we focused this normalization study on experiments performed 24 h after perturbation. The software and methodology should, however, apply to any time-points(s).

The data is distributed to the community at four different levels of processing. Here, we concentrate on the Q2NORM format of 978 transcript gene expression profiles which have been deconvoluted from the Luminex beads and normalized using invariant set scaling (Pelz et al., 2008) followed by quantile normalization (Bolstad et al., 2003). This is the data version in which it is straightforward to access all the 978 transcripts for each of the 1.3 million experiments, making it convenient for users to base analyses freely on for example all the replicate experiments of a perturbation, although these originate from different plates. Q2NORM has gone through a normalization step already, but it is the format of this data version rather than it's preprocessing which motivates us to use it. Our procedure could just as well have been applied before any normalization. We think of the L1000 Q2NORM data as a matrix $Y(m \times n)$ holding m arrays, or experiments, each of $n = 978$ gene transcript expression levels on a log-2 scale. We separately process the three partitions of Y: shRNA data, drug data and ORF data, and do not investigate the <50 protein perturbations in L1000.

The preprocessing description of the Q2NORM data suggests the gene expression profiles are ready to explore, but as we will demonstrate, they suffer from severe bias. In this project, we aim to reduce this bias in order to obtain accurate estimates of *fold change profiles* from the Q2NORM data, in particular with shRNA perturbations. The fold change profile of cell type i under perturbation j is the vector of true, unknown fold changes,

$$\boldsymbol{a_{ij}} = \{\log_2(e_{ijg}/e_{ibg}), g = 1, \ldots, 978\} \quad (1)$$

over the gene transcripts g, where e_{ijg} refers to the expression level of transcript g after applying the active (i.e. a drug, shRNA or ORF) perturbation j in cell line i, and e_{ibg} to the expression level of transcript g after applying a relevant *baseline* perturbation in the same cell line i: The L1000 data includes baseline arrays, assayed with an empty shRNA vector (for shRNA data), Green Fluorescent Protein (GFP, for ORF data) or dimethyl sulphoxide (DMSO, for drug data). In the remainder of this paper we focus on the cell types for which both active and baseline perturbations at 24 h are available (16 cell types with a total of 688,274 arrays for shRNA, 10 cell types with 127,522 arrays for ORF and 24 cell types with 806,083 arrays for drug data, Appendix Tables A1–A3).

A particular design feature of shRNA and ORF data, is that for each cell type, several of the perturbed genes are also present among the 978 gene transcripts. That means their true (expected) direction of regulation is known, a fact that we will use for evaluation of normalization methods.

The resulting experimental design of the shRNA data at our hands is summarized in Appendix Table A4. The number of replicates of each perturbation within each cell type differ. For example, we have 12,359 gene expression arrays of NPC cells, distributed across 36 different 384 well plates. Eight hundred seventy-three of these arrays are replicate baseline arrays, with 21–27 of them on each of the 36 plates. The remaining arrays represent 1075 distinct knockdowns, each replicated in total 3–73 times (mostly 9–12 times) evenly distributed across 2–8 plates. Appendix Table A5 shows the exact numbers of replicates for each of the perturbations in subset 2 of the NPC cell. A smaller example is the SHSY5Y cell, for which we have 1055 shRNA perturbed arrays from 3 different 384 well plates. Sixty-six of the arrays are replicate baseline experiments (22 on each plate),

and 126 distinct knockdown experiments are replicated in 3–9 arrays each (most of them have 9 replicates, 3 on each plate, Appendix Table A6).

1.2 Demonstration of bias

In this section we demonstrate the typical structure of bias present in fold change profiles as estimated directly from Q2NORM data, or following only naïve normalization attempts of Q2NORM data. We estimate each fold change profile $\{\hat{\boldsymbol{a}}_{ij}\}$ of cell i by the average log-2 expressions across all the replicate arrays of perturbation j minus the average log-2 expressions across all the replicate baseline arrays. That alone is a great efficiency advantage compared to just comparing single gene expression profiles. (The latter is currently the case in extant L1000 analysis tools, which are based on viewing each profile as an 'instance' which can be searched in a data-base like fashion at http://apps.lincscloud.org/.)

We first estimate fold change profiles based on data as provided (Q2NORM). We organize the fold change profiles as columns of a matrix A: Rows of A are the 978 readout genes and columns are the different knockdown perturbations. The Figure 1 heatmap of A shows a subset of shRNA perturbations of NPC cells, and gives a clear indication of severe bias in the fold change profiles estimated: Firstly, the heatmap contains vertical 'stripes', suggesting that a majority of the applied shRNAs globally suppress or activate all the measured 978 transcripts (Figure 1A,B). While global regulators of transcription have indeed been suggested, particularly MYC (Kress et al., 2015), the magnitude and number of the stripes clearly suggests technical bias as the more likely explanation. Secondly, we found horizontal stripes that would suggest that some of the measured transcripts respond the same to all the applied perturbations. Again, while not impossible in principle, we interpreted this as a clear indication of bias.

Attempting to reduce the bias, we explore a set of standard normalization methods. Inspired by the clear plate differences in Figure 1D, we applied ~~plate median normalization~~ (Figure 1C), but that does not seem to reduce bias much. We also fruitlessly attempted ~~quantile normalization with respect to plates~~, and estimating the fold changes using mixed models with a random factor of plate. We applied the established ComBat normalization method (Leek et al., 2012), in order to see if the batch effects of plates, and hence the visual bias, could be reduced, with some but not a great effect (Figure 1E). A recent paper, which adjusts L1000 data for spatial bias according to the well's location on plates (Lachmann et al., 2016), removed much of the visual bias in a few small cell lines with replicate arrays only distributed across a handful of plates, but failed our purposes with the vast majority of cells (Figure 1G). The lack of success with these existing methods motivated the development of a more specified normalization system. Figure 1F,H display our RUV optimal λ (Lambda) output which we are yet to describe. A useful visual evaluation of a normalization method, in addition to the absence of vertical and horizontal stripes, is that we genes knocked down to be down-regulated. On the zoom-ins of Figure 1B,C,E,G and H, we expect to see this as a green diagonal line. We see that the green diagonal becomes more apparent after the RUV normalization.

1.3 RUV

RUV (Remove Unwanted Variation) is a set of methods to reduce bias and variance in high dimension data by decomposition of data into signal, bias and noise (Gagnon-Bartsch and Speed, 2012; Gagnon-Bartsch et al., 2013; Jacob et al., 2015). RUV models are designed to find and correct for bias from unknown sources, which are always present in large gene expression datasets. In L1000, for instance, factors such as screening plates and bead arrays are known sources of unwanted variation, whereas there are potentially others such as cell passage, drug batch, equipment units, personal involvement etc. that are likely to influence transcript expression levels as well. As demonstrated in the previous section, naïve normalization methods fail of to reduce bias in L1000 data with respect to fold change profile estimation. This motivates the examination of RUV performed in this paper.

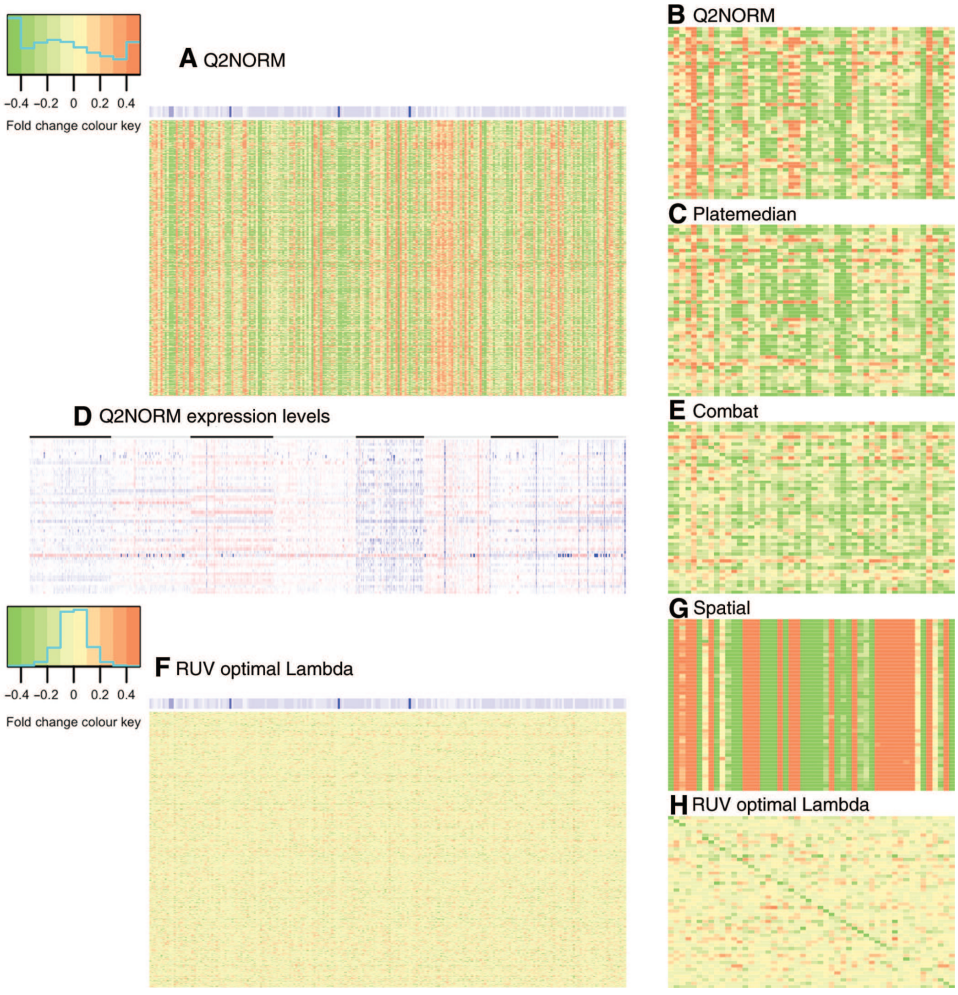

Figure 1: L1000 fold change estimates crucially depend on the normalization method.
(A) Heatmap showing limma-obtained fold change profiles, using the distributed version Q2NORM of L1000 data. Rows are the 978 assay readout genes, columns are shRNA knockdowns (one column for each unique gene target). Note the presence of vertical and horizontal stripes, strongly suggesting bias in the data. (B) Zoom-in of the upper left corner of a fold change matrix derived from Q2NORM, (C) after plate median, (E) Combat, or (G) Spatial (H) or RUV normalization. We have matched target and readout genes, meaning that we expect a diagonal of negative values (since we expect that knockdown of gene 1 leads to suppression of gene 1, and so on for gene 2, 3 etc). This diagonal, which is an important internal control, is more clearly seen in RUV normalized data. (F) Full heatmap of RUV (optimal λ see main text) normalized fold change profiles, of which (H) is a zoom-in. (D) Expression levels (blue to red) of 50 random gene transcripts (rows) across the arrays (columns) of 8 plates (black/grey bars) indicate systematic differences between plates. Representative data for one of the shRNA data cell lines (Neural Progenitor Cells, NPC, subset 2) shown, with identical colour scales for all panels except (D). Horizontal bars above (A) and (F) shows the number of replicate arrays of the shRNA knockdown incorporated into the fold change estimates of the column (white to blue scale is linear from 3 to 67).

Applied to L1000 gene expression data, RUV is based on representing data by

$$Y = X\beta + W\alpha + \epsilon, \tag{2}$$

where $Y(m \times n)$ are the gene expression levels of m arrays and $n = 978$ gene transcripts on the log-2 scale. The first term on the right side, $X\beta(m \times d, d \times n)$ is the linear term with X carrying known effects of interest, and our aim is to estimate β. We recognize that $\boldsymbol{a_{ij}}$ (equation 1) may be estimated by one row of β, and that the $\{\hat{\boldsymbol{a}}_{\boldsymbol{ij}}\}$ described in the previous section are exactly the least squares estimates of equation 2 which we would get were the second term omitted. This second term, $W\alpha$ $(m \times k, k \times n)$ is a similar linear term of systematic

noise with unknown dimension k, but W is unobserved. The matrix ϵ ($m \times n$) is random noise assumed Gaussian with the same variance for measurements on the same gene, but possibly different variances for different genes. The estimation of W is based on a *negative control* gene set c, for which it is assumed that $\beta = \mathbf{0}$. W may be estimated directly from $Y_c = W\alpha_c + \epsilon_c$ (c indicating the columns of negative controls) by factor analysis or by different methods exploiting the same idea. The parameters α and β are estimated by regressing Y onto X and $\hat{W}$.

There are different versions (algorithms) of RUV corresponding to different ways of estimating W, α and k. The performances of the algorithms differ between datasets. In this study we explore **RUV4**, **RUVinv**, **RUVIII** and **replicateRUV**. RUV4 and RUVinv are fully described in (Gagnon-Bartsch et al., 2013), while replicate RUV is described in (Jacob et al., 2015). ReplicateRUV and its refinement RUVIII both use replicate arrays to estimate α, and can be used when X is not known and we seek a normalized version of the dataset the same format as the original one.

Given a specific RUV algorithm, the method is far from instantly applied, but needs to be customized for the estimation problem at hand. All RUV algorithms rely on *negative controls*. They are gene transcripts specifically selected from the expression arrays so that on average across the transcripts, no variation in expression level is expected biologically across arrays, and hence systematic variation found among these transcripts are used to estimate bias.

The use of RUV is driven through biological and statistical evaluation of the output of different RUV settings. In practice, there are three parameters which must be optimized: the RUV method, the negative control set used, and the value of the parameter k where applicable. This optimization is a challenging problem with ordinary expression data sets and even more so with L1000, which is both extremely large, rich in systematic errors, and has a complex experimental design. The measures of evaluation of different RUV settings must be designed from each specific study context, and the derivation of such measures for L1000 fold change estimation is a major contribution of the current paper.

After the above description of L1000 data, it's normalization challenges with respect to fold change profile estimation, and the introduction to the RUV normalization framework, we now proceed to describe this project.

2 Strategy overview

In this report, we systematically analyze the crucial impact of RUV and alternative (plate median, ComBat: Johnson and Rabinovic, 2007; Spatial: Lachmann et al., 2016) normalization methods for L1000. The key goal of the analysis is to obtain accurate estimates of transcript fold changes following treatment by gene knockdowns (shRNA), drugs or over-expression (ORF) in each of the involved cell lines. A central item is the evaluation of different normalization methods and RUV settings. Given that the exact true fold change profiles of L1000 are unknown, it is not possible to base an evaluation on a golden standard reference. It is, however, quite possible to use internal controls and statistical criteria to assess the quality of bias removal and fold change estimation. We therefore suggest a set of 7 evaluation criteria (the endpoints unifKS, λ, Q3P, AdistKS, slopeHoriz, slopeVerti and MAD), described below. In the next step, we run RUV with different settings separately on subsets of shRNA data, and select the *optimal RUV* settings with respect to each criterion/endpoint. The analysis highlights that the endpoints prioritize different features in data, and therefore tends to select slightly different settings. As a head-to-head comparison of the optimal RUV outputs of the different evaluation endpoints, we measure how similar the fold changes of the endpoints' optimal RUV outputs are across all the cell types. Based on the assembled results, we suggest that good normalization performance is obtained by a particular version of RUV, RUV4, using p-value inflation (λ) as the recommended endpoint. This normalization, which differs substantially from existing normalizations but meets rigorous evaluation standards, is therefore the RUV settings we generally recommend for normalization of L1000 data. While the analysis focuses on the shRNA portion of L1000, which encompasses 400,000 arrays and is particularly well suited for evaluation because of the internal knockdown controls, the benefit of our

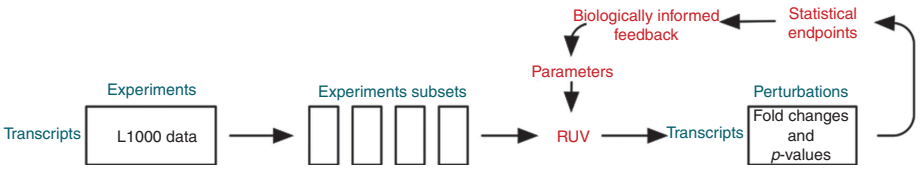

Figure 2: FC1000 flowchart.
The L1000 data matrix is split into experiment subsets, each of a size which can be handled with a standard computational capabilitiy. RUV is applied with ∼100 different settings to each subset, to give estimated fold changes and *p*-values. Our 7 statistical endpoints are evaluated for each RUV output. The 7 endpoint specific optimal RUV outputs of the complete database are queried for biologically informed feedback through between cell correlations of fold change estimates. The winning endpoint, together with the settings (parameters) which most often gives the optimal RUV output with respect to that endpoint, provide the RUV strategy and settings we generally recommend for estimation of normalized FC1000 fold changes.

normalization is analyzed and confirmed for the gene overexpression (Open Reading Frame, ORF) and drug parts of L1000 as well. Figure 2 summarizes the FC1000 normalization strategy as a flowchart.

3 Data preparation, normalization runs and fold change estimation

The shRNA data is the main dataset of this project, and have driven the development of normalization strategies. Therefore, methods are explained in terms of shRNA, but ORF and drug data were prepared similarly.

3.1 Division of data into subsets

Since it is not feasible to run RUV for the entire L1000, we processed data in subsets. Natural subsets are the cell types that have been perturbed, but most cell type subsets must be split even further for the normalization methods to come through on our cluster core (we used a cluster with 208 16-core nodes, each with 128GB RAM). Cell types with more than 2500 active perturbations where divided so that each subset contained all the baseline arrays of the cell type plus all the arrays of each of $d \approx 200$ distinct active perturbations. This algorithm resulted in 181 subsets of shRNA data, 385 subsets of drug data, and 101 subsets of ORF data. Next, we used cluster computing to systematically process each subset. Hence, we analyze each subset independently, although all subsets of the same one cell type include the same baseline replicate arrays.

3.2 RUV settings in normalization runs

Each L1000 data subset was assessed with different choices of RUV algorithms (RUV4, RUVinv, replicateRUV and RUVIII), negative control gene sets and parameter k values (see RUV introduction), each assessment which we refer to as a normalization *run*.

A particular challenge, which to our knowledge has not been thoroughly assessed with RUV earlier, is to find a negative control set rich enough to capture the bias structures in data although the arrays include only 978 gene transcripts, out of which most are expected to have some true biological and not just noise variation. While Gagnon-Bartsch and Speed, 2012 has a thorough discussion about different types of negative control gene sets, our efforts came down to simply comparing RUV run outputs based on each of the following sets of negative controls c: housekeeping genes (**HK**, the 54 gene transcripts from Eisenberg and Levanon 2003 present on the L1000 array), genes stable across cancer cells (**CCLE**, 476 genes with low gene expression variance across all cells in the Cancer Cell Line Encyclopedia, Barretina et al., 2012), the transcripts in the union of HK and CCLE for which the corresponding genes were not knocked out or overexpressed in the particular data subset (**HKCCLE**, this set varies between different data subsets), transcripts with low

variance in the data subset (**Empirical**, 100 transcripts selected across the range of expression levels as in Freytag et al., 2015) and all the 978 transcripts (**All978**, this may be useful in this case of a small array where truly stable genes have been deliberately removed, recalling that we look for average, or common, behaviours of negative controls).

Given an RUV algorithm and negative control set, we ran RUV with the parameter values k $\in$ {5, 10, 20, . . ., 90, 100, 125, 150}, under the restriction k $< d - 10$ (d the number of active perturbations in the data subset). RUV4 and RUVinv are known to be relatively insensitive to the number of unwanted factors k in the model, as long as the number n_c of negative controls is large enough, while RUVinv estimates the gene-specific variance with an "inverse method" and does not need k to be estimated. RUVIII can be run with k or without specifying k. Overall, this resulted in at most 205 runs per subset. For each subset, around 100–170 of the RUV runs successfully gave output fold change estimates.

For all RUV normalization runs, mean centered gene expression levels across subset arrays were used.

3.3 A note on fold change profile estimation

For each subset, RUV4 or RUVinv produces a matrix of fold change estimates $A = \{\hat{\boldsymbol{a}}_{i1}, \hat{\boldsymbol{a}}_{i2}, \ldots, \hat{\boldsymbol{a}}_{id}\} = \hat{\beta}'$ ($978 \times d$), and corresponding p-values P ($978 \times d$) to assess the alternative hypothesis of each fold change being different from zero. ReplicateRUV, RUVIII and other normalization methods (ComBat, Spatial, plate median and unprocessed Q2NORM) which do not estimate fold changes directly, were followed by linear regression fold change estimation with limma (Ritchie et al., 2015, as described in the Demonstration of bias) and we derived ordinary t-test p-values P for each data subset, comparable to those of RUV4 and RUVinv. Each column of A is referred to as the (fold change) profile of a perturbation.

4 Optimal RUV settings by evaluation endpoints

RUV is a broad class of methods, and the choice of k (the dimension of the bias component of the data), the choice of the negative control gene set c, and the choice of RUV algorithm will significantly affect the results. A crucial step in the RUV application process, which is specific to each experimental context, is the evaluation of the end results (here the estimated fold change profiles) under each setting. In fact, such evaluation is equally important with any normalization method, and it is, or arguably should be, the standard process to carefully evaluate the normalization performance even when the normalization method is not in itself driven by parameter optimization as with RUV. In this section we present statistical endpoints to assess the normalization quality of estimated fold change profiles from L1000 or other, similar datasets.

4.1 Suggested normalization evaluation endpoints

We define a set of 7 possible evaluation endpoints, described next, to assess the quality of fold change estimates from L1000 after application of different RUV settings and other normalization methods. Each endpoint has its own biological and statistical motivation.

Evaluation by p-value distribution: The first two evaluation endpoints are based on the observation that systematic errors in data can sometimes be spotted in the distribution of p-values. With many perturbations we expect most transcripts not to be influenced, giving p-values randomly distributed in (0, 1), and some transcripts to be truly regulated, giving low p-values. Hence, we expect $\{P\}$ to follow an inflated uniform distribution and have a completely flat histogram above say 0.001 but a spike of increased histogram frequencies of p-values below 0.001. In Figure 3 this is best illustrated in panel a showing the λ optimal RUV p distribution for the NPC cell shRNA data of Figure 1. Figure 3B shows the p-value distribution of the corresponding Q2NORM data. It has a systematic inflation of "low but not significantly low" p-values. While

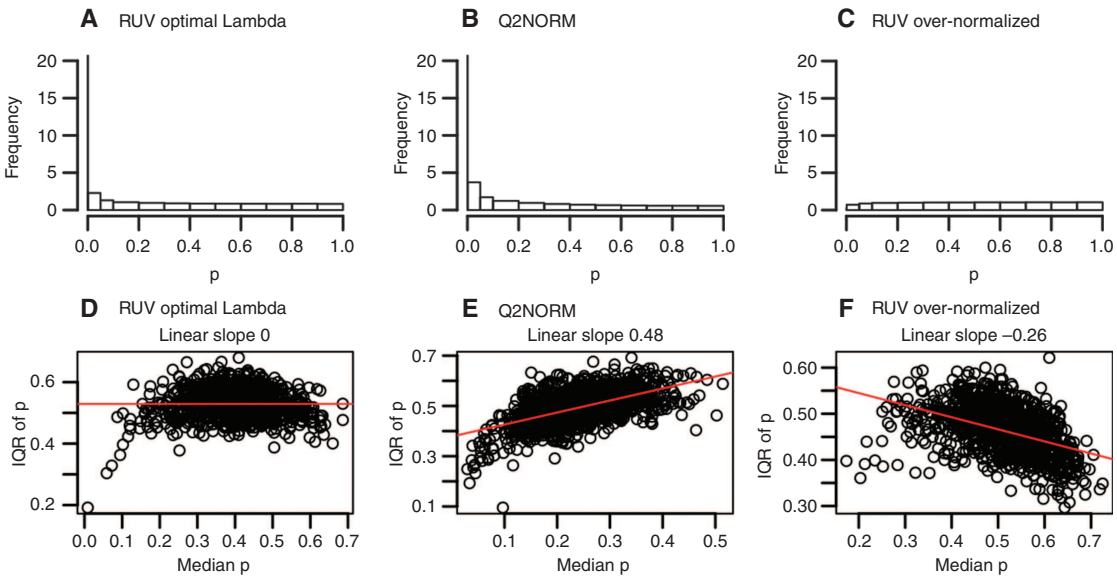

Figure 3: Fold change *p*-value distribution from (A) λ optimal RUV output, (B) original Q2NORM data and (C) over-normalized RUV output.

A dataset with no bias is expected to have uniform *p*-values (a flat histogram), except for a spike of low *p*-values to the left representing truly differentially expressed gene transcripts (see main text). The resemblance to the gold standard *p*-value distribution is measured by the endpoints unifKS and λ. Note that the three leftmost histogram bars of each panel are narrow (0–0.001, 0.001–0.05, 0.05–0.1), to illustrate the systematic overrepresentation of low but not significantly low *p*-values of Q2NORM. (D) shows the λ optimal RUV *p*-value distribution within rows of {*P*}, each row represented by the IQR versus median *p*-value. (E) reveals a systematic overrepresentation of low *p*-values (low IQR, low median) indicating bias, (F) similarly shows a systematic overrepresentation of high *p*-values (low IQR, high median), also indicating bias. The evaluation endpoint slopeHoriz is the linear regression slope of (D–F) and summarizes the *p*-value distribution within rows (gene transcripts). Example *p*-values from subset 2 of NPC cell type shRNA data is shown in A, B, D and E. The "bad" example of c and f originates from subset 1 of SHSY5Y cell type shRNA data.

low but not significantly low *p*-values can be caused by small, true effects in data, a consistent slope of *p*-value frequencies through a substantial part of the [0, 1] *p*-value range indicates bias and a need for more normalization (Gagnon-Bartsch and Speed, 2012). The endpoint **unifKS** is the Kolmogorov-Smirnov distance (Daniel 2000) between the subset of *p*-values larger than 0.001 {*P: P* > 0.001} and the uniform distribution on the same domain, U(0.001, 1). UnifKS measures how well the *p*-values follow the uniform distribution, but disregarding of the lowest *p*-values (<0.001). The inflation factor $\boldsymbol{\lambda}$ (**Lambda**) measures the amount of inflation of the median *p*-value: $\lambda = median[\chi_1^2(\{1-P\})]/\chi_1^2(0.5)$, where $\chi_1^2(x)$ is the 1 degrees of freedom Chi-square quantile of x, is used in a different context in Yang et al. (2011). With both these endpoints, a low value favors good normalization.

Evaluation by knockdown controls: The next two endpoints use biological information specific to shRNA or ORF data in that the direction of the fold change is sometimes known: some of the applied shRNAs are present among the 978 gene transcripts, and are hence known to be down-regulated. We call their estimates the *known negative fold changes*, and recognize that there is at most one such fold change in each shRNA profile (c.f. Figure 1). Known negative fold changes should, if not biased, be statistically different from zero. Consequently, they should have low *p*-values, relative to most other *p*-values in {*P*}. We rank all the 978 × d values of {*P*} (smallest p gives rank 1) and let **Q3P** be the 3rd quartile of the ranks of the known negative fold changes. A good normalization method should have a low Q3P (Figure 4). With a well performing normalization method, the known negative fold changes should include only negative values, whereas the other fold changes should be a mixture of negative, positive and (close to) zero values. **AdistKS** is the Kolmogorov-Smirnov distance between the distributions of these two subsets of {*A*}. The good normalization method will have a large AdistKS.

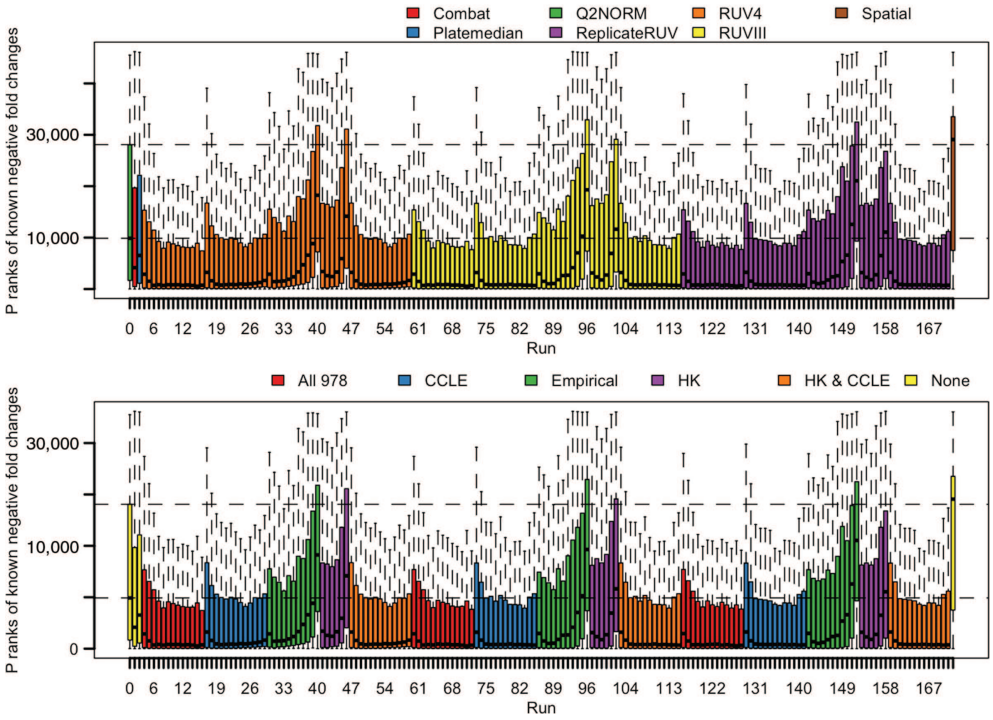

Figure 4: *p*-Value ranks of known negative fold changes in NPC cell shRNA data subset 2, coloured by normalization method (top) and negative control gene set (bottom) respectively.
Each box represents one normalization method or RUV setting. Ideally, we like all the ranks to be very low, but ultimately, we seek the lowest Q3P, the 75th percentile (upper edge of the box). We learn that within an RUV method and negative control set, Q3P tends to decrease with an increased low to moderate amount of bias subtraction (the RUV parameter k increases from left to right), except with Empirical and HK negative controls which are outperformed by other negative control sets. The leftmost bar is the unprocessed Q2NORM, with median and Q3P marked by dashed horizontal lines.

Evaluation by patterns in the matrix P: slopeHoriz and slopeVerti. For poorly normalized data, the heatmaps of {A} (Figure 1) reveal horizontal and vertical "stripes" of consistently low and high fold changes. Such stripes contradict the reasonable biological assumption that there is likely no transcript that is consistently up- or down-regulated by all perturbations in a subset, and that most perturbations only influence a few transcripts. (Highly global regulators of multiple transcripts were proposed, e.g. the gene MYC, Kress et al. (2015), but are likely rare). To detect such unwanted 'stripyness' we make use of the fact that 'stripes' lead to a specific distributional pattern within columns and rows of $\{P\}$. To illustrate this, each point in Figure 3D–F represents the interquartile range (IQR) versus the median of p-values for all the fold changes in one row of A. If p-values were all uniformly distributed, we would see an ellipse of points centered at (0.5, 0.5) and with vertical/horizontal principal axes. Since we do expect a zero inflated uniform distribution of $\{P\}$, we think that the better normalization method is similar to this pattern but with a slight overrepresentation of low-IQR-and-low-median points. With unprocessed Q2NORM p-values (Figure 3E), we see an enourmous overrepresentation of low-IQR-and-low-median points which indicates systematic bias. This dependency, quantified as a linear regression coefficient, is termed **slopeHoriz** for row wise stripiness and **slopeVerti** when instead summarizing column wise p-values. The better normalization method gives slopes close to zero.

Evaluation by the distribution within the matrix A: MAD reflects the width of the estimated fold change distribution, see the upper left colour key histograms of Figure 1A and F (MAD = Median Absolute Deviation from zero of $\{A\}$). This endpoint is included for reference, although it is not entirely motivated. Since Q2NORM shRNA data has an overrepresentation of fold changes with a large magnitude, efficient normalization will lower MAD. However, we note that MAD will also decrease if we just scale A down by a constant, an operation which does not reduce the systematic bias structures in data.

4.2 Optimal RUV settings under each of the 7 evaluation endpoints

To achieve the most appropriate bias removal and fold change estimation, we evaluated RUV and alternative normalization methods across a large range of parameter settings in a computationally intense comparison, comprising up to 205 RUV runs per data subset, to optimize a set of evaluation endpoints. For each of the 181 shRNA data subsets, optimal RUV settings with respect to each of the 7 endpoint were retrieved by choosing the run that minimized or maximized the value or magnitude of the endpoint appropriately. Figure 5A shows the endpoint values across all the normalization runs of an example shRNA subset (NPC cell subset 2).

The different endpoints produce systematically different optimal RUV outputs. To illustrate this, Figure 5B shows the endpoint values of the RUV4 runs with all 978 transcripts as negative controls only (runs 3–15). The runs are ordered by the parameter k, and hence the amount of bias removed. The runs coloured with pink are optimal with respect to one or more endpoints (maximal AdistKS, minimal magnitude slopeHoriz or slopeVerti, or minimal values of the other endpoints). Typically, the relatively smallest degree of normalization

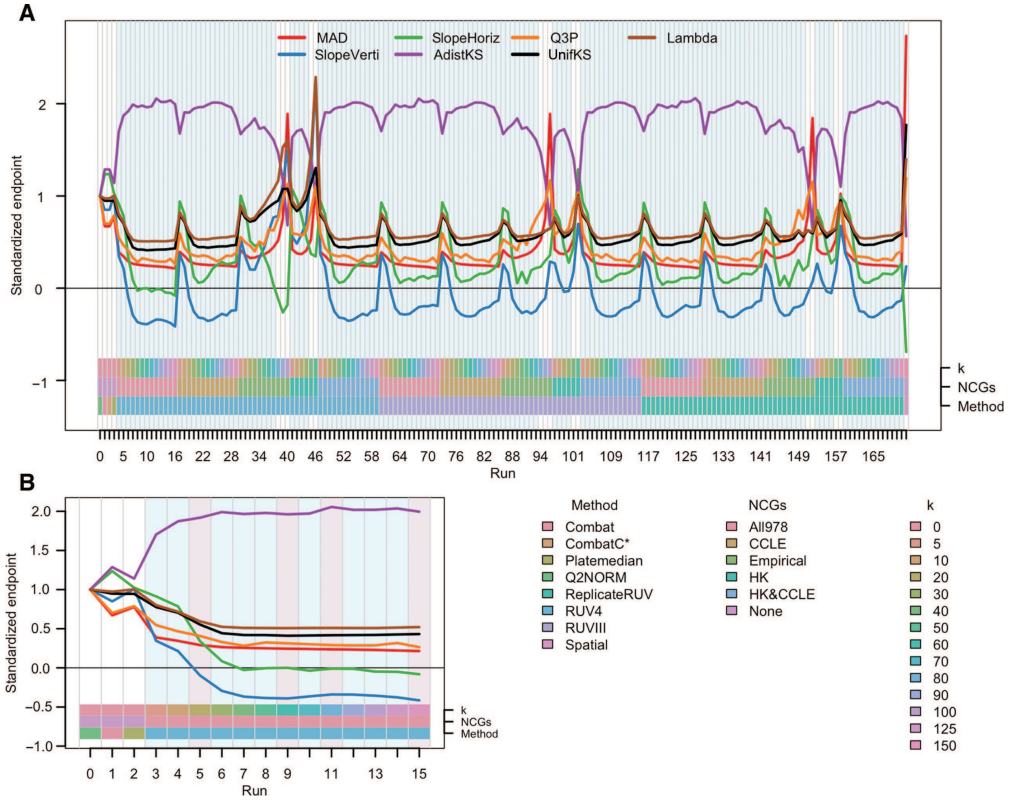

Figure 5: Assessment of large scale computing normalization by evaluation endpoints.
We used a set of statistically and biologically motivated evaluation endpoints (Y-axes) to summarize the quality of fold change estimates after applying different RUV settings and other normalization methods (runs, X-axes). The leftmost vertical bar of each panel shows unprocessed Q2NORM performance, to which each endpoint has been standardized so that Q2NORM has standardized endpoint level = 1. Optimal runs have high AdistKS, and low magnitude of λ (Lambda), unifKS, slopeHoriz, slopeVerti and to some extent MAD. (A) Quality of fold changes from Q2NORM and all 172 normalization runs that rendered output for a representative shRNA data subset (Neural Progenitor Cells, NPC, subset 2). Within each RUV algorithm and negative control gene set (NCG), the RUV parameter k (the number of potential bias vectors subtracted from data) increases from left to right. Note that for all settings, very low k gives clearly worse output according to all endpoints. Runs that passed an initial filtering for MAD = 0.2 and Pratio >1 (light blue) were further searched for optimality (see Appendix). (B) Quality of fold changes of our recommended RUV settings (RUV4 with All978 negative controls, runs 3–15). For these RUV settings, all endpoints indicate a decrease of bias as k increases from 5 to 20. MAD, by definition, advocates the maximum $k = 150$ (run 15, pink), joined by Q3P in this particular subset. SlopeVerti is optimal for $k = 20$ (run 5, pink), λ, unifKS and slopeHoriz for $k = 60$ (run 9, pink) and AdistKS for $k = 80$ (run 11, pink) in this data subset. *CombatC is Combat normalization based on mean standardized gene expression subset data.

is favored by slopeHoriz and slopeVerti, and the highest by MAD followed by Q3P and AdistKS. Hence, if aiming for more conservative normalization, subtracting less bias with the risk of keeping noise, RUV can be optimized for e.g. slopeVerti instead of for λ. Similarly, if aiming for more liberal normalization, subtracting more bias with the risk of losing true signal, RUV can be optimized for MAD instead of for λ.

More details on the running performance of different RUV settings and other normalization methods are shown in the Appendix.

5 Biological verification and generally recommended RUV settings

While the fold change estimates were statistically optimized into 7 suggested versions in the previous section, the ultimate aim of normalization is to increase the true biological information gained and decrease false positive results. With 7 full sets of RUV estimated fold change profiles for each cell of shRNA data, plus those of unprocessed Q2NORM, plate median, ComBat and spatial normalization proceed to a head-to-head method comparison for biological outcome.

We make use of the reasonable assumption that the estimated fold change profiles should – to a degree – correlate between cell types, for the same perturbation. This step was performed with the 70 most common perturbations among the 16 cell types in shRNA data, each of which was assayed in at least 13 cell types. Given the fold change profiles from a normalization method, let Θ_j be the set of N_j cell types in which perturbation j has been assayed. Denote by ρ_{ikj} the correlation between cell i's and cell k's profiles under perturbation j. We collect the correlations between all cell pairs $\psi_j = \{\rho_{ikj}\colon (i,k) \in \Theta_j, i < k\}$ and let the cell correlations Ψ be the set of such correlations over all 70 perturbations: $\Psi = \{\psi_j\colon j = 1,\ldots,70\}$. The cell correlations were compared between methods by density plots, using permutation distributions as a negative control. The permuted correlations were computed after randomizing perturbation labels of fold change profiles within each cell type. While randomly chosen perturbations might sometimes produce similar transcriptional effects in cells, we expect most of the random cell correlations to be close to zero. Following bias removal and fold change estimation, the endpoint optimized RUV outputs render a much more plausible distribution of random cell correlations, with values mostly around zero (Figure 6). The Kolmogorov-Smirnov distance D between the cell correlation and permuted cell correlation distributions was calculated to summarize the performance of each method. D is chosen as an acceptable and conservative approximation. Clearly, different cell lines are expected to produce somewhat different results. Also, randomly chosen pairs of shRNAs may be biologically related and could produce similar profiles.

In L1000 shRNA data, D is notably higher after the RUV method (Figure 6, Bottom 8 panels, $D \geq 0.591$) than after the other normalization methods (Figure 6, top 4 panels, $D \leq 0.285$). This suggests that RUV indeed makes an improvement to the Q2NORM shRNA data quality, which is more substantial than that of plate median, ComBat or Spatial normalization. We further see that the RUV outputs optimized with respect to of λ (Lambda) and unifKS (the endpoints measuring how uniform the p-value distribution is, both $D = 0.748$) or AdistKS and Q3P (using the known regulation direction of knocked down genes, $D = 0.741$, 0.733) outperform slopeHoriz and slopeVerti (the endpoints measuring the distribution of p-values within each row and column of fold changes, $D = 0.591$, 0.720) with respect to their capability to separate potentially correlated fold change profiles from random pairs of profiles. Notably, MAD does well with $D = 0.783$, but we do not genuinely consider this endpoint due to lack of statistical foundation and the risk that it will favor over-normalization.

Taken together, we choose to generally recommend λ as a useful endpoint for RUV optimization of shRNA data, because it has a high D, it can be applied to either of shRNA, ORF and drug data (as opposed to AdistKS and Q3P which can only be applied to part of shRNA and ORF data) and since it is threshold-free (as opposed to unifKS which measures the uniformness of p-values >0.001, an arbitrary cutoff).

Based on our results, we further recommend to use the RUV4 algorithm, using all 978 transcripts as the negative control set, since a total of 131/181 cluster runs gave the best λ value for this particular setting (more details are given in the Appendix). Separate λ optimization of shRNA fold changes with these recommended RUV settings resulted in a convincingly large $D = 0.754$ (Lambda* in Figure 6).

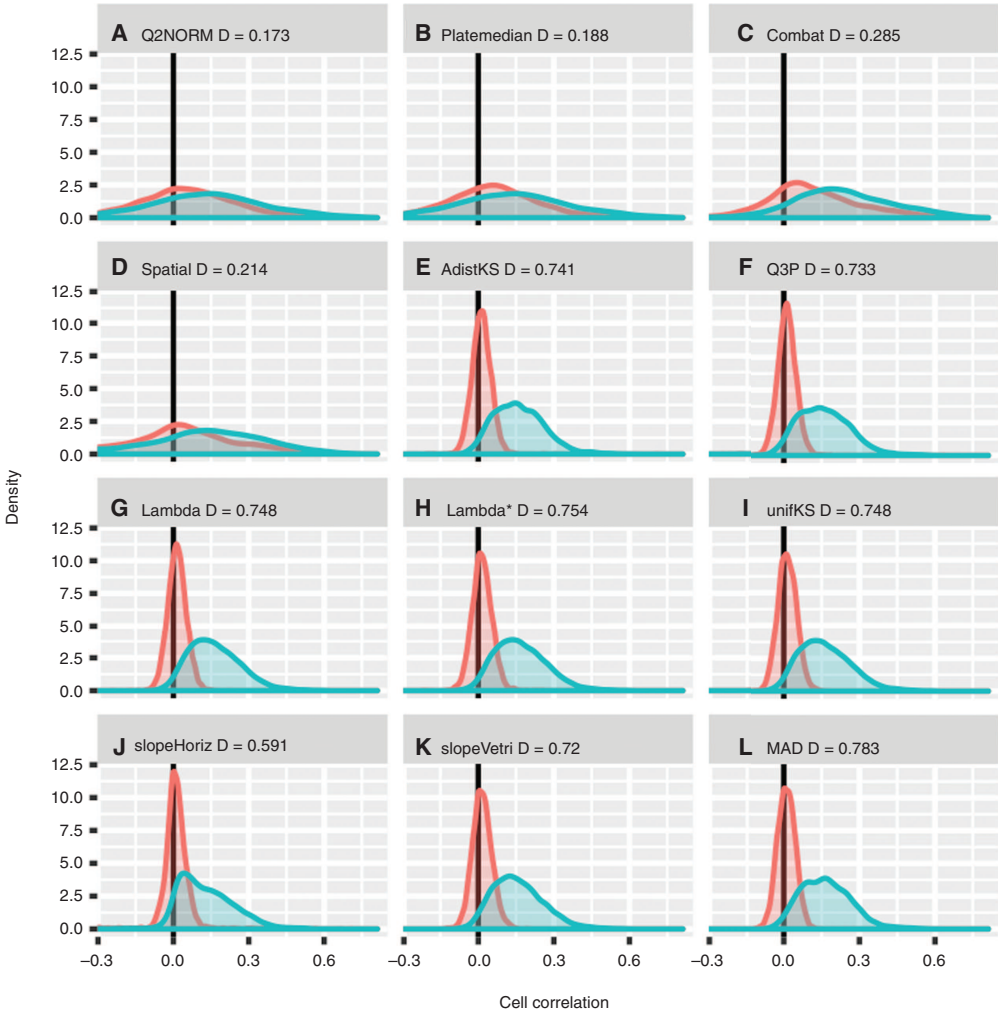

Figure 6: Global assessment of shRNA data normalization by between cell type validation.
The blue distributions represent the Pearson correlations between fold change profiles for the same perturbation but on pairs of different cell types, based on the 70 most common perturbations, each assayed in 13–16 cells. The red distributions represent the corresponding distribution after random permutation of perturbation labels. Thus, assuming that several perturbations tend to produce reasonably similar responses in different cell lines, the separation of the two distributions, measured as Kolmogorov-Smirnov distance *D*, provides an empirical summary of the quality of fold change profiles given by each normalization method or RUV optimization endpoint. Note that all RUV fold change profiles (E–L) are suggested to have a higher quality (higher *D*) than the other normalization methods (A–D). Only the 60 shRNA data subsets for which the true direction of regulation is known for some gene transcripts are included in this assessment, to render a fair comparison for AdistKS and Q3P, which can only be derived for such data subsets. Lambda* reflects the quality of fold changes from an RUV λ optimization in which only our generally recommended RUV settings (RUV4 with All978 negative controls) were considered.

5.1 L1000 fold-changes in ORF and drug data with our generally recommended RUV settings

With the above optimized RUV settings (RUV4 with all 978 transcripts as the negative control set) we proceed to estimate fold changes of L1000 ORF and drug data, in addition to that of shRNA. For drug data, we also see an improved overall performance (measured as D) from λ optimal RUV output compared to those of Q2NORM, plate median and ComBat (Appendix Figure A1). Similarly, for ORF data λ optimal RUV output is that with the highest D, but all the distances are very low (≤ 0.166). This may indicate that the ORF partition of L1000 is not of the same quality as the other types of treatment, or that there are dramatic differences in how cell lines repond to gene over-expression.

Unlike the shRNA and ORF data, which are gene-oriented, the drug data cannot be assessed by its effect on the target gene (which is frequently unknown, may not be unique, and may not be transcriptionally affected). However, drug data includes fold change profiles of several doses for many drugs, evaluated at a range of doses between nanomolar concentrations up to 300 μM. As a further verification of the data quality after normalization, we investigated dose-response trends, i.e. whether for any given drug there are readout transcripts that respond in a dose-dependent fashion, as determined by a trend test p-value (Siegel and Castellan, 1988). The trend tests were applied to multidose drugs (>2 doses) with at least one tentatively significant fold change ($p < 0.1$), rendering different numbers of trend tests for the different normalization methods (e.g. 403,020 with Q2NORM and 386,896 with λ optimal RUV, Appendix Table A7). In this analysis we found that λ optimal RUV output has the highest fraction of significant p-values in fold change trend tests across those doses, compared to Q2NORM, Platemedian and ComBat (Appendix Table A7). The fractions of p-values <0.05 range from 13.5% in Q2NORM and Platemedian to 17.4% with λ optimized RUV. This may at first seem a small improvement, but considering that the percentages relate to as many as 403,020 trend tests of Q2NORM fold changes, and 386,896 trend tests of RUV fold changes, the increased number of sensible findings is really quite substantial. Furthermore, the fact that RUV has the lowest number of trend test indicates that it has the highest sensitivity to false positive fold changes. Thus, there is good reason to assume that the proposed normalization will be applicable for assessment of drug-induced transcriptional changes.

6 Availability and implementation: FC1000

FC1000 is an acronym for Fold Change estimates for L1000 data. The computed FC1000 fold change matrices of shRNA, drug and ORF data, normalized by our generally recommended RUV setting (RUV4 with all 978 transcripts as negative controls) are distributed freely at our ftp server (nelanderlab.org/FC1000.html), together with the R package FC1000 which is needed to extract these matrices. Furthermore, the FC1000 R package contains easy to use source code to customize and perform RUV normalization and fold change estimation from L1000 data or other datasets from scratch. The derivation of processed fold change results of L1000 and similar datasets by massive computing will thus be readily available to users.

The FC1000 R package is thoroughly documented, and the Appendix includes some example R scripts and further descriptions to demonstrate its use.

7 Discussion

In summary, we have established that the Q2NORM L1000 gene expression data as downloaded from LINCSCLOUD suffers from substantial bias that naïve data normalization methods fail to remove. In order to retain fold change profiles from the L1000 bead arrays, the RUV method offers a flexible system with which systematic bias can be removed at the same time as estimating fold changes. In this project we have developed a framework that enables RUV application to L1000 gene expressions. Key challenges include the fact that computational time prohibits direct application to a dataset with more than 1 million arrays, and that RUV can be run with several different settings (parameters) that substantially tune the result. It is not clear how to process the L1000 in batches using RUV or how to adjust the RUV method to obtain globally valid results. To solve this problem, we developed a set of metrics, termed evaluation endpoints, to measure the quality of the fold change profile estimates. These evaluation endpoints are based on p-value distributions (unifKS, λ), internal knockdown controls (Q3P, AdistKS) and assessment of 'stripyness' and overall variability (slopeVerti, slopeHoriz and MAD). Based on the endpoints, we have optimized the RUV framework for the shRNA part of L1000 data, and derived settings which we recommend for all the three types of data: shRNA, drug and ORF. The RUV provides an improvement to the existing methods plate specific median, ComBat and spatial normalization. We supply an easy-to-use R package for retrieving RUV normalized fold changes from L1000 with any RUV settings, and we also supply the full set of L1000 gene expression fold changes normalized with our recommended RUV settings online.

The normalization done through RUV is dependent upon what we choose to estimate (β in equation 2). This makes our results deliberately optimized for the fold change profiles, but not for the original format of the Q2NORM data. However, some of the RUV methods do produce a "cleaned" version of the original data matrix as a by-product. It is beyond the scope of this project to evaluate the quality of such cleaned data, but somebody with an interest can easily use and examine it further.

A general issue with RUV normalization of bead array data is that with only few transcripts (978 in L1000), many genes which would biologically have been expected to be fairly invariable, and which would have been suitable negative controls for RUV, are deliberately not on the arrays. With L1000 data, RUV performed well using all the 978 transcripts as negative controls, but it is possible that results could have been even better had the Q2NORM data held some of the so called invariant genes, which are available at LINCSCLOUD in a less mature version of the data (LXB).

Spatial normalization across the 384 well plates outperformed RUV and ComBat for two small subsets of shRNA data (out of the 181 subsets), RUV and ComBat not even almost λ optimal. These subsets belong to two cells with few perturbations: all the arrays of each cell are in a single subset and originate from 2 to 6 plates respectively. The number of plates of all the 181 shRNA data subsets varies from 2 through 361, but is most often within 100–200. It is an open question whether spatial normalization does well within plates but sometimes fail to remove discrepancy between plates. That would explain why no large subsets are even almost λ optimal after spatial normalization. Spatial in combination with RUV did often perform much better than just spatial normalization, but did not in general make an improvement to that of just RUV.

One primary feature of RUV is the ability to remove unwanted variation in data without assessing its causes. The curious reader may still wonder about potential sources of unwanted variation in the L1000 dataset. Towards this aim we made a small investigation on our example data (NPC cell subset 2). For each gene separately we estimated variance components of perturbation, plate and well, naïvely regarding all these as random effects. Summarizing over the 978 genes we observed that most of the variance in the model was accounted for by plate [mean (inter-quartile range) 52% (47%–59%)] followed by perturbation [2.8% (1.7%–3.5%)] and well [0.1% (0%–0.2%)]. Notably, the residual variance was generally high [45% (39%–50%)], suggesting that much of the variance is due to yet other factors. One additional type of unwanted variation is seen as a negative correlation between the number of replicate arrays of a perturbation and its fold change estimates (blue horizontal bar above the Figure 1A heatmap).

As more L1000 data, or similarly structured data, will soon be available, its normalization and use deserves further study. For instance, the fact that multiple cell lines are included opens for interesting opportunities to compare transcriptional response across a broad range of tissue derivations. Similarly, accurate estimation of fold changes across several forms of perturbation, opens for association between compounds and shRNAs, which can gain new insight into targeting mechanisms of small molecules as well as gene function. The idea of normalization as a means to correct for systematic, unwanted variation is standard to the field of bioinformatics, but is worth repeatedly pointing out as a central strategy of efficacy improvement in data analysis of basically any multivariate dataset. The FC1000 RUV and endpoint optimization framework is specifically designed to normalize (i) data in the form of treatment (perturbation) rows $\times$ instance (gene transcript) columns, (ii) with respect to estimates of changes between each active treatment and a control treatment, and (iii) where for most treatments, most instances are not expected to change. However, exactly the same strategy as well as most parts of the software could be applied straight away to alternative situations like L1000 experiments aiming for more intricate effects estimates (with more advanced experimental designs), other array gene expression datasets similar to L1000, gene expression datasets of different data types like Perturb-SeqSeq data (Dixit et al., 2016), biological datasets where the instances are not genes, like RNA expression or copy number alteration data (Jörnsten et al., 2011), just any dataset satisfying (i)–(iii), or – with additional extensions – to datasets satisfying (i), aiming to estimate linear model effects and for which there are groups of instances for which the average expected effects are known. Both the RUV optimization of endpoints concept and the software FC1000 hence have potentials far beyond the retrieval of L1000 fold change estimates. Please see the Appendix for how the FC1000 R package can be used for different purposes.

On the topic of applying the FC1000 strategy to other types of data, we note that the endpoints unifKS, λ and MAD are applicable when most genes are expected to show no change in expression levels between conditions. The stripyness endpoints slopeHoriz and slopeVerti also rely on the assumption that with most perturbations, most genes are expected not to change, but these endpoints were particularly motivated by the stripy appearance of the data matrices, which was not expected for a biological reason. The Q3P and AdistKS are endpoints customized to the groups of genes with known up- or down-regulation. A different dataset, with a different design, may require partly or completely different endpoints to drive RUV normalization. With L1000 we chose to adhere to the endpoint which gave the most plausible biological results as quantified by the cell correlation densities. Using positive feedback in this way makes us less vulnerable to whether the endpoint assumptions are 100% fulfilled. The fact that correlations between fold changes of different cells do indeed increase on average, is confirming the usefulness of the endpoints. If normalizing a dataset different from L1000, and if there is no way to assess overall performance of different endpoints by positive feedback, then the choice of endpoints will be more crucial.

Normalization of the L1000 data can probably be further improved, a challenge which we leave open: There are still stripes after RUV normalization (Figure 1F and H), and we make no claim of having removed all the bias. We do, however, suggest that RUV decreases the risk of type I errors: It lowers the rate of false positive regulated genes. We believe that the proposed strategy of RUV with evaluation endpoints will help refine future normalization strategies of both L1000 and other high dimensional datasets.

Acknowledgment: We thank Johann Gagnon-Bartsch and Terry Speed for advice about RUV, and Patrik Johansson for useful discussions.

Funding: This work is supported by strategic research initiative eSSENCE, the Swedish Research Council (2014-03314), the Swedish Cancer Society (CAN 2011/1198, CAN 2014/579), the AstraZeneca-Scilifelab research collaboration, Strategic Research Foundation (BD15-0088) and the Swedish Childhood cancer foundation (PR2014-0143).

Conflict of interest statement: The authors declare that no conflict interest exists.

Appendix

A1 Performance of different RUV settings and other normalization methods

All the RUV settings were run on each of the 181 shRNA data subsets. Most commonly, the runs with very large k (>100) did not come through because of memory limitations, but most runs with $k \leq 100$ (which is the more reasonable range of values) did. Each RUV setting (algorithm and negative control set) did give output for some values of k for all the 181 subsets except those of RUVinv, which only gave output for two subsets (both with All 978 transcripts as negative controls).

We classified most RUV outputs as *acceptable* in a quick filtering for MAD $<$ 0.2 and Pratio >1 (Appendix Table A8). MAD $<$ 0.2 crudely ensures that the fraction of extreme fold change estimates is at least a little bit reasonable, as opposed to heatmaps indicating that almost all fold changes are non-zero in Figure 1A. Pratio is the ratio of frequencies in the leftmost to the second leftmost histogram bars of the p-value histograms in Figure 3A–C. In rare runs, basically all signal is removed (over-normalization), resulting in all fold change profiles effectively equal to zero. In L1000, this phenomenon systematically comes with Pratio <1, which motivates this filtering. All normalization runs of an example shRNA data subset (NPC cell subset 2) are summarized in Figure 5A, with acceptable runs highlighted blue.

Out of the 181 λ optimal shRNA data subset outputs (the general recommended RUV setting, see main text), 131 were produced by RUV4 with All978 negative controls, 48 by Combat and 2 by Spatial normalization. Subset details are shown in Appendix Table A4.

Some normalization runs have very similar endpoint values. We defined *almost λ optimal runs* as runs with $\lambda_{run}/\lambda_{Raw} \leq \lambda_{optimal} + 0.003$, where λ_{Raw} is the λ of Q2NORM data before further processing and $\lambda_{optimal}$ is the minimum observed λ across all runs of the subset. Appendix Table A9 shows the number of shRNA data subsets for which each RUV setting or normalization method was almost λ optimal. With RUV4 and All978, 162 out of the 181 subsets have almost λ optimal output. That strengthens our decision to name RUV4 with All978 our generally *optimized RUV setting*.

A2 Appendix Tables and Figure

Table A1: Cell types assessed by shRNA perturbations.

Cell ID	Cell type	N. subsets
1	HEK293T	1
2	SHSY5Y	1
3	SW480	1
4	HEKTE	2
5	SKL	3
6	ASC	5
7	NPC	5
8	MCF7	15
9	HCC515	18
10	HEPG2	18
11	A549	19
12	HA1E	19
13	PC3	17
14	A375	19
15	HT29	18
16	VCAP	20

Table A2: Cell types assessed by drug perturbations.

Cell ID	Cell type	N. subsets
1	THP1	1
2	NOMO1	1
3	HUH7	1
4	HEPG2	1
5	FIBRNPC	1
6	MCF10A	1
7	HS578T	1
8	BT20	1
9	MDAMB231	1
10	SKBR3	1
11	NKDBA	1
12	NEU	9
13	PHH	9
14	HCC515	8
15	SKB	12
16	ASC	12
17	HA1E	11
18	NPC	14
19	HT29	33
20	A375	36
21	A549	57
22	PC3	48
23	MCF7	52
24	VCAP	73

Table A3: Cell types assessed by ORF perturbations.

Cell ID	Cell type	N. subsets
1	HEK293T	6
2	VCAP	9
3	HEPG2	11
4	A549	9
5	HCC515	11
6	HA1E	11
7	MCF7	11
8	HT29	11
9	A375	11
10	PC3	11

Table A4: Descriptive Table of the 181 shRNA data subsets, including, for each *subset* (see main text) the numbers of active perturbations represented (N. active perturbations), the number of arrays which represent active perturbations (N. active arrays), the number of arrays which represent replicate baseline perturbations (N. baseline arrays), the total number of arrays (N. arrays), the total number of 384 well plates represented (N. plates) and the λ optimal RUV settings (see main text, "RUV4" means RUV4 with the All978 negative control gene set).

Global subset ID	Cell ID	Cell	Subset	N. active perturbations	N. active arrays	N. baseline arrays	N. arrays	N. plates	Lambda optimal method
1	1	HEK293T	1	71	667	16	683	2	Spatial
2	2	SHSY5Y	1	126	989	66	1055	3	RUV4
3	3	SW480	1	345	2028	58	2086	6	Spatial
4	4	HEKTE	1	199	1594	79	1673	9	RUV4
5	4	HEKTE	2	199	1485	79	1564	9	RUV4
6	5	SKL	1	187	1484	334	1818	14	RUV4
7	5	SKL	2	186	1484	334	1818	14	RUV4
8	5	SKL	3	187	1530	334	1864	14	RUV4
9	6	ASC	1	215	2290	824	3114	34	RUV4
10	6	ASC	2	215	2347	824	3171	34	RUV4
11	6	ASC	3	214	2110	824	2934	34	RUV4
12	6	ASC	4	215	2129	824	2953	34	RUV4
13	6	ASC	5	215	2225	824	3049	34	RUV4
14	7	NPC	1	215	2245	873	3118	36	RUV4
15	7	NPC	2	215	2293	873	3166	36	RUV4
16	7	NPC	3	214	2355	873	3228	36	RUV4
17	7	NPC	4	215	2277	873	3150	36	RUV4
18	7	NPC	5	215	2298	873	3171	36	RUV4
19	8	MCF7	1	202	2953	1261	4214	127	RUV4
20	8	MCF7	2	201	2849	1261	4110	127	RUV4
21	8	MCF7	3	202	2750	1261	4011	127	RUV4
22	8	MCF7	4	201	2963	1261	4224	127	RUV4
23	8	MCF7	5	202	2300	1261	3561	127	RUV4
24	8	MCF7	6	202	2082	1261	3343	127	RUV4
25	8	MCF7	7	201	2108	1261	3369	127	RUV4
26	8	MCF7	8	202	2073	1261	3334	127	RUV4
27	8	MCF7	9	201	2120	1261	3381	127	RUV4
28	8	MCF7	10	202	2175	1261	3436	127	RUV4
29	8	MCF7	11	202	2039	1261	3300	127	RUV4
30	8	MCF7	12	201	2083	1261	3344	127	RUV4
31	8	MCF7	13	202	2073	1261	3334	127	RUV4
32	8	MCF7	14	201	2081	1261	3342	127	RUV4

Table A4 (continued)

Global subset ID	Cell ID	Cell	Subset	N. active perturbations	N. active arrays	N. baseline arrays	N. arrays	N. plates	Lambda optimal method
33	8	MCF7	15	202	1995	1261	3256	127	RUV4
34	9	HCC515	1	196	2482	1421	3903	122	RUV4
35	9	HCC515	2	195	2180	1421	3601	122	RUV4
36	9	HCC515	3	196	2383	1421	3804	122	RUV4
37	9	HCC515	4	196	2491	1421	3912	122	RUV4
38	9	HCC515	5	195	2215	1421	3636	122	RUV4
39	9	HCC515	6	196	1682	1421	3103	122	Combat
40	9	HCC515	7	196	1667	1421	3088	122	Combat
41	9	HCC515	8	195	1878	1421	3299	122	Combat
42	9	HCC515	9	196	1640	1421	3061	122	Combat
43	9	HCC515	10	196	1707	1421	3128	122	Combat
44	9	HCC515	11	195	1972	1421	3393	122	Combat
45	9	HCC515	12	196	1890	1421	3311	122	Combat
46	9	HCC515	13	196	1855	1421	3276	122	Combat
47	9	HCC515	14	195	1780	1421	3201	122	Combat
48	9	HCC515	15	196	1838	1421	3259	122	Combat
49	9	HCC515	16	196	1670	1421	3091	122	Combat
50	9	HCC515	17	195	1618	1421	3039	122	Combat
51	9	HCC515	18	196	1735	1421	3156	122	Combat
52	10	HEPG2	1	198	2539	1578	4117	122	RUV4
53	10	HEPG2	2	198	2359	1578	3937	122	RUV4
54	10	HEPG2	3	198	2670	1578	4248	122	RUV4
55	10	HEPG2	4	198	2659	1578	4237	122	RUV4
56	10	HEPG2	5	197	2323	1578	3901	122	RUV4
57	10	HEPG2	6	198	1742	1578	3320	122	RUV4
58	10	HEPG2	7	198	1682	1578	3260	122	RUV4
59	10	HEPG2	8	198	1780	1578	3358	122	RUV4
60	10	HEPG2	9	198	1638	1578	3216	122	RUV4
61	10	HEPG2	10	198	1805	1578	3383	122	RUV4
62	10	HEPG2	11	198	1857	1578	3435	122	RUV4
63	10	HEPG2	12	198	1754	1578	3332	122	RUV4
64	10	HEPG2	13	198	1726	1578	3304	122	RUV4
65	10	HEPG2	14	197	1654	1578	3232	122	RUV4
66	10	HEPG2	15	198	1737	1578	3315	122	RUV4
67	10	HEPG2	16	198	1705	1578	3283	122	RUV4
68	10	HEPG2	17	198	1704	1578	3282	122	RUV4
69	10	HEPG2	18	198	1642	1578	3220	122	RUV4
70	11	A549	1	196	2553	1570	4123	135	RUV4
71	11	A549	2	196	2167	1570	3737	135	RUV4
72	11	A549	3	196	2472	1570	4042	135	RUV4
73	11	A549	4	196	2586	1570	4156	135	RUV4
74	11	A549	5	196	2392	1570	3962	135	RUV4
75	11	A549	6	196	2099	1570	3669	135	RUV4
76	11	A549	7	196	1921	1570	3491	135	RUV4
77	11	A549	8	196	1847	1570	3417	135	RUV4
78	11	A549	9	196	2057	1570	3627	135	RUV4
79	11	A549	10	196	1928	1570	3498	135	RUV4
80	11	A549	11	196	2041	1570	3611	135	RUV4
81	11	A549	12	196	2182	1570	3752	135	RUV4
82	11	A549	13	196	1811	1570	3381	135	RUV4
83	11	A549	14	196	2138	1570	3708	135	RUV4
84	11	A549	15	196	1933	1570	3503	135	RUV4

Table A4 (continued)

Global subset ID	Cell ID	Cell	Subset	N. active perturbations	N. active arrays	N. baseline arrays	N. arrays	N. plates	Lambda optimal method
85	11	A549	16	196	1976	1570	3546	135	RUV4
86	11	A549	17	196	1906	1570	3476	135	RUV4
87	11	A549	18	196	1778	1570	3348	135	RUV4
88	11	A549	19	196	1899	1570	3469	135	RUV4
89	12	HA1E	1	200	2669	1629	4298	143	RUV4
90	12	HA1E	2	200	2283	1629	3912	143	RUV4
91	12	HA1E	3	200	2612	1629	4241	143	RUV4
92	12	HA1E	4	200	2522	1629	4151	143	RUV4
93	12	HA1E	5	200	2444	1629	4073	143	RUV4
94	12	HA1E	6	200	1907	1629	3536	143	RUV4
95	12	HA1E	7	200	1880	1629	3509	143	RUV4
96	12	HA1E	8	200	1887	1629	3516	143	RUV4
97	12	HA1E	9	200	2027	1629	3656	143	RUV4
98	12	HA1E	10	201	1969	1629	3598	143	RUV4
99	12	HA1E	11	200	1991	1629	3620	143	RUV4
100	12	HA1E	12	200	2316	1629	3945	143	RUV4
101	12	HA1E	13	200	1868	1629	3497	143	RUV4
102	12	HA1E	14	200	2076	1629	3705	143	RUV4
103	12	HA1E	15	200	1950	1629	3579	143	RUV4
104	12	HA1E	16	200	1989	1629	3618	143	RUV4
105	12	HA1E	17	200	1908	1629	3537	143	RUV4
106	12	HA1E	18	200	1797	1629	3426	143	RUV4
107	12	HA1E	19	200	1879	1629	3508	143	RUV4
108	13	PC3	1	200	2975	1553	4528	153	RUV4
109	13	PC3	2	201	2752	1553	4305	153	RUV4
110	13	PC3	3	200	2819	1553	4372	153	RUV4
111	13	PC3	4	201	3050	1553	4603	153	RUV4
112	13	PC3	5	200	2530	1553	4083	153	Combat
113	13	PC3	6	201	2095	1553	3648	153	Combat
114	13	PC3	7	200	2142	1553	3695	153	Combat
115	13	PC3	8	201	2262	1553	3815	153	Combat
116	13	PC3	9	200	2291	1553	3844	153	Combat
117	13	PC3	10	201	2127	1553	3680	153	Combat
118	13	PC3	11	200	2315	1553	3868	153	Combat
119	13	PC3	12	201	2043	1553	3596	153	Combat
120	13	PC3	13	200	2408	1553	3961	153	Combat
121	13	PC3	14	201	2229	1553	3782	153	Combat
122	13	PC3	15	200	2276	1553	3829	153	Combat
123	13	PC3	16	201	2165	1553	3718	153	Combat
124	13	PC3	17	200	2073	1553	3626	153	Combat
125	14	A375	1	201	2762	1924	4686	155	RUV4
126	14	A375	2	202	2495	1924	4419	155	RUV4
127	14	A375	3	201	2841	1924	4765	155	RUV4
128	14	A375	4	201	2669	1924	4593	155	RUV4
129	14	A375	5	202	2491	1924	4415	155	RUV4
130	14	A375	6	201	1968	1924	3892	155	RUV4
131	14	A375	7	202	1920	1924	3844	155	RUV4
132	14	A375	8	201	1917	1924	3841	155	RUV4
133	14	A375	9	201	1950	1924	3874	155	RUV4
134	14	A375	10	202	2016	1924	3940	155	RUV4
135	14	A375	11	201	1983	1924	3907	155	RUV4

Table A4 (continued)

Global subset ID	Cell ID	Cell	Subset	N. active perturbations	N. active arrays	N. baseline arrays	N. arrays	N. plates	Lambda optimal method
136	14	A375	12	201	2270	1924	4194	155	RUV4
137	14	A375	13	202	1831	1924	3755	155	RUV4
138	14	A375	14	201	2017	1924	3941	155	RUV4
139	14	A375	15	202	2072	1924	3996	155	RUV4
140	14	A375	16	201	1886	1924	3810	155	RUV4
141	14	A375	17	201	1955	1924	3879	155	RUV4
142	14	A375	18	202	1826	1924	3750	155	RUV4
143	14	A375	19	201	1772	1924	3696	155	RUV4
144	15	HT29	1	204	3354	2074	5428	158	RUV4
145	15	HT29	2	203	3243	2074	5317	158	RUV4
146	15	HT29	3	204	3339	2074	5413	158	RUV4
147	15	HT29	4	203	3531	2074	5605	158	RUV4
148	15	HT29	5	204	2836	2074	4910	158	RUV4
149	15	HT29	6	204	2153	2074	4227	158	Combat
150	15	HT29	7	203	2073	2074	4147	158	Combat
151	15	HT29	8	204	2262	2074	4336	158	RUV4
152	15	HT29	9	204	2043	2074	4117	158	Combat
153	15	HT29	10	203	2282	2074	4356	158	Combat
154	15	HT29	11	204	2355	2074	4429	158	RUV4
155	15	HT29	12	203	2309	2074	4383	158	Combat
156	15	HT29	13	204	2223	2074	4297	158	Combat
157	15	HT29	14	204	2206	2074	4280	158	Combat
158	15	HT29	15	203	2214	2074	4288	158	Combat
159	15	HT29	16	204	2223	2074	4297	158	RUV4
160	15	HT29	17	203	2072	2074	4146	158	Combat
161	15	HT29	18	204	2011	2074	4085	158	Combat
162	16	VCAP	1	204	3171	2052	5223	191	Combat
163	16	VCAP	2	204	2750	2052	4802	191	Combat
164	16	VCAP	3	204	2880	2052	4932	191	Combat
165	16	VCAP	4	204	3169	2052	5221	191	Combat
166	16	VCAP	5	204	3131	2052	5183	191	Combat
167	16	VCAP	6	204	2520	2052	4572	191	RUV4
168	16	VCAP	7	204	3182	2052	5234	191	Combat
169	16	VCAP	8	204	2722	2052	4774	191	RUV4
170	16	VCAP	9	204	3662	2052	5714	361	RUV4
171	16	VCAP	10	204	2711	2052	4763	191	Combat
172	16	VCAP	11	204	2850	2052	4902	191	RUV4
173	16	VCAP	12	204	3635	2052	5687	191	Combat
174	16	VCAP	13	204	3092	2052	5144	191	Combat
175	16	VCAP	14	204	2968	2052	5020	191	RUV4
176	16	VCAP	15	204	3598	2052	5650	191	Combat
177	16	VCAP	16	204	2919	2052	4971	191	Combat
178	16	VCAP	17	204	2782	2052	4834	191	RUV4
179	16	VCAP	18	204	2886	2052	4938	191	Combat
180	16	VCAP	19	204	2757	2052	4809	191	RUV4
181	16	VCAP	20	204	2382	2052	4434	191	RUV4

Table A5: Frequencies of arrays for the perturbations in subset 2 of NPC cells.

Number of arrays	Number of perturbations
3	1
4	5
5	7
6	17
7	4
8	8
9	49
10	10
11	24
12	85
26	2
42	1
55	1
67	1
873	1

The single perturbation replicated in 873 arrays is the control (baseline).

Table A6: Frequencies of arrays for the perturbations of SHSY5Y cells.

Number of arrays	Number of perturbations
3	20
4	1
6	3
7	1
8	9
9	92
66	1

The single perturbation replicated in 66 arrays is the control (baseline).

Table A7: Percentage of significant dose response trends ($p < 0.05$ and $p < 0.10$), out of those within drug series of fold changes that showed a tentatively significant ($p < 0.1$) change of expression levels for at least one dose.

Method	Number of trends tested	$p < 0.05$ (%)	$p < 0.10$ (%)
Q2NORM	403020	13.5	50.2
Platemedian	504989	13.5	49.9
Combat	416063	15.5	53.5
λ optimized RUV	386896	17.4	54.2

Table A8: Number of the 181 shRNA data subsets with acceptable output fold changes (with for RUV methods at least one of the parameter k values we tried).

Negative controls	Platemedian	Combat	Spatial	ReplicateRUV	RUV4	RUVinv	RUVIII
All978				181	181	1	181
CCLE				180	180	0	180
Empirical				159	157	0	157
HK				96	95	0	95
HKCCLE				180	180	0	180
None	4	75	2				

Table A9: Number of the 181 shRNA data subsets with almost λ optimal output.

Negative controls	Platemedian	Combat	Spatial	ReplicateRUV	RUV4	RUVinv	RUVIII
All978				38	162	0	33
CCLE				41	84	0	36
Empirical				1	0	0	2
HK				1	0	0	1
HKCCLE				41	84	0	36
None	0	61	2				

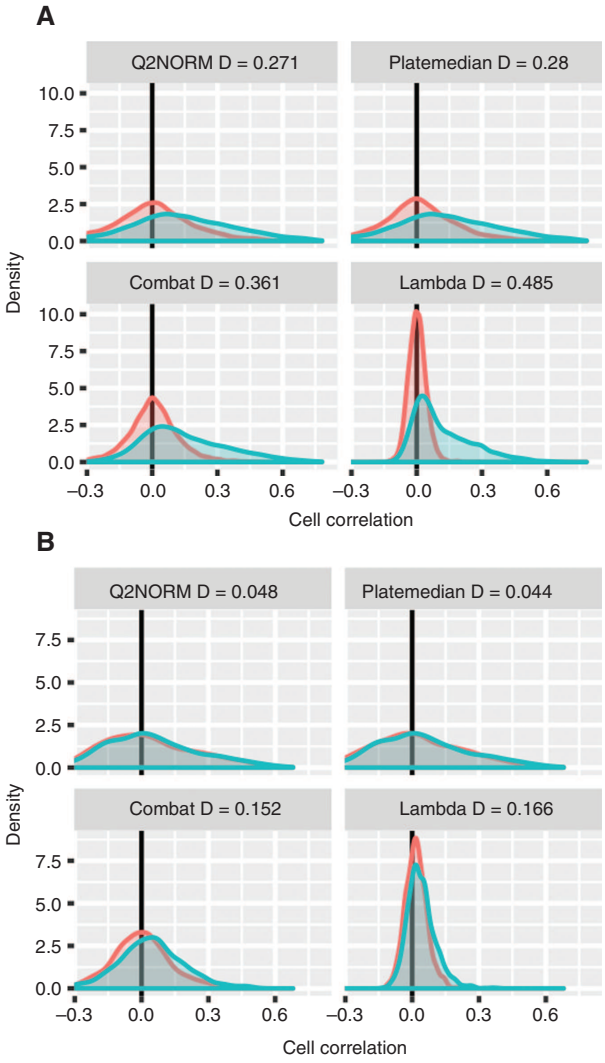

Figure A1: RUV improves fold change estimates for drug and ORF data.
Biological verification across cell lines to validate (A) drug treatment data and (B) Open Reading Frame overexpression data, respectively (see description Figure 5). λ (Lambda) optimized RUV produces better results than alternatives, as measured by distribution separation D, although the ORF data seems to contain relatively little information or be poorly suited for across cell line validation.

A3 Demonstration of FC1000 R-package

FC1000 is an R package which can be used to
- access FC1000 fold change profiles from L1000, normalized by our generally recommended RUV setting (RUV4 with all 978 transcripts as negative controls)
- customize and perform RUV normalization and fold change estimation from L1000 data from scratch
- normalize and estimate fold changes in big datasets other than L1000.

The FC1000 source code is freely available at our ftp server at Nelanderlab. Load the package into R like this:

```
library(FC1000)
```

Access FC1000 fold change profiles from L1000

To load RUV normalized fold change profiles into R, first dowload the shRNA, drug or ORF data file from Nelanderlab. This example will assume shRNA data is your interest. Save and unpack the downloaded folder (`tar -xzf shRNA.tar.gz` in the console) in the directory from which you will run your R session. In your data folder shRNA you will find a tab separated text file (`shRNA_subsets.tsv`) listing available cell types and subsets within which the RUV model was applied. The only fold change profile estimates available in the downloaded dataset are those obtained by our generally recommended RUV settings (RUV4 with all 978 transcripts as negative controls), optimized for the endpoint λ.

Load the complete set of lambda optimal RUV fold change profiles for the cell numbered 7 (NPC cell, merging all subsets), from the folder `shRNA`. Note that the folder must be named `shRNA`, `drug` or `ORF`.

```
A = getCellFC(cell = 7, folder = 'shRNA', optEndpoint = 'Lambda')
```

Customized RUV normalization and fold change estimation of L1000 data in 5 steps

In order to do your own, customized analysis with FC1000 you must first download the complete Q2NORM dataset, see http://www.lincscloud.org/, and then run our FC1000_data_prep matlab script available at Nelanderlab, which will prepare data and annotation matrices in a separate folder. The name of the folder, including the path to it, is handled by the variable 'inpathL1000' in the FC1000 R functions, with dummy value `mypath` below.

The normalization and estimation is divided into five steps. To enable demonstration of these scripts without downloading the complete L1000 dataset, the `FC1000` R package contains one data subset (NPC cell subset 2 of shRNA data, the main example of this paper). R code alternative to step 1 below is provided, which will create a small fake shRNA folder structure `shRNA_example`, upon which the other analysis steps can run.

1. Setup folder structure for shRNA L1000 data and split into subsets. This command will not work unless L1000 data has been downloaded and prepared with matlab script, but please also see the alternative code further below.

   ```
   nsubsets = subsets_FC1000(myRptype = "shRNA_example", inpathL1000 = mypath)
   ```

   The output `nsubsets` will only hold the integer number of subsets created. The function `subsets_FC1000` will have created a folder structure `shRNA_example where` data for the different subsets are stored. In addition, a tab separated text file (`shRNA_subsets.tsv`) is created in the shRNAfolder, listing the subsets by `Run = 1:nsubsets`.

Alternative R code, to create a small fake shRNA folder structure `shRNA_example`, upon which the other analysis steps can run:

```
data(NPC_2_shRNA)
attach(NPC_2_shRNA)
run = 15
folder = 'shRNA_example'
dir.create(folder, showWarnings = FALSE, recursive = TRUE)

i = INFO$Cell.ID[INFO$Run == run]
mycell = as.character(INFO$Cell[INFO$Run == run])
s = INFO$Subset[INFO$Run == run]
mycelldir = paste(i, mycell, sep = '_')
mydir = file.path(folder, mycelldir, paste('Subset_', s, sep = ''))

#Write subset specific data
dir.create(mydir, showWarnings = FALSE, recursive = TRUE)
save(list = c('Rgenes','design', 'plateT','plateC','pertT','pertT_unique', 'mycell', 'mydir'
),
     file = file.path(mydir , paste(i, '_', mycell, '_design_', s, '.Rdata', sep = '')))
saveRDS(cdat, file = file.path(mydir, paste(i, '_', mycell, '_Cdat_', s, '.rds', sep = '')))
saveRDS(mydat, file = file.path(mydir, paste(i, '_', mycell, '_MYdat_', s, '.rds', sep = '')
))


#Output information about the subsets of this cell
dir.create(file.path(folder, mycelldir, 'Information'), showWarnings = FALSE, recursive = TR
UE)

write.table(info, file.path(folder, mycelldir, 'Information', paste(i, '_', mycell, '_subset
_information.tsv', sep = '')),
            row.names = FALSE, quote = FALSE, sep = '\t')

#Output information about all subsets in this folder
write.table(INFO[INFO$Run == run,], file.path(folder, paste(folder,'subsets.tsv', sep = '_')
),
              row.names = FALSE, quote = FALSE, sep = '\t')
detach(NPC_2_shRNA)
```

In the next 4 steps, each subset is processed separately. The complete shRNA dataset is processed by looping the following subset specific functions over all subsets (181 subsets for shRNA data), preferably on a computer cluster. The subsets are called by the argument run, which refers to the `Run` numbering of subsets in the list of `shRNA_subsets.tsv`.

2. Run chosen normalization settings on one subset (number 15)

```
norm_FC1000(run = 15, folder = 'shRNA_example', doRaw = TRUE, doCombat = FALSE, doCombatC =
FALSE, doPlateMedian = TRUE,  methods = 'RUV4',  NCGs = 'All978', ks = c(10, 30, 50))
```

3. Calculate evaluation endpoints for one subset (number 15)

```
evalNormEndpoints_FC1000(run = 15, folder = "shRNA_example")
```

4. Plot evaluation endpoints for one subset (number 15) In this optional step, an Rmarkdown script is called which plots normalization performance summary plots for the given subset. Any Rmarkdown script can be called. The script `Rmarkdown_template_02.Rmd` is supplied with the `FC1000` R package and is in part a demonstration of the summary plot functions available in `FC1000`.

```
myRmd = system.file("rmd/Rmarkdown_template_02.Rmd", package="FC1000") #Use example Rmarkdow
n script provided
evalNormPlots_FC1000(run = 15, folder = "shRNA_example", rmdscript = myRmd) #Optional
```

5. Delete unnecessary files for one subset This step deletes a lot of files no longer necessary. It is optional but recommended to save memory. By default, only λ optimal and the lowest k almost λ optimal RUV

fold change estimates are kept, but more or other versions can be saved by altering the arguments `keep_optimal` and `keep_min_k_amongbest`.

```
clearFolders_FC1000(run= 15, folder = "shRNA_example") #Optional
```

After these steps of processing have been applied to all subsets of data, estimates of fold change profiles can be retrieved with `getCellFC` as described above. Please see the R help files of each function for details on avaliable analysis alteration options.

Normalize and estimate fold changes in big datasets other than L1000

In order to use the FC1000 R package for other datasets than L1000, a preparation step which puts the data into the structure set by our FC1000_data_prep matlab script available at Nelanderlab is needed. After that, follow steps 1–5 above, acknowledging that changes will be needed in the function `subsets_FC1000` of step 1, to extract the desired parts of the dataset into subsets of sizes which can be handled. The `subsets_FC1000` function is also where a different time point after treatment than the current 24 h could be specified for L1000.

References

Barretina, J., G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M. F. Berger, J. E. Monahan, P. Morais, J. Meltzer, A. Korejwa, J. Jané-Valbuena, F. A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I. H. Engels, J. Cheng, G. K. Yu, J. Yu, P. Aspesi, M. de Silva, K. Jagtap, M. D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R. C. Onofrio, T. Liefeld, L. MacConaill, W. Winckler, M. Reich, N. Li, J. P. Mesirov, S. B. Gabriel, G. Getz, K. Ardlie, V. Chan, V. E. Myer, B. L. Weber, J. Porter, M. Warmuth, P. Finan, J. L. Harris, M. Meyerson, T. R. Golub, M. P. Morrissey, W. R. Sellers, R. Schlegel and L. A. Garraway (2012): "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity," Nature, 483, 603–607.

Bolstad, B. M., R. A. Irizarry, M. Astrand and T. P. Speed (2003): "A comparison of normalization methods for high density oligonucleotide array data based on bias and variance," Bioinformatics, 19, 185–193.

Daniel, W. W. (2000): "Kolmogorov–Smirnov one-sample test," Applied Nonparametric Statistics, 2nd Ed., Duxbury Press, CA, USA, pp. 319–330.

Dixit, A., O. Parnas, B. Li, J. Chen, C. P. Fulco, L. Jerby-Arnon, N. D. Marjanovic, D. Dionne, T. Burks, R, Raychowdhury, B. Adamson, T. M. Norman, E. S. Lander, J. S. Weissman, N. Friedman and A. Regev (2016): "Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens," Cell, 167, 1853–1866.

Eisenberg, E. and E. Y. Levanon (2003): "Human housekeeping genes are compact," Trends Genet., 19, 362–365.

Freytag, S., J. Gagnon-Bartsch, T. P. Speed and M. Bahlo (2015): "Systematic noise degrades gene co-expression signals but can be corrected," BMC Bioinformatics, 16, 309.

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

Gagnon-Bartsch, J., L. Jacob and T. P. Speed (2013): Removing unwanted variation from high dimensional data with negative controls, Tech.report, Department of Statistics, University of California, Berkeley.

Jacob, L., J. Gagnon-Bartsch and T. P. Speed (2015): "Correcting gene expression data when neither the unwanted variation nor the factor of interest are observed," Biostatistics, 17, 16–28.

Johnson, W. E. and A. Rabinovic (2007): "Adjusting batch effects in microarray expression data using empirical Bayes methods," Biostatistics, 8, 118–127.

Jörnsten, R., T. Abenius, T. Kling, L. Schmidt, E. Johansson, T. E. Nordling, B. Nordlander, C. Sander, P. Gennemark, K. Funa, B. Nilsson, L. Lindahl and S. Nelander (2011): "Network modeling of the transcriptional effects of copy number aberrations in glioblastoma," Mol. Syst. Biol., 7, 486.

Kress, T. R., A. Sabò and B. Amati (2015): "MYC: connecting selective transcriptional control to global RNA production," Nat. Rev. Cancer, 15, 593–607.

Lachmann, A., F. M. Giorgi, M. J. Alvarez and A. Califano (2016): "Detection and removal of spatial bias in multiwell assays," Bioinformatics, 32, 1959–1965.

Leek, J. T., W. E. Johnson, H. S. Parker, A. E. Jaffe and J. D. Storey (2012): "The sva package for removing batch effects and other unwanted variation in high-throughput experiments," Bioinformatics, 28, 882–883.

Peck, D., E. D. Crawford, K. N. Ross, K. Stegmaier, T. R. Golub and J. Lamb (2006): "A method for high-throughput gene expression signature analysis," Genome Biol., 7, R61.

Pelz, C. R., M. Kulesz-Martin, G. Bagby and R. C. Sears (2008): "Global rank-invariant set normalization (GRSN) to reduce systematic distortions in microarray data," BMC Bioinformatics, 9, 520.

Ritchie, M. E., B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi and G. K. Smyth (2015): "Limma powers differential expression analyses for RNA-sequencing and microarray studies," Nucleic Acids Res., 43, e47.

Siegel, S. and N. J. Castellan (1988): Non-parametric statistics, McGraw-Hill, New York, pp. 399.

Yang, J., M. N. Weedon, S. Purcell, G. Lettre, K. Estrada, C. J. Willer, A. V. Smith, E. Ingelsson, J. R. O'Connell, M. Mangino, R. Mägi, P. A. Madden, A. C. Heath, D. R. Nyholt, N. G. Martin, G. W. Montgomery, T. M. Frayling, J. N. Hirschhorn, M. I. McCarthy, M. E. Goddard, P. M. Visscher and the GIANT Consortium (2011): "Genomic inflation factors under polygenic inheritance," European J. Hum. Genet., 19, 1–6.