Supplement

S5 Assessing The Relationship Between Actual and Measured Expression

One of the main components of dtangle’s approach is a linear model relating actual gene expression to measured gene expression. To explore its plausibility, we consider this linear model’s application to Affymetrix DNA microarray data and Illumina RNA-seq data.

S5.1 Microarray Data

To explore the relationship between the amount of transcripts and the measured expression from microarray technology we consider the Latin Square data set from Affymetrix (Irizarry *et al.*, 2003). This data set was created by hybridizing a solution of complex human background mRNA with 42 transcripts spiked in at concentrations ranging from 0.125pM to 512pM. The spike-ins were done with 3 technical replicates of 14 hybridization experiments in a Latin square design. This data set lets us explore the relationship between measured expression and abundance of the transcripts because for each of the spiked-in transcripts we know both the expression measured by the array and the amount in which the transcript was spiked in.

The expression as measured by the microarray is best explained by a logistic fit in the spike-in amount (Supplementary Figure S27a). However the linear fit that dtangle assumes does quite well. The logistic fit has a slightly smaller R^2 than the linear fit however there are several reasons we choose to model the relationship between spike-in amount and measured expression as linear. Firstly, the linear model is much simpler than the logistic model and has almost as good of a fit. For the linear model $R^2 = 0.957$ while for the logistic least squares fit we have $R^2 = 0.992$. Thus we gain relatively little for using the more complex model. Furthermore, the simplicity of the linear fit can also be thought of as a regularization of the logistic model. The non-linearity of the logistic curve means it is a very unstable model for measured expressions on both the high and low ends. That is, its inverse is undefined at or beyond these points. If the logistic model is used to estimate the true gene transcriptional abundance from measured expression data then small changes in measured expression might correspond to large changes in predicted amount. Indeed, the logistic curve will fail completely for measured expressions above its maximum or below its minimum. The linear model can be thought of as a regularized model between these two quantities. It ensures that a linear change in measured expression will only ever effect a linear change in amount. While there is probably a true non-linear relationship between the expression measured by microarrays and the amounts of transcripts in the samples, a linear fit does quite well at approximating this relationship and is a regularized model for the truth.

S5.2 RNA-seq

Another reason we favor linear modeling of the relationship between amount and measured expression is because it is not only reasonable for microarray technology but a reasonable model for RNA-seq. To explore how our model interacts with RNA-seq technology we consider data from the Sequencing Quality Control project (SEQC Consortium, 2015). These data are available on GEO with accession GSE47774. Here we look at RNA-seq analysis run on Ambion ERCC Spike-In Control Mix 1 using Illumina HiSeq technology. The ERCC spike-in control mix contains 92 transcripts spiked-in at known concentrations. Hence this data set allows us to look at the relationship between measured expression and amount because both are known.

For this data the measured expression values ($\log_2$ of the read count plus one) are well approximated by a linear relationship to the spike-in concentration amount (Supplementary Figure S27b). Unlike the previously discussed microarray data the RNA-seq data does not seem well approximated by a logistic fit. For a simple linear regression we find $R^2 = 0.955$ and so a linear fit seems reasonable.

S5.3 Estimating The Slope

We have thus seen that both microarray and RNA-seq measured gene expressions are well modeled as linear (on the log-scale) in the actual expressions. For the RNA-seq data we find that the slope of the linear relationship is approximately one. However for microarray data the relationship is better modeled as linear with a slope slightly smaller than one. Doing so will help account for the true logistic relationship that is affected by saturation and attenuation of the measured expressions at the low and high ends.

We denote the slope of this relationship as γ in our model and replace it with its estimate $\widehat{\gamma}$ when estimating the cell type proportions using dtangle. The value of $\widehat{\gamma}$ in dtangle’s algorithm may be set by the user if desired. However a pre-set value of $\widehat{\gamma}$ will be used by default if none is supplied. If $\widehat{\gamma}$ is not specified by the user, one need only specify the type of technology as either probe-level microarray, gene-level microarray, or RNA-seq. From here a default value of $\widehat{\gamma}$ is chosen. These default values are estimated from spike-in experiments like those just discussed. For both RNA-seq and microarray spike-in data we fit regression models of measured expression on spike-in amount. These are the linear models seen in in the previous sections. We then take the median value of all the estimates of the slopes from each gene’s regression model. These form the estimate of $\widehat{\gamma}$. This is done for the RNA-seq data (on the $\log_2$ of the counts plus one) and microarray data (at the RMA-summarized gene level and raw $\log_2$ probe-level). These estimates set the default values for $\widehat{\gamma}$ at .452 for probe-level microarrays, .699 for gene-level microarrays, and .943 for RNA-seq data. For other applications or situations lacking intuition for γ we recommend setting γ to one.

S5.3.1 Slope Sensitivity

In order to evaluate the sensitivity of dtangle to changes in $\widehat{\gamma}$ we conduct a meta-analysis of dtangle over many values for $\widehat{\gamma}$ (Supplementary Figure S10, S11, S12). dtangle seems to perform poorly if $\widehat{\gamma}$ is less than 0.5. However for $\widehat{\gamma}$ above about 0.5 dtangle is not particularly sensitive to the parameter. In any case dtangle seems robust to changes in $\widehat{\gamma}$ with best performance when $\widehat{\gamma}$ is between 0.5 and 1.

S6 Investigations Using Simulated Mixtures

To further investigate the role of robust scales, marker genes, cell type co-linearity, and the accuracy of dtangle we investigated the performance of deconvolution methods on a wide range of simulated data.

S6.1 Methods and Data

Broadly, the data simulation approach we take is to generate a matrix $U \in \mathbb{R}^{K\times N}$ of K reference cell type profiles across N genes, and a matrix $M \in \mathbb{R}^{S\times K}$ of K cell type mixing proportions across S samples and take their product (with some noise) to form a mixture gene expression matrix $X \in \mathbb{R}^{S\times N}$. We simulate data using both Gaussian and Poisson error at the log and linear scales, respectively, so that

1. in the Gaussian case $X \overset{def}{=} \exp(\log(MU) + E)$ where $E_{mn} \overset{iid}{\sim} N(0, f\sigma)$, $\sigma = sd(\log(1 + \text{vec}(U))$, and f is a multiplicative error factor controlling the level of noise
2. and in the Poisson case we let $Y_{mn} \overset{iid}{\sim} Pois((MU)_{mn})$.

The Gaussian error model will simulate data with character similar microarray data while the Poisson model will simulate data that more closely resembles RNA-seq.

We generate M with the following structure

$$M = (I_K \mid I_K \mid R)'$$

where the $S - 2K$ columns of R are uniformly drawn from the $(K-1)$-dimensional probability simplex. This structure of M means that the first $2K$ rows of X are just the references U with some error. These first rows of X are thus used as the reference data for deconvolution.

To generate U we follow two broad schemes. We will call the first the "artificial cell type" scheme and the second the "real cell type" scheme.

S6.1.1 Artificial Cell Type Mixtures

The artificial cell type references were simulated as follows. First we generated a baseline profile $B \in \mathbb{R}^N$ by taking the un-normalized read counts from the Parsons data set and gene-wise taking the median across the 39 samples for each of the $N = 23459$ genes. This baseline profile was then perturbed to create reference profiles for $K = 3$ artificial cell types as follows.

Let $\rho \in (0, 1)$ be the percentage of N genes that are markers of some cell type. Then let $G_k \subset \{1, \ldots, N\}$ be a set of $\lfloor N\rho/K \rfloor$ randomly selected type k marker genes. These genes are randomly selected among those genes in the top quartile of expression in B so that the G_k are mutually disjoint. We then form U through the following two steps:

1. make each reference profile a copy of B,

$$U \leftarrow \mathbb{1}_K \otimes B$$

2. for each cell type k set the expression level of marker genes G_k to some small value $\mu \in \mathbb{R}$ for all reference profiles other than the type k reference,

$$U_{tn} = \mu \text{ for all } n \in G_k, t \neq k \text{ and } k = 1, \ldots, K.$$

This scheme ensures that each cell type k has some set of marker genes G_k that are highly expressed in the type k reference (they are among the top 25% of overall expression) but lowly expressed in all other cell type references (at a low expression level μ).

S6.1.2 Real Cell Type Mixtures

Our scheme to generate "real" cell type references is much simpler. We let U be the reference of an existing data set. We use references from the Parsons or Linsley data in our simulations. Given the data set (Parsons or Linsley), we let the k^{th} row of U be the median of the reference profiles for the k^{th} cell type in the data set. For both the Parsons and Linsley data $K = 3$, while $N = 23459$ for the Parsons data and $N = 21421$ for the Linsley.

The artificial cell type simulations are useful because they allow us control over many simulation parameters. In addition to controlling the noise level f in the Gaussian case, we can control the percentage of actual marker genes ρ, and the expression level of marker genes in other cell types μ. On the other hand the real cell type simulations are interesting because they are more realistic than the artificially created cell types but also because they allow us to investigate realistic situations where the cell types are very different (Parsons) and very similar (Linsley).

In all cases after X has been generated the data is TPM normalized and analyzed using precisely the same procedure as described in the main body of this paper treating the first $2K$ rows of X as reference samples of the K cell types. Notably we do not reveal the true marker genes to the deconvolution methods in the case of the artificial cell type simulations.

We evaluated the performance of dtangle, the four other partial deconvolution algorithms (CIBERSORT, EPIC, LS Fit, and Q Prog) and a simple linear regression approach, on this simulated data. For all methods other than dtangle we evaluated the algorithms using both the linear scale data as generated, and logarithmically transforming the data using a base-2 logarithm of one plus the expression. Importantly, the other partial deconvolution methods do not fit using log scale data. For example, CIBERSORT's code explicitly forces data to be on a linear scale and EPIC uses linear-scale TPM-transformed read counts. Nonetheless, it will be instructive to look at these methods using both linear and log-scale expressions. We do not do this similar comparison for dtangle because its approach does not fall nicely into either category, it combines both scales. Hence such a comparison does not make sense for dtangle. Instead we put dtangle in its own hybrid category.

The simple regression approach, mentioned above, simply estimates M by regressing the mixture samples' expressions onto the reference expressions. This is done using both linear and log-scale expressions. We included this regression approach because it serves as an easily understandable baseline against which we may compare other methods.

S6.2 Scale and Robustness

In Supplementary Figure S28 we plot boxplots of error and correlation along with scatter plots of estimates against true mixing proportions showing the performance of dtangle, the four partial deconvolution methods, and the linear regression approach, on artificial cell type simulated data with a low level of Gaussian error. We display these plots of the methods using both linear and log-scale expressions. The data was simulated so that 15% of the genes were markers ($\rho = .15$), the marker genes were only expressed by the cell type they mark ($\mu = 0$), and the added Gaussian error is 2.5% of the typical variance among expressions ($f = .025$). We plot similar figures using the Poisson error structure and the same values of ρ and μ in Supplementary Figure S33. Similarly we plots these figures for the real cell type mixtures using the Parsons data (Gaussian error: Supplementary Figure S37, Poisson error: Supplementary Figure S40) and the Linsley data (Gaussian error: Supplementary Figure S42, Poisson error: Supplementary Figure S45). The same value of f is used for the real cell type simulations with Gaussian error.

We can see from all of these figures that broadly dtangle out-performs other methods but also that the other partial deconvolution methods tend to perform better deconvolving linear scale expressions than the log-scale expressions. This makes sense because our data has been simulated as a linear mixture of linear scale expressions and since the simulation error in these figures is small ($f = .025$ in for the Gaussian simulations). The simulated data follows exactly the model presumed by these methods.

We argue that while a linear mixing of linear expressions is a plausible model, it is not robust. To show that this is true we adjust our simulations in two ways. First, we look at the same Gaussian simulations but change the error factor f from 2.5% to 75% (i.e. $f = .75$). The same plots with a high level of Gaussian error are Supplementary Figure S29, S38, S43. While dtangle still out-performs other methods, we see now that the other partial deconvolution algorithms perform better using log-scale expressions than linear scale expressions. Notably the data has still been generated using a linear mixing model of linear expressions, we have only increased the Gaussian error. Yet the log-scale expressions give a better fit even though the model is mis-specified fitting with log-scale expressions. The reason the log-scale expressions give a better fit in the high-error situation is because the logarithmic transformation attenuates the effects of the highly skewed data and the undue influence of points in the tail of the data.

A similar situation occurs if we leave the error low ($f = .025$ for Gaussian simulations) but add outliers to the data. For both the Gaussian and Poisson error structure we simulate data as previously described but then add five random outliers to each of the reference profiles in X. The value of these outlying points is set (on the log-scale) as 1.25 times the largest observation before adding any outliers. We plot similar figures for each of the simulations after adding outliers (see Supplementary Figure S30, S34, S39, S41, S44, S46). Largely the results are the same as the high-error Gaussian case. We see that the other partial deconvolution algorithms perform better after a logarithmic transformation since this ameliorates the effects of the outliers. Notably dtangle is relatively unperturbed by the outliers. This is because dtangle robustly combines expressions on a log-scale before averaging. Furthermore dtangle's averaging approach is not highly influenced by a single outlying point. In contrast, the outlying point becomes a high-leverage point for the other regression-like partial deconvolution approaches and thus is highly-influential on the estimates.

S6.3 Marker Genes

S6.3.1 Marker Gene Expression

A central feature of dtangle's approach its use of marker genes. Broadly, we define a marker gene as one which is expressed predominantly in only one cell type. All deconvolution methods seem to benefit from marker genes. Typically, they are used to sub-set the data on which the model is fit. dtangle has a unique use of marker genes and rigorously defines marker genes as only being expressed in one cell type. Nonetheless, we realize that this assumption is a mathematical approximation to the truth and so it is worth investigating what happens to dtangle when it is violated.

To investigate this we simulate data according to our artificial cell type simulation scheme with $\rho = .15$ so that 15% of the genes are marker of some cell type and using both a Gaussian error structure ($f = .025$) and a Poisson error structure. We then vary the expression of marker genes in other cell types μ. In Supplementary Figure S32 and Supplementary Figure S36 we plot (A) the absolute error and (B) correlation of dtangle's estimates from the truth varying the value of μ from the minimum of the data to letting μ be the maximum of the data. We plot the error or correlation on the y-axis and set μ to be the q^{th} quantile of the data varrying q along the x-axis.

In either case we see that as we increase the value of μ, and as it gets further from our mathematical assumption that $\mu = 0$, the error of dtangle increases. However this increase is very slow. Indeed, the marker genes do not need to have a true expression of $\mu = 0$ in all other cell types. So long as the expression of marker genes in other cell types is in, say, the bottom 25% of all gene expression dtangle does very well. Thus dtangle seems quite robust to this marker gene assumption.

S6.3.2 Number of Marker Genes and Co-linearity

For all deconvolution algorithms, including dtangle, it is important to find a good set of marker genes. In real data, the primary reason it can be difficult to find marker genes is some combination of (1) that there are many cell types we wish to deconvolve and (2) the cell types we wish to deconvolve are closely related. This follows because our definition of a marker gene is a gene that is highly expressed in only one cell type. Thus the more cell types we have, the harder it is to find a gene is that expressed highly in only one of the cell types. Similarly, if the cell types we wish to deconvolve a closely related and their expression profiles are highly co-linear then finding genes highly expressed in one cell type but not the others is difficult.

To explore the performance of dtangle in situations where marker genes are hard to identify we simulate according to our artificial cell type scheme and vary the percentage of genes that are markers of some cell type (ρ). For a Gaussian error structure we plot in Supplementary Figure S31 the (A) error or (B) correlation of dtangle's estimates against the true mixing proportions, on the y-axis, against the percentage of marker genes in the data (ρ), on the x-axis. We vary ρ from 0.01 to 0.2. We plot a similar plot in Supplementary Figure S35 using the Poisson error structure. What we can see from these two figures is that dtangle's performance only suffers drastically when less than about 2-3% of the genes are good markers of a cell type. So long as at least 3-5% of the genes in the data are markers of some type dtangle does quite well.

To explore this issue further we also revisit the results from the real cell type mixture simulations, Supplementary Figure S37-S46. In these simulations we simulated mixtures of using the references from the Parsons and Linsley data sets. The Parsons data set is a mixture of three very distinct cell types: Brain, Liver and Muscle. On the other hand the Linsley data set is a mixture of three closely related white blood cell classes: Lymphocytes, Monocytes, and Neutrophils. It should be relatively easy to find marker genes for the Parsons data set, because the cell types are very distinct, and relatively more difficult to find marker genes for the simulated mixture of closely-related white blood cells using the Linsley reference data. While we do see that dtangle has relatively more trouble deconvolving the Linsley-derived simulations than the Parsons-derived simulations, e.g. compare Supplementary Figure S37 to Supplementary Figure S42, dtangle still does well over-all. Indeed dtangle still out-performs the other partial deconvolution methods.

Over all we see that dtangle, like other deconvolution algorithms, will suffer if there are almost no marker genes of the cell types. However dtangle is quite robust and works well with as few as a couple of percent of the genes being marker of some cell type.

S6.4 Other Remarks

S6.4.1 Accuracy as a Function of the Truth

We see from all of these simulations that the accuracy of dtangle's estimates do not seem to depend strongly on the true mixing proportion. That is, dtangle estimates accurately when the true mixing proportion is close to zero and when the true mixing proportion is close to one.

S6.4.2 Gamma

The simulations we have explored in this section follow a linear mixing model of linear expressions. This is a simplification of the model that dtangle posits. It is simplified because dtangle's model also includes an adjustment term γ. Hence simulations strictly according to dtangle's model would posit a linear mixing of slightly transformed linear expressions. We chose to simulate linear mixing on a linear scale because this is the model assumed by other deconvolution algorithms. Thus our simulations should be a fair analysis of these other deconvolution algorithms because it follows their model, not dtangle's. Effectively, we have simulated data assuming $\gamma = 1$. This shows that dtangle works quite well even when γ is not required in the model.

S7 The Mathematics of dtangle

Assume we have a mixture sample of K cell types. Let $Y \in \mathbb{R}^N$ be the (base-2) log-scale expression measurements of this mixture sample and $p_1, \ldots, p_K$ be the mixing proportions of the cell types. For $k = 1, \ldots, K$ assume that there are ν_k reference samples of cell type k and let $Z_{kr} \in \mathbb{R}^N$ be the log-scale expressions of the r^{th} type k reference. Furthermore, let $G_k \subset \{1, \ldots, N\}$ be the set of type k marker genes. We require that these marker gene sets are mutually disjoint.

Let $g_k = |G_k|$ and define $\overline{Y_{G_k}} = \frac{1}{g_k} \sum_{n \in G_k} Y_n$ and $\overline{Z_{G_k}} = \frac{1}{g_k \nu_k} \sum_{n \in G_k} \sum_{r=1}^{\nu_k} Z_{krn}$ to be the average of all type k marker genes across the mixture and reference samples, respectively. Finally, denote our "adjustment term" as $\gamma \approx 1$.

Let η_{kn} be the actual linear-scale expression of the n^{th} gene in a sample of type k cells and η_n be the actual linear-scale expression in the mixture, then dtangle assumes these actual expressions mix linearly,

$$\eta_n = \sum_{k=1}^{K} p_k \eta_{kn}. \tag{1}$$

Furthermore dtangle assumes that the measured log-scale expressions are linear in the actual log-scale expressions,

$$\begin{aligned} Y_n &= \mu + \theta_n + \gamma \log_2(\eta_n) + \varepsilon_n \\ Z_{krn} &= \alpha + \theta_n + \gamma \log_2(\eta_{kn}) + \varepsilon_{krn}. \end{aligned} \tag{2}$$

and that marker genes are (approximately) expressed by only one cell type so that if n is a marker gene for cell type k ($n \in G_k$) then

$$\eta_{\ell n} = 0 \text{ for all } \ell \neq k. \tag{3}$$

Combining Equation (1), Equation (2) and Equation (3) we then have that for $n \in G_k$,

$$\begin{aligned} Y_n &= \mu + \theta_n + \gamma \log_2(p_k \eta_{kn}) + \varepsilon_n \\ &= \mu + \theta_n + \gamma \log_2(p_k) + \gamma \log_2(\eta_{kn}) + \varepsilon_n \\ Z_{krn} &= \alpha + \theta_n + \gamma \log_2(\eta_{kn}) + \varepsilon_{krn}. \end{aligned} \tag{4}$$

So for $n \in G_k$ we have

$$\begin{aligned} \overline{Y_{G_k}} &= \mu + \overline{\theta_{G_k}} + \gamma \log_2(p_k) + \gamma\overline{\log_2(\eta_{G_k})} + \overline{\varepsilon_{G_k}} \\ \overline{Z_{G_k}} &= \alpha + \overline{\theta_{G_k}} + \gamma\overline{\log_2(\eta_{G_k})} + \overline{\overline{\varepsilon_{G_k}}}. \end{aligned} \tag{5}$$

where

$$\begin{aligned} \overline{\theta_{G_k}} &= \frac{1}{g_k} \sum_{n \in G_k} \theta_n \\ \overline{\log_2(\eta_{G_k})} &= \frac{1}{g_k} \sum_{n \in G_k} \log_2(\eta_{kn}) \\ \overline{\varepsilon_{G_k}} &= \frac{1}{g_k} \sum_{n \in G_k} \varepsilon_n \\ \overline{\overline{\varepsilon_{G_k}}} &= \frac{1}{g_k \nu_k} \sum_{n \in G_k} \sum_{r=1}^{\nu_k} \varepsilon_{krn}. \end{aligned}$$

This means

$$\begin{aligned} \overline{Y_{G_k}} - \overline{Y_{G_t}} &= \gamma \log_2(p_k/p_t) \\ &\quad + \overline{\theta_{G_k}} - \overline{\theta_{G_t}} + \gamma\overline{\log_2(\eta_{G_k})} - \gamma\overline{\log_2(\eta_{G_t})} \\ &\quad + \overline{\varepsilon_{G_k}} - \overline{\varepsilon_{G_t}} \\ \overline{Z_{G_k}} - \overline{Z_{G_t}} &= \overline{\theta_{G_k}} - \overline{\theta_{G_t}} + \gamma\overline{\log_2(\eta_{G_k})} - \gamma\overline{\log_2(\eta_{G_t})} \\ &\quad + \overline{\overline{\varepsilon_{G_k}}} - \overline{\overline{\varepsilon_{G_t}}} \end{aligned} \tag{6}$$

and so

$$\begin{aligned} D_{kt} &= \frac{1}{\gamma}\left(\left(\overline{Y_{G_k}} - \overline{Y_{G_t}}\right) - \left(\overline{Z_{G_k}} - \overline{Z_{G_t}}\right)\right) \\ &= \log_2(p_k/p_t) + \delta \end{aligned} \tag{7}$$

where $\delta = \frac{1}{\gamma}\left(\overline{\varepsilon_{G_k}} - \overline{\overline{\varepsilon_{G_k}}} - \overline{\varepsilon_{G_t}} + \overline{\overline{\varepsilon_{G_t}}}\right)$.

Now as $g_k \to \infty$ for all k then $\delta \to 0$ and so for a reasonably large number of marker genes

$$D_{kt} \approx \log_2(p_k/p_t)$$

and so since $D_k = (D_{k1}, \ldots, D_{kK})$ then

$$D_k \approx (\log_2(p_k/p_1), \ldots, \log_2(p_k/p_K))$$

and so if $L_k(x) = 1/\left(1+\sum_{t\neq k} 2^{-x_t}\right)$ then

$$L_k(D_k) \approx p_k$$

and so the dtangle estimator $L_k(D_k)$ approximates p_k.

References

Abbas, A. R., Wolslegel, K., Seshasayee, D., Modrusan, Z., and Clark, H. F. (2009). Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. *PLoS ONE*, **4**(7).

Becht, E., Giraldo, N. A., Lacroix, L., Buttard, B., Elarouci, N., Petitprez, F., Selves, J., Laurent-Puig, P., Sautès-Fridman, C., Fridman, W. H., and de Reyniès, A. (2016). Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression. *Genome biology*, **17**(1), 218.

Gaujoux, R. and Seoighe, C. (2012). Semi-supervised Nonnegative Matrix Factorization for gene expression deconvolution: A case study. *Infection, Genetics and Evolution*, **12**(5), 913–921.

Gong, T., Hartmann, N., Kohane, I. S., Brinkmann, V., Staedtler, F., Letzkus, M., Bongiovanni, S., and Szustakowski, J. D. (2011). Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples. *PLoS ONE*, **6**(11).

Irizarry, R. A., Hobbs, B., Collin, F., BeazerâŁłBarclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**(2), 249–264.

Kuhn, A., Thu, D., Waldvogel, H. J., Faull, R. L. M., and Luthi-Carter, R. (2011). Population-specific expression analysis (PSEA) reveals molecular changepcts restrict to markers, s in diseased brain. *Nature methods*, **8**(11), 945–7.

Linsley, P. S., Speake, C., Whalen, E., and Chaussabel, D. (2014). Copy Number Loss of the Interferon Gene Cluster in Melanomas Is Linked to Reduced T Cell Infiltrate and Poor Patient Prognosis. *PLoS ONE*, **9**(10), e109760.

Liu, R., Holik, A. Z., Su, S., Jansz, N., Chen, K., Leong, S., Blewitt, M. E., Smyth, G. K., and Ritchie, M. E. (2015). Why weight ? Modelling sample and observational level variability improves power in RNA-seq analyses. **43**(15).

MAQC (2006). The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. **24**(9), 1151–1161.

Newman, A. M., Chih Long Liu, Michael R. Green, Andrew J. Gentles, W. F., Yue Xu, C. D. H., Diehn, M., and Alizadeh, A. A. (2015). Robust enumeration of cell subsets from tissue expression profiles. *Nat Methods.*, **12**(5), 193–201.

Parsons, J., Munro, S., Pine, P. S., Mcdaniel, J., Mehaffey, M., and Salit, M. (2015). Using mixtures of biological samples as process controls for RNA-sequencing experiments. *BMC Genomics*, pages 1–13.

Racle, J., de Jonge, K., Baumgaertner, P., Speiser, D. E., and Gfeller, D. (2017). Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data. *eLife*, **6**.

Repsilber, D., Kern, S., Telaar, A., Walzl, G., Black, G. F., Selbig, J., Parida, S. K., Kaufmann, S. H. E., and Jacobsen, M. (2010). Biomarker discovery in heterogeneous tissue samples -taking the in-silico deconfounding approach. *BMC bioinformatics*, **11**, 27.

SEQC Consortium (2015). HHS Public Access. **32**(9), 903–914.

Shen-Orr, S. S., Tibshirani, R., Khatri, P., Bodian, D. L., Staedtler, F., Perry, N. M., Hastie, T., Sarwal, M. M., Davis, M. M., and Butte, A. J. (2010). Cell type-specific gene expression differences in complex tissues. *Nat Methods*, **7**(4), 287–289.

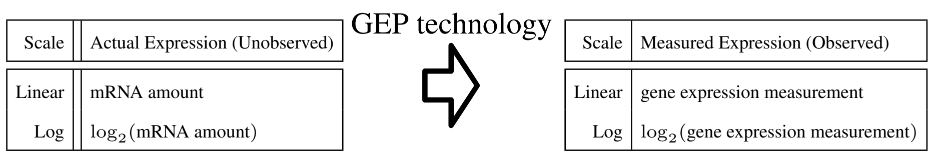
Zhong, Y., Wan, Y.-W., Pang, K., Chow, L. M. L., and Liu, Z. (2013). Digital sorting of complex tissues for cell type-specific gene expression profiles. *BMC bioinformatics*, **14**, 89.

Deconvolution Methods		
Name	Citation	
dtangle	(This publication)	Partial
CIBERSORT	(Newman *et al.*, 2015)	Partial
EPIC	(Racle *et al.*, 2017)	Partial
LS Fit	(Abbas *et al.*, 2009)	Partial
Q. Prog	(Gong *et al.*, 2011)	Partial
deconf	(Repsilber *et al.*, 2010)	Full
DSA	(Zhong *et al.*, 2013)	Full
ssFrobenius	(Gaujoux and Seoighe, 2012)	Full
ssKL	(Gaujoux and Seoighe, 2012)	Full

Table S1. Nine deconvolution algorithms we compare.

Name	Citation	Accession	Tech.	Truth	Genes	Samples	Reference (num., source)	Cell Types (num., source)	Species
Shi	MAQC (2006)	GSE5350	ma	mix	54,676	60	60, internal	2, universal, brain	human
Gong	Gong *et al.* (2011)	GSE29832	ma	mix	54,676	9	6, internal	2, blood, breast	human
Shen-Orr	Shen-Orr *et al.* (2010)	GSE19830	ma	mix	31,100	33	9, internal	3, liver, brain, lung	rat
Abbas	Abbas *et al.* (2009)	GSE11058	ma	mix	54,676	12	12, internal	4, leukocytes	human
Becht	Becht *et al.* (2016)	GSE64385	ma	mix	54,676	10	766, external	6, colorectal carcinoma, leukocytes	human
Kuhn	Kuhn *et al.* (2011)	GSE19380	ma	mix	31,100	10	16, internal	4, brain	rat
Newman FL	Newman *et al.* (2015)	GSE65136	ma	cyto.	11,189	14	113, external	12, leukocytes	human
Newman PBMC	Newman *et al.* (2015)	GSE65133	ma	cyto.	11,049	20	113, external	22, leukocytes	human
Parsons	Parsons *et al.* (2015)	PRJEB8231	seq	mix	23,459	30	9, internal	3, brain, liver, muscle	human
Liu	Liu *et al.* (2015)	GSE64098	seq	mix	23,056	24	16, internal	2, adenocarcinoma	human
Linsley	Linsley *et al.* (2014)	GSE60424	seq	cyto.	21421	5	3, external	3, lymphocytes, monocytes, neutrophils	human

Table S2. Benchmark data sets on which we compare deconvolution algorithms. The accession key is for GEO (or in the case of Parsons, ENA). The technology producing the data is either “ma” for microarray or “seq” for RNA-seq. The column “Truth” distinguishes between mixture experiments “mix” or data where the truth is known from flow cytometry “ctyo.” The number of gene expression measurements made by the technology is the column “Genes” and the number of unknown heterogeneous samples deconvolved is the column “Samples.” The column “Reference” lists the number of samples in the reference data along with the designation of “internal” if the pure reference samples were created part and parcel with the mixture experiment or “external” if the reference samples were collected from external data sources (typically GEO). The column “Cell Types” lists the number of cell types in the mixture samples and provides a description of the cell types along with the species from which the cell types come (in the column “Species”).



Scale	Actual Expression (Unobserved)
Linear	mRNA amount
Log	$\log_2$(mRNA amount)

Scale	Measured Expression (Observed)
Linear	gene expression measurement
Log	$\log_2$(gene expression measurement)

Fig. S1: Measured expressions (log or linear) arise from a measurement process on the actual expressions (log or linear).

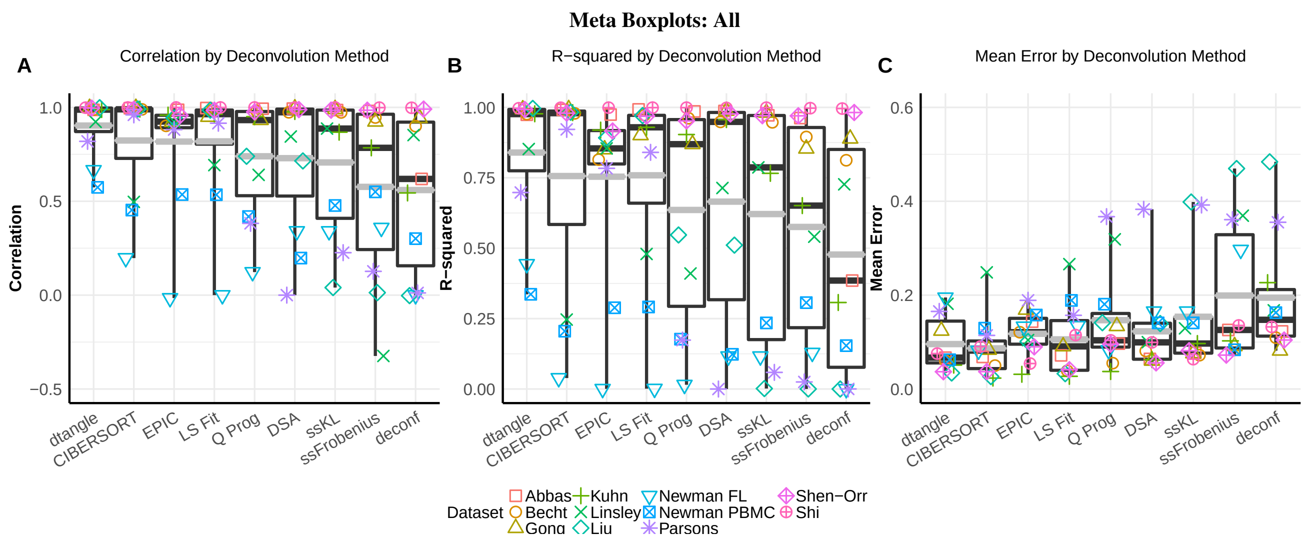


Fig. S2: Boxplots of all deconvolution methods across all data-sets. Top 10% of the 25% of most variable genes are used as marker genes used for deconvolution. Marker genes determined by median differences across reference samples. Slope (γ) for dtangle determined automatically by data-type. (A) For each cell type the correlation of the true mixing proportions against the estimated mixing proportions is calculated. If the s.d. of the estimates is zero, we say the correlation is zero. If the s.d. of the true proportions is zero we do not calculate the correlation. Each point is the median of the correlations across cell types. We calculate this median correlation for each data-set and each deconvolution method. (B) Similar to (A) except using R^2 instead of correlation. (C) is similar to (A) but using grand means instead of correlation. For each cell type the absolute value of the error of the estimated mixing proportions from the true mixing proportions is calculated. Each point is the mean of the errors across cell types. We calculate this mean for each data-set and each deconvolution method.

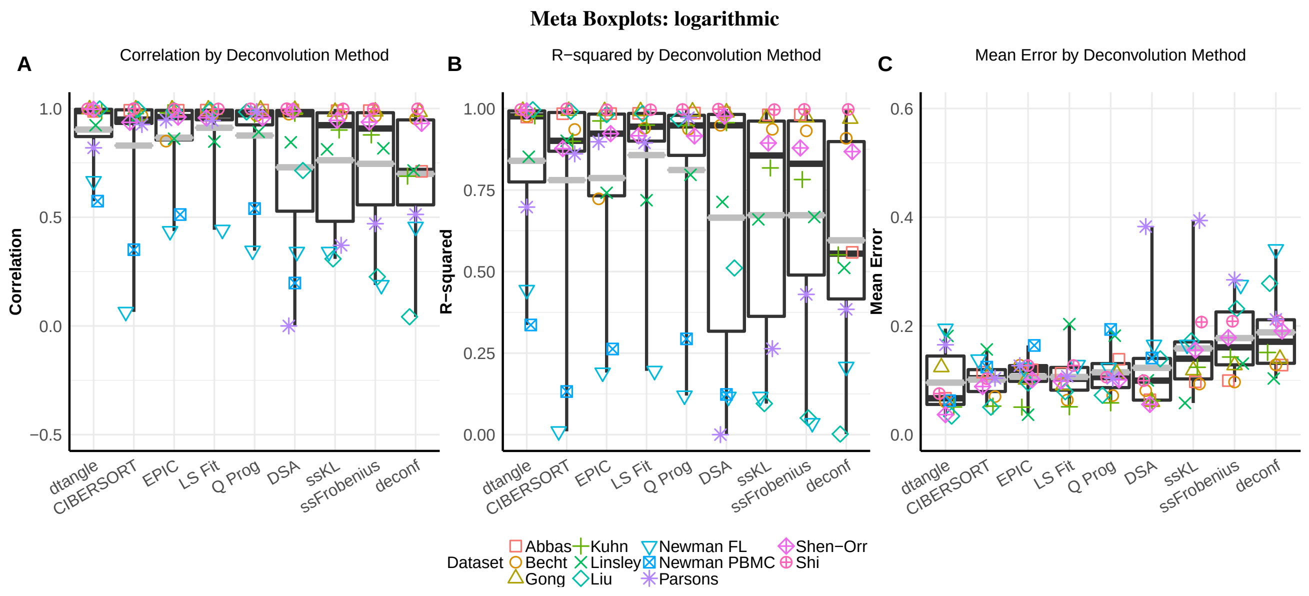


Fig. S3: Similar to Figure S2 but applying methods to log transformed data.

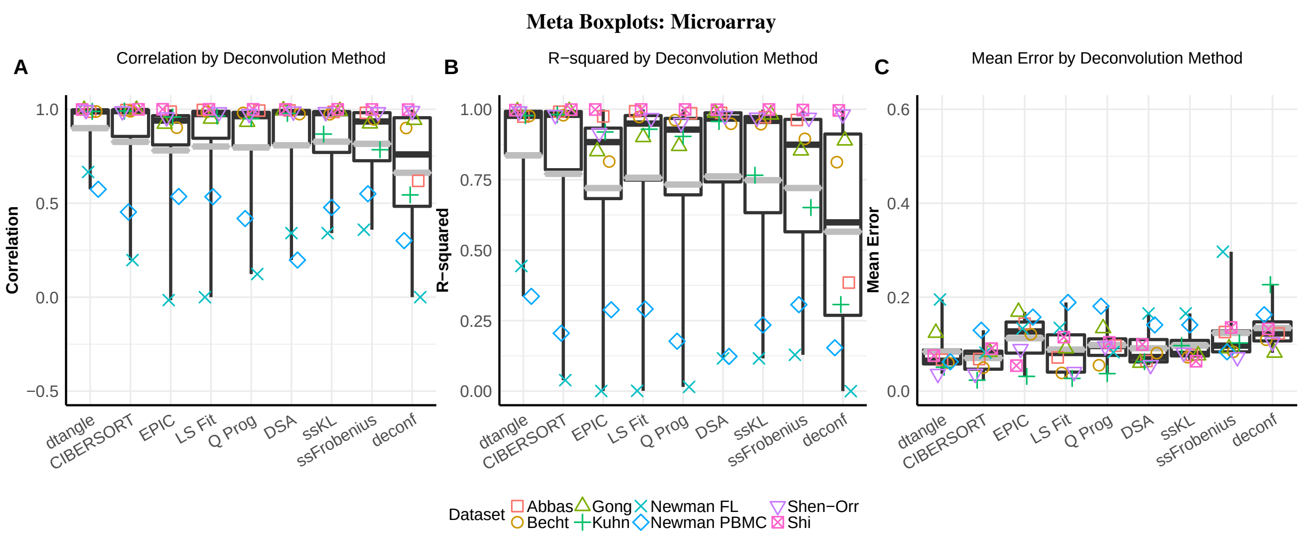


Fig. S4: Similar to Figure S2 but only comparing microarray data-sets.

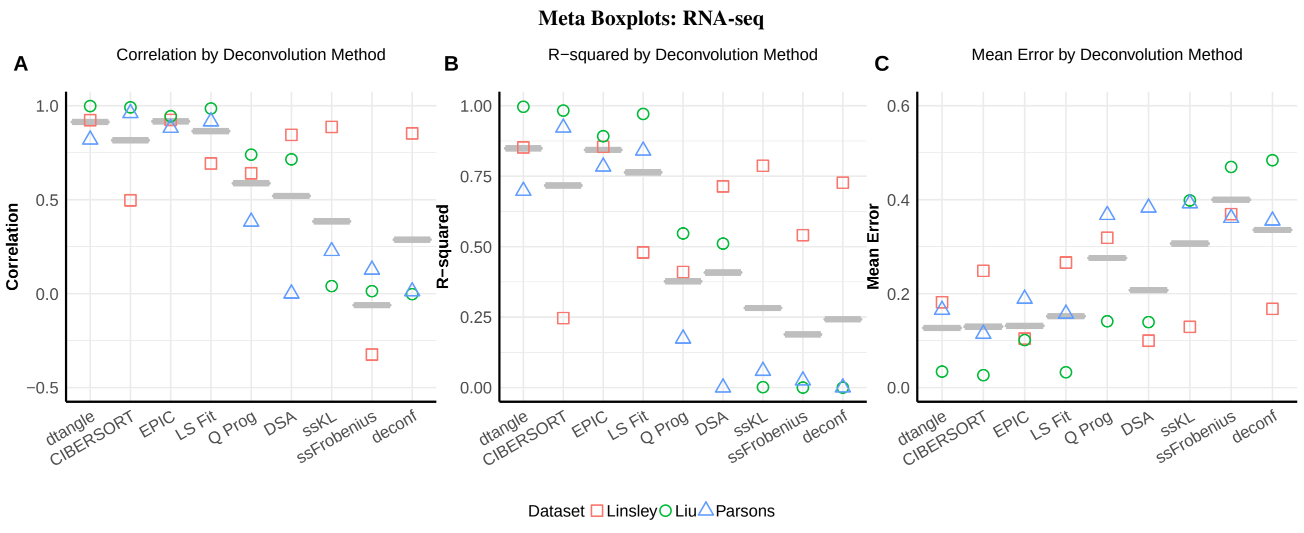


Fig. S5: Similar to Figure S2 but only comparing RNA-seq data-sets.

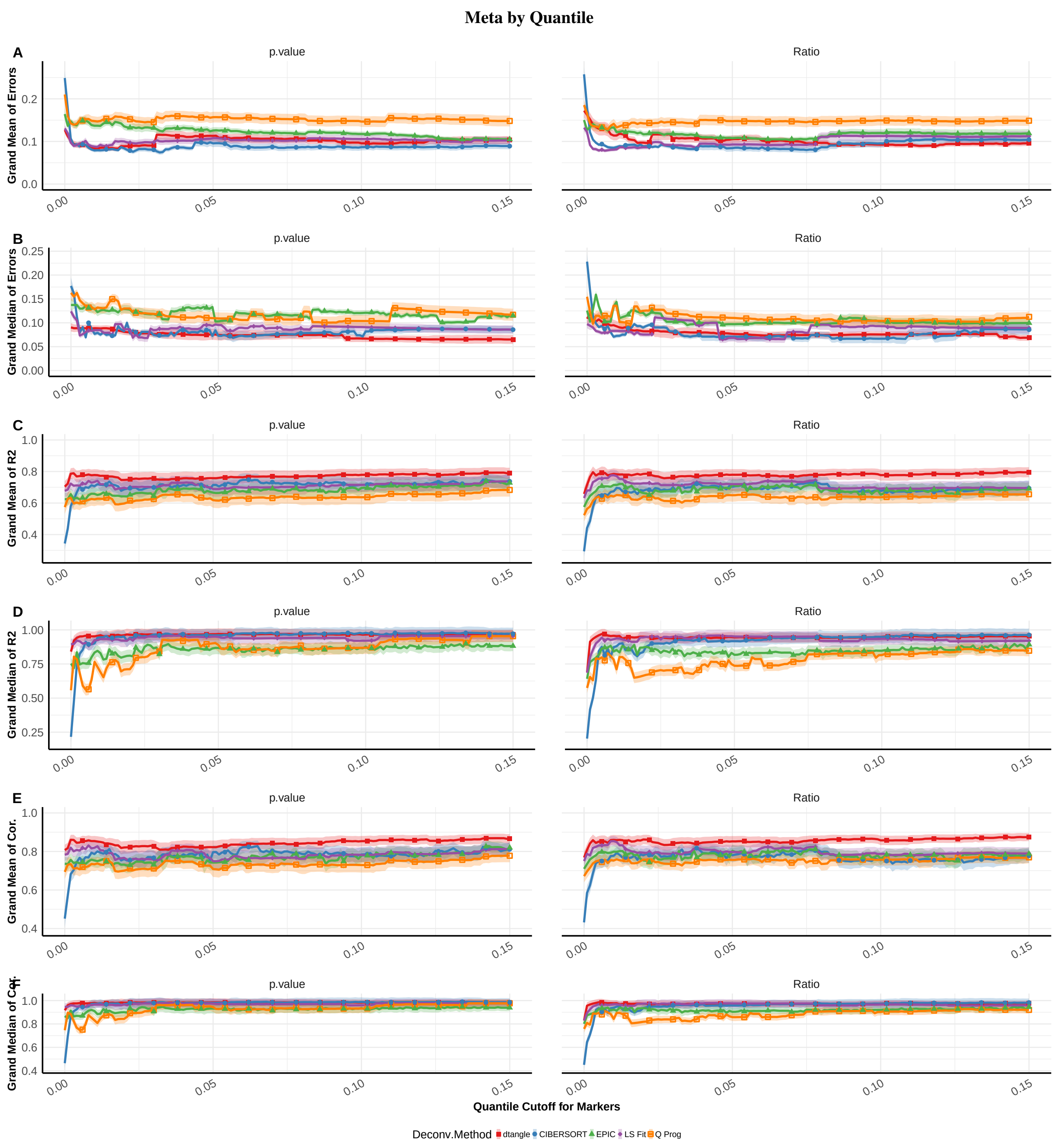


Fig. S6: Partial deconvolution methods performance (y-axis) by number of marker genes (quantile, x-axis). Slope (γ) for dtangle determined automatically by data-type. Top $q\%$ of top 25% of most variable genes used for deconvolution where q varies over the x-axis from 1% to 15% (in increments of 1%). Marker genes determined by p-value (Left) and ratio of the linear expression of each type to the expression in all other types (Right). The y-axis is the grand (A) mean or (B) median (over data-sets and cell types) of the absolute error of the true proportions from the estimated proportions, or the grand (C) mean or (D) median of the R^2 or correlations (E, F) of the estimated proportions against the true proportions. The correlation is zero if the s.d. of the estimates is zero and the correlation is not computed if the s.d. of the true proportions is zero. One line is plotted for each partial deconvolution method. Error ribbons displaying 95% confidence intervals.

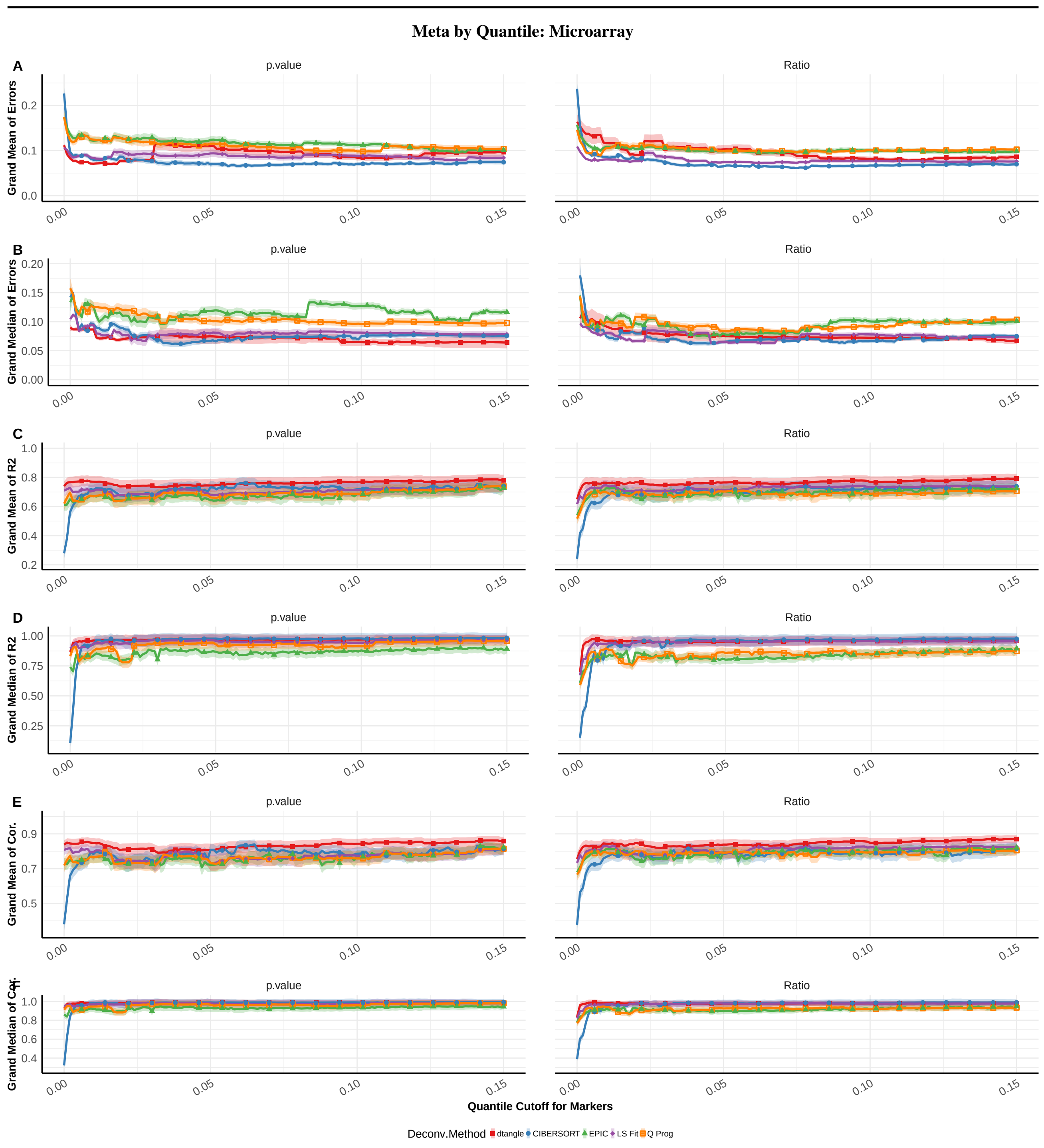


Fig. S7: Similar to Figure S6 except only comparing microarray datasets.

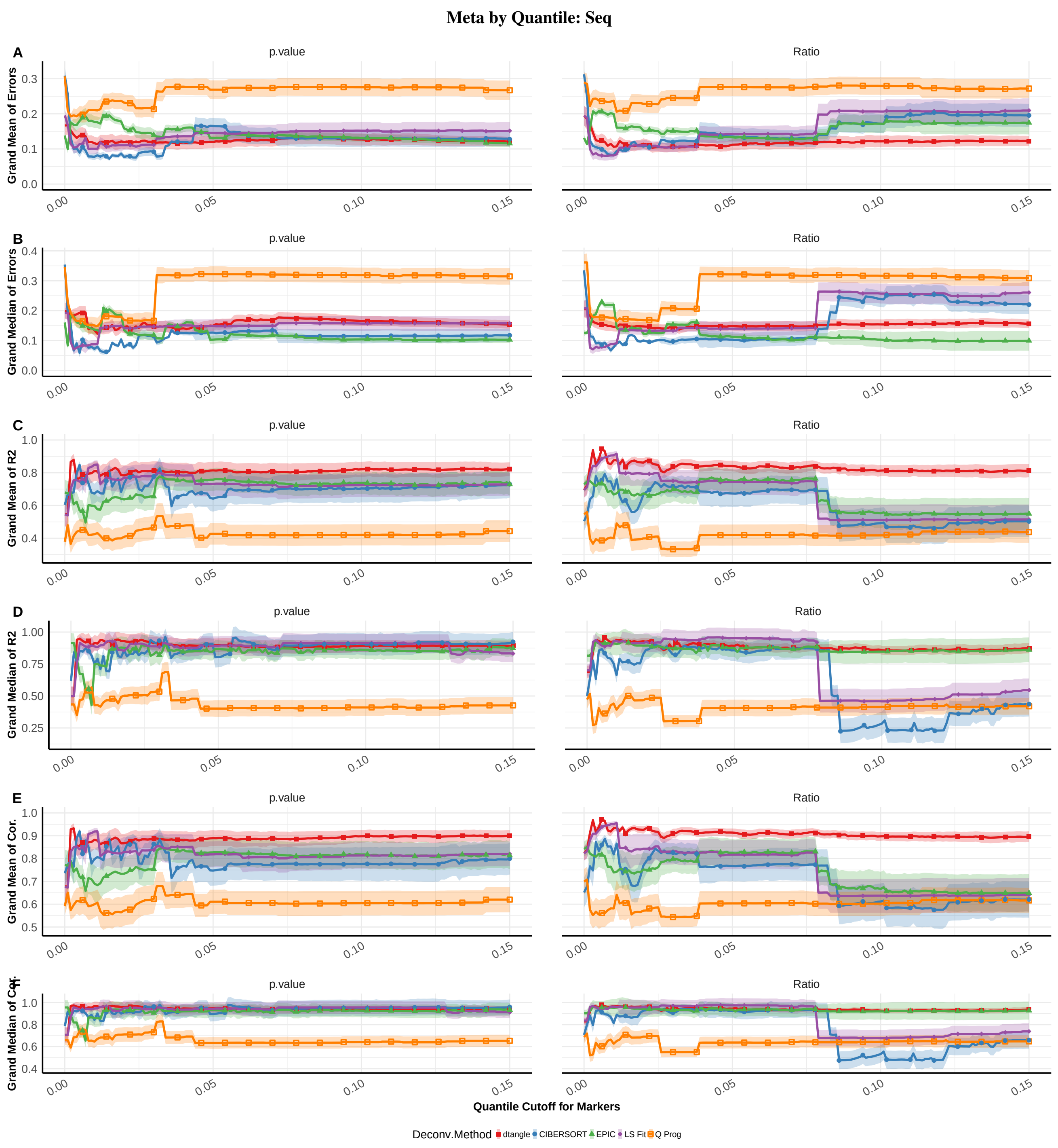


Fig. S8: Similar to Figure S6 except only comparing RNA-seq datasets.

Efficiency by Quantile

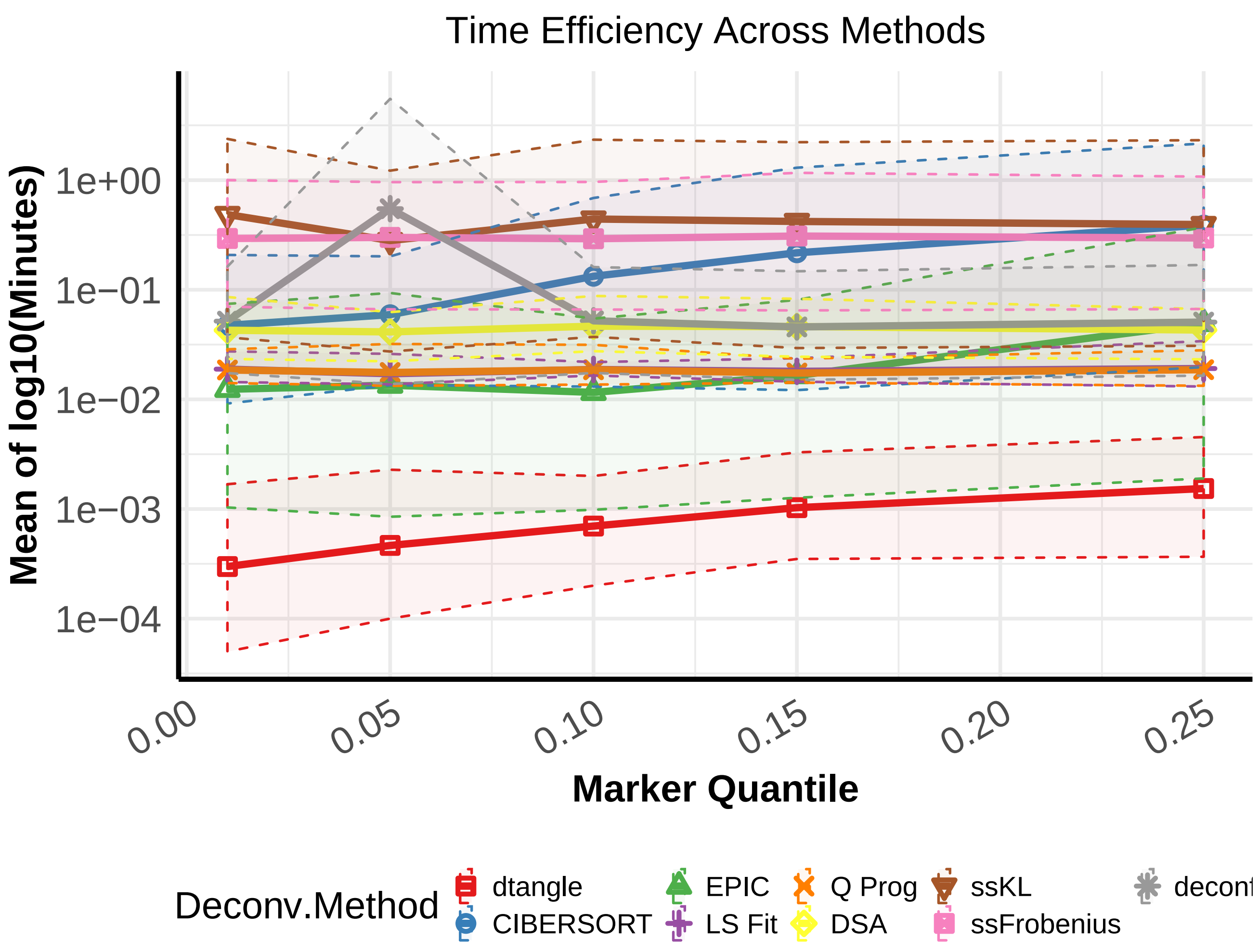


Fig. S9: Mean of $\log_{10}$ of time (in minutes) each algorithm took to deconvolve all data sets. Maximum and minimum value envelope is included.

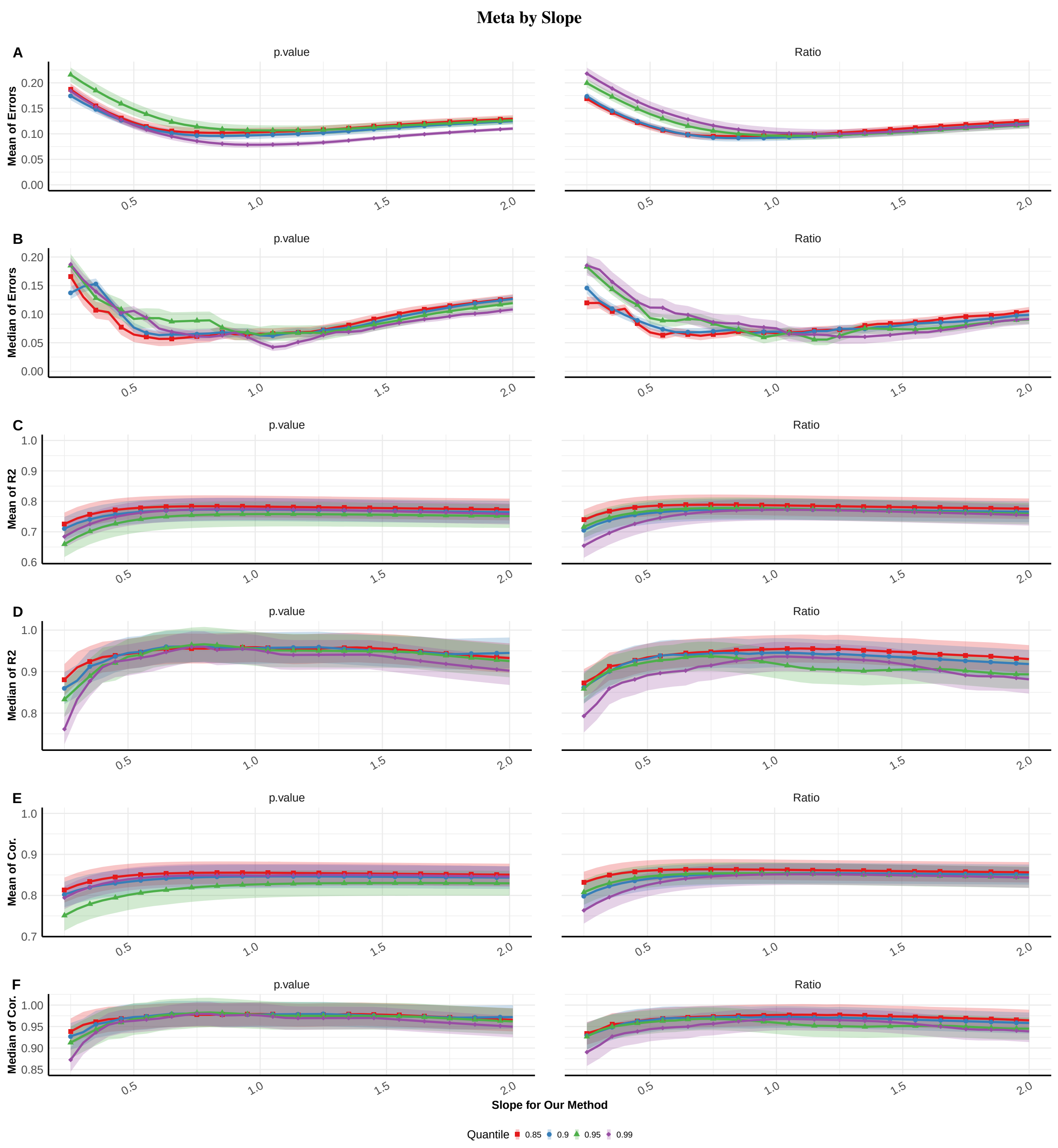


Fig. S10: dtangle performance (y-axis) by slope (γ) varrying over x-axis from 0.25 to 2 (in increments of 0.05). Marker genes determined by p-value (Left) and ratio of the linear expression of each type to the expression in all other types (Right). The y-axis is the grand (A) mean or (B) median (over data-sets and cell types) of the absolute error of the true proportions from the estimated proportions, or the grand (C) mean or (D) median of the correlations of the estimated proportions against the true proportions. The correlation is zero if the s.d. of the estimates is zero and the correlation is not computed if the s.d. of the true proportions is zero. One line is plotted for four choices of number of markers using only the top 1%, 5%, 10% or 15% of top 25% most variables genes as markers. Error ribbons displaying 95% confidence intervals.

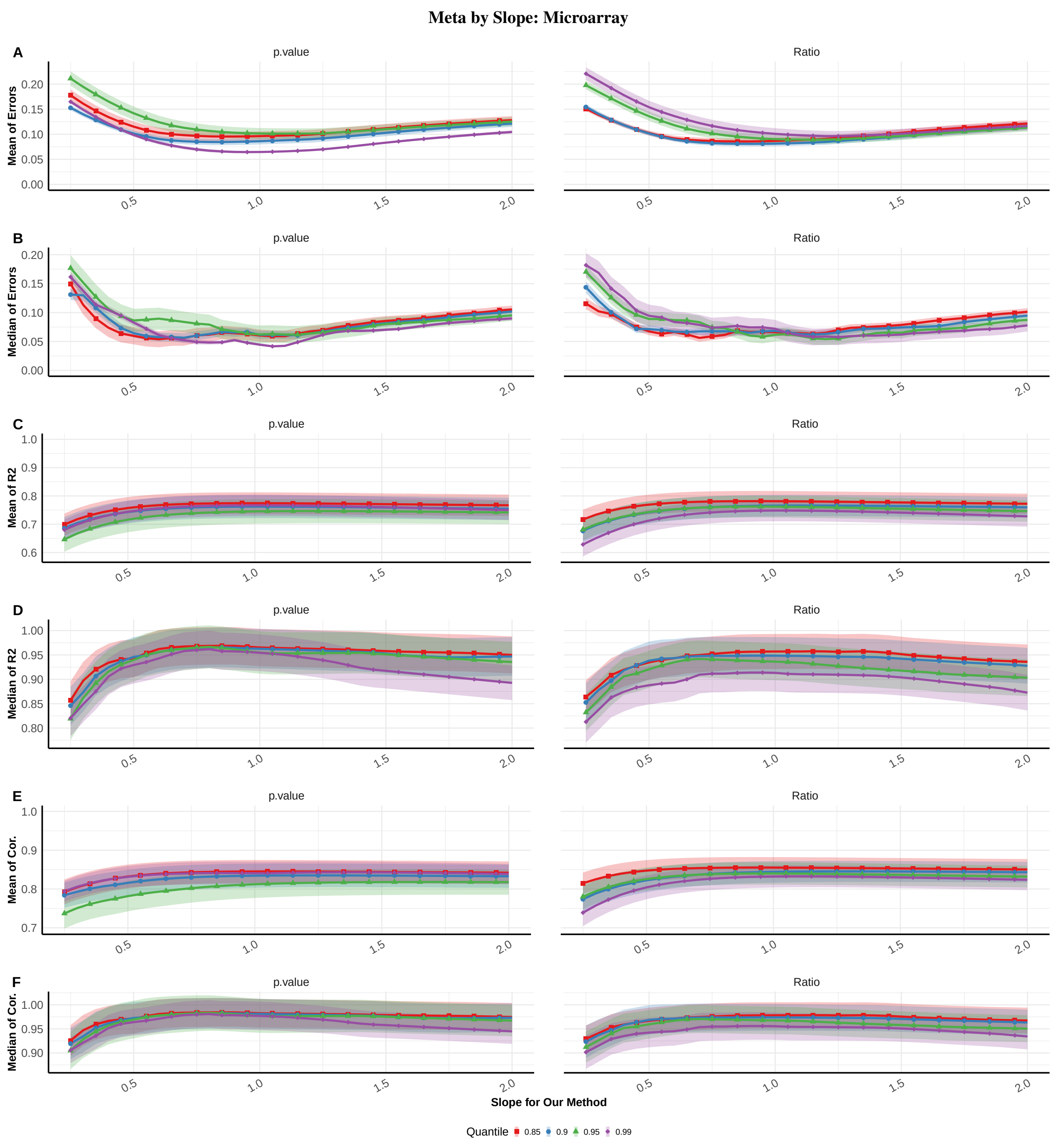


Fig. S11: Similar to Figure S10 but only comparing microarray data-sets.

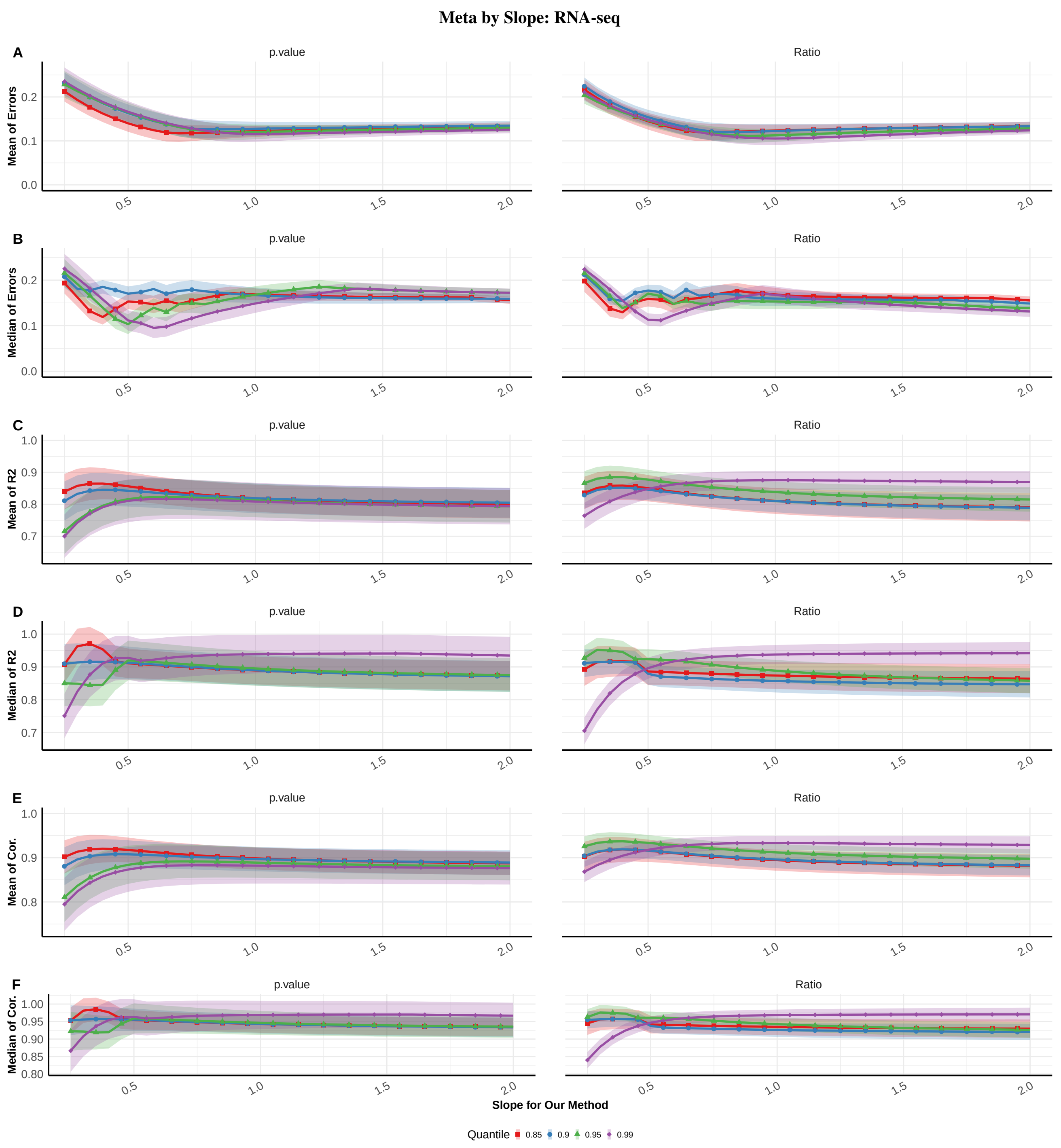


Fig. S12: Similar to Figure S10 but only comparing RNA-seq data-sets.

Dataset: Abbas

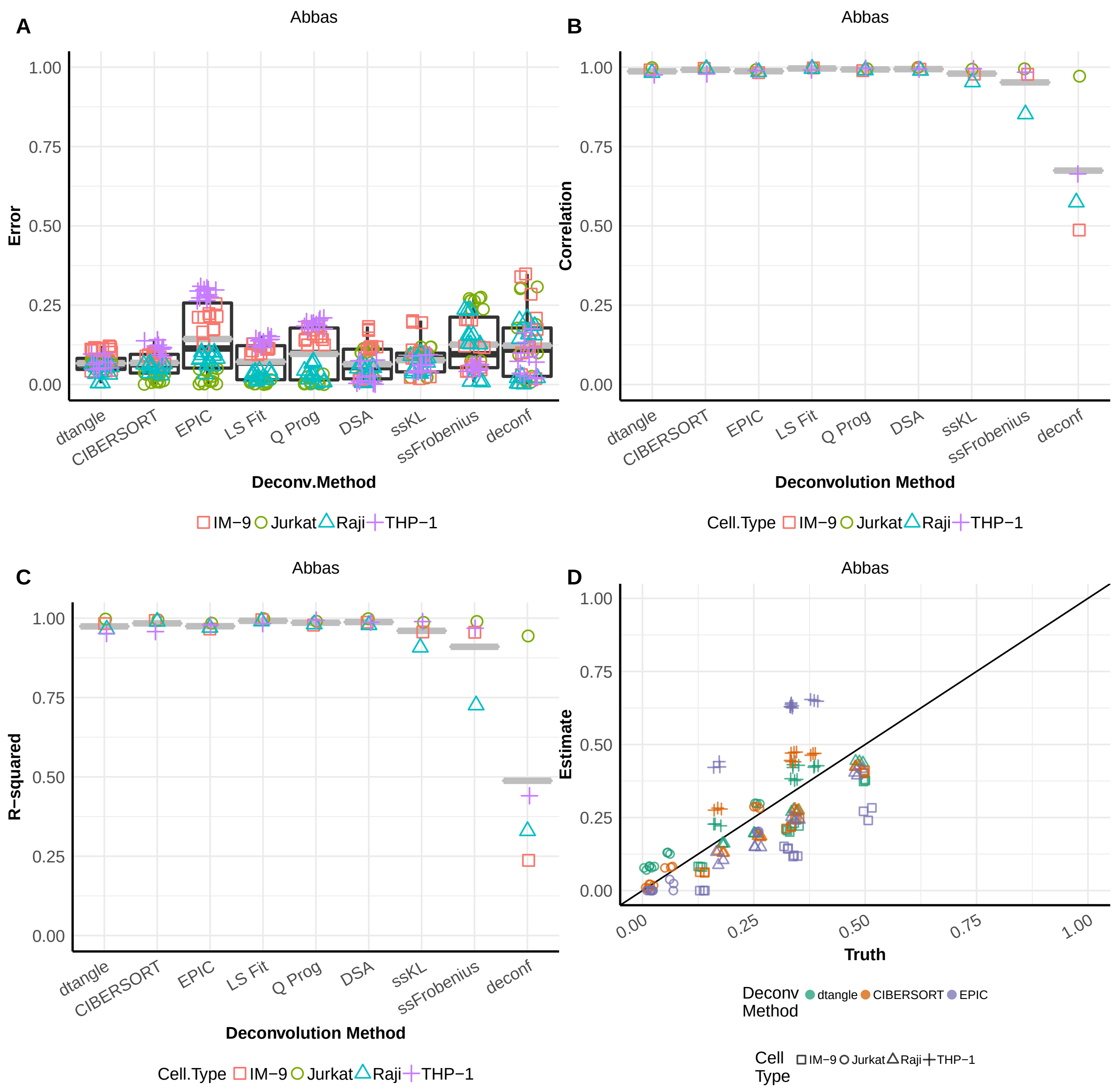


Fig. S13: Deconvolution methods performance on Abbas data-set. Slope (γ) for dtangle determined automatically by data-type. Top 10% of marker genes among the 25% most variable genes are used for deconvolution. Marker genes determined by median differences across reference samples. (A) Boxplots of error for each algorithm. y-axis is the absolute value of the error of the estimates from the true mixing proportions. Black line is the median absolute error, grey line is the mean absolute error. (B) Boxplots of correlation. For each cell type the correlation of the true mixing proportions against the estimated mixing proportions is calculated. If the s.d. of the estimates is zero, we say the correlation is zero. If the s.d. of the true proportions is zero we do not calculate the correlation. (C) Simlar to (B) but using R^2 instead of correlation. (D) Scatter plots of estimated mixing proportions again true mixing proportions for dtangle, CIBERSORT and EPIC. Orange line is a 45° line through zero.

Dataset: Becht

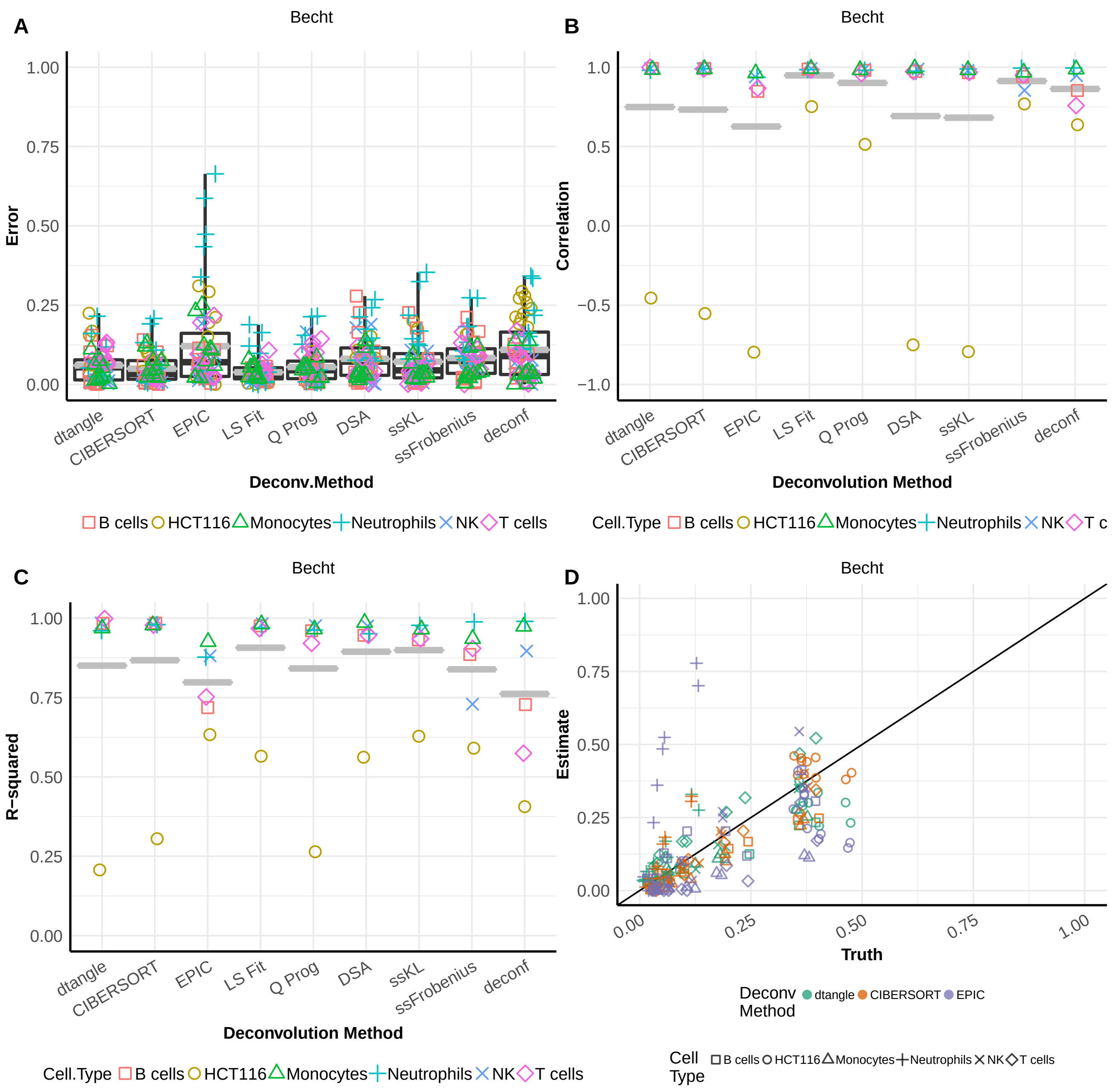


Fig. S14: Similar to Figure S13 but for the Becht data-set.

Dataset: Gong

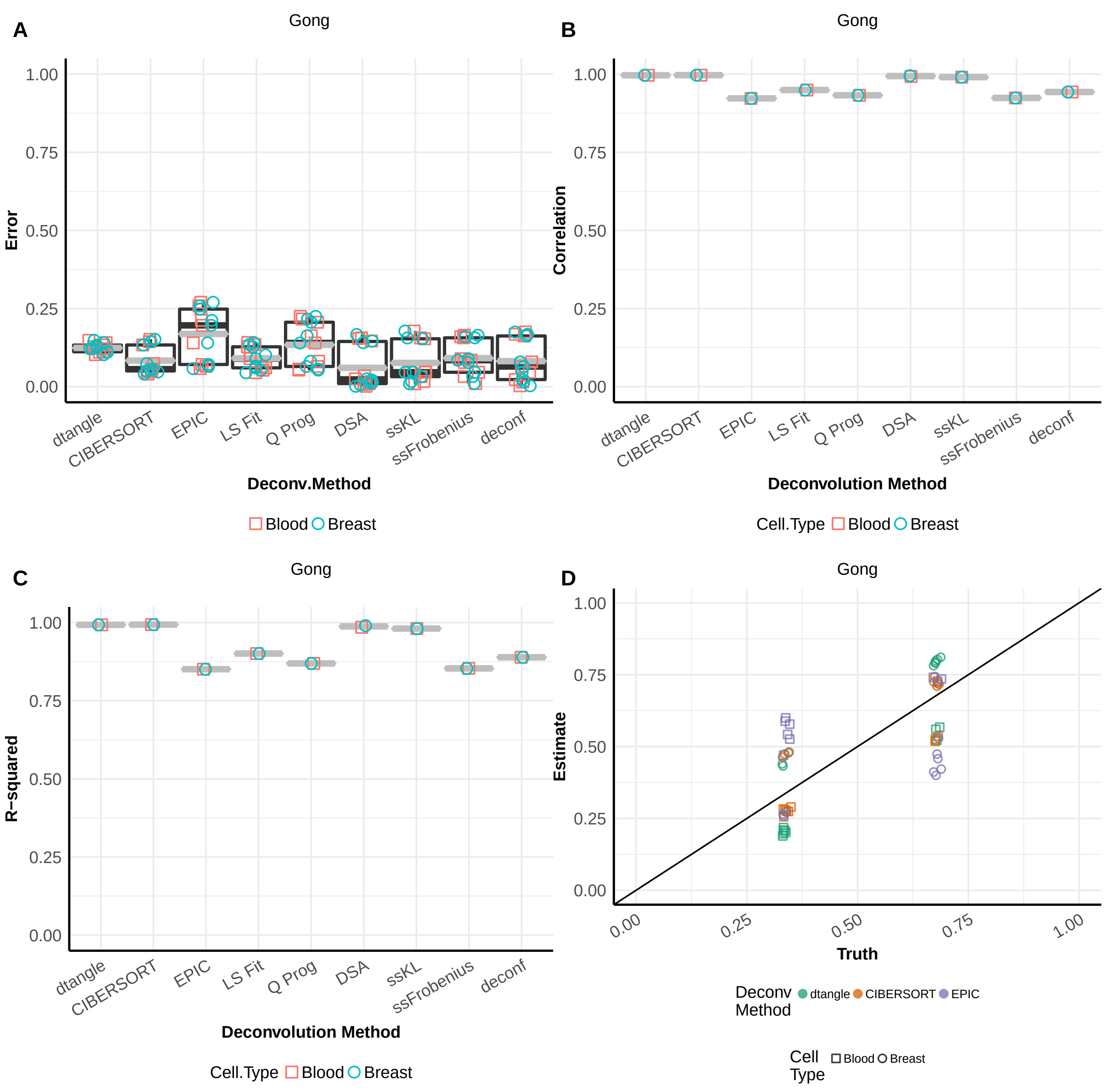


Fig. S15: Similar to Figure S13 but for the Gong data-set.

Dataset: Kuhn

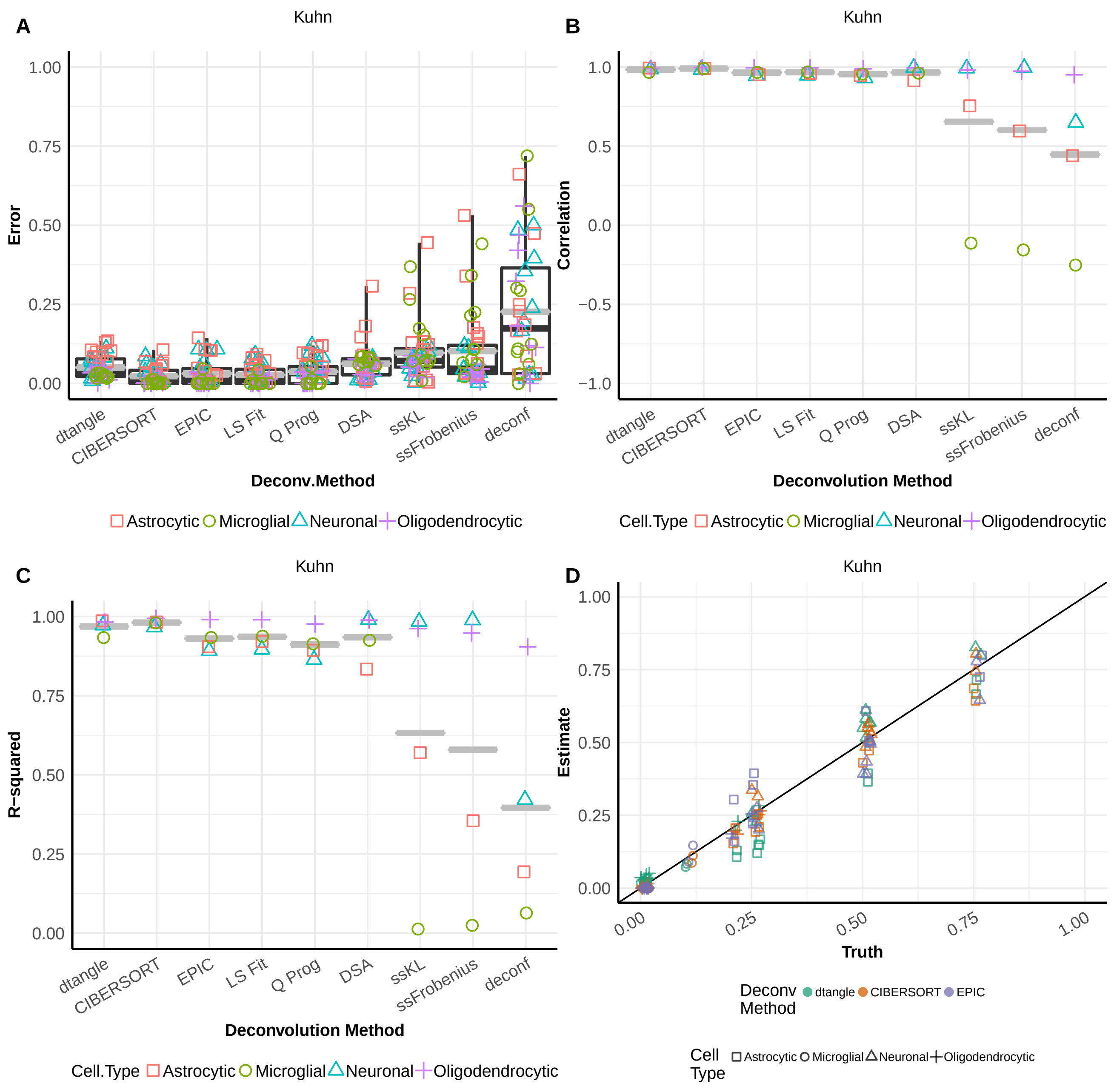


Fig. S16: Similar to Figure S13 but for the Kuhn data-set.

Dataset: Linsley

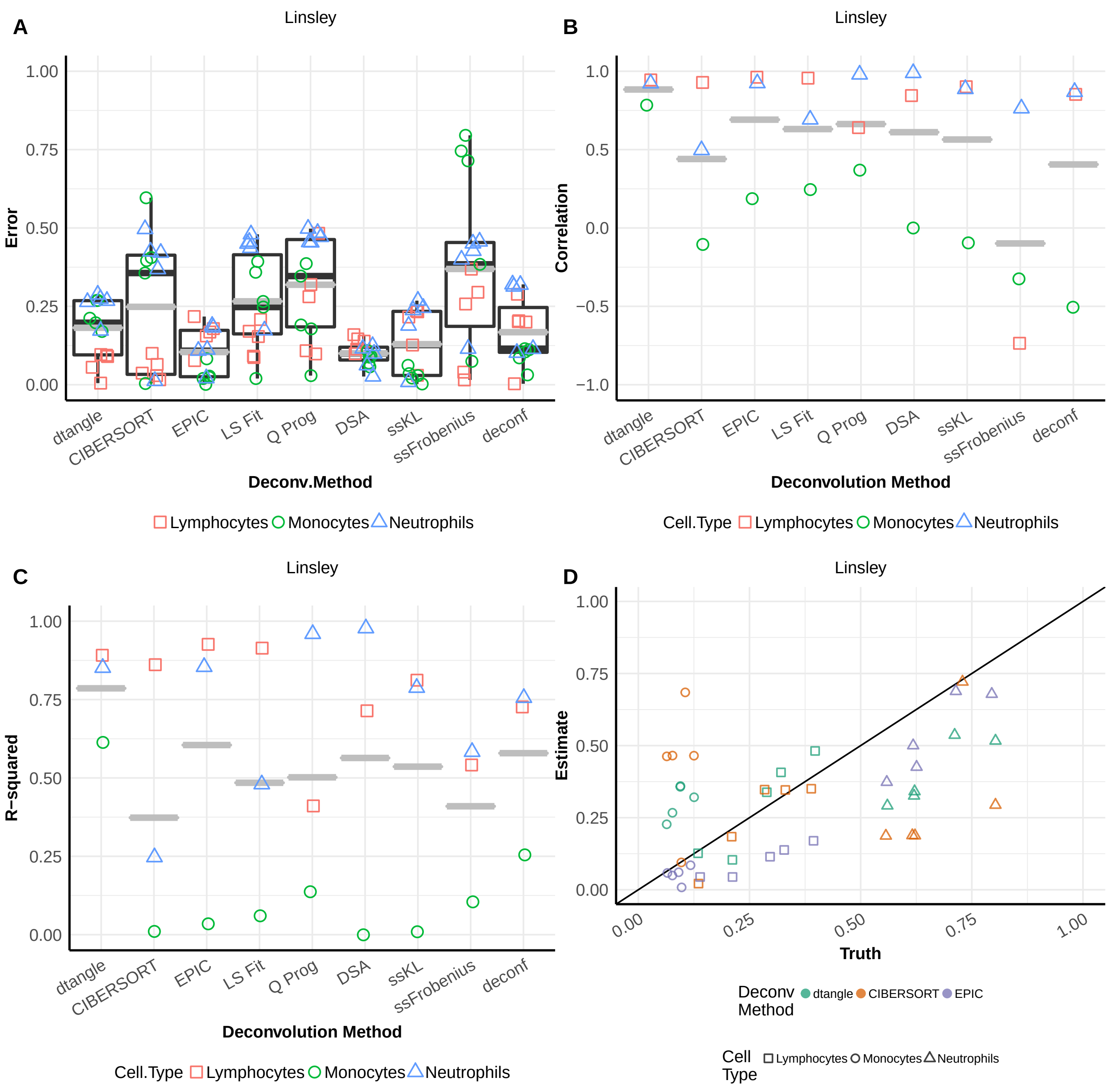


Fig. S17: Similar to Figure S13 but for the Linsley data-set.

Dataset: Liu

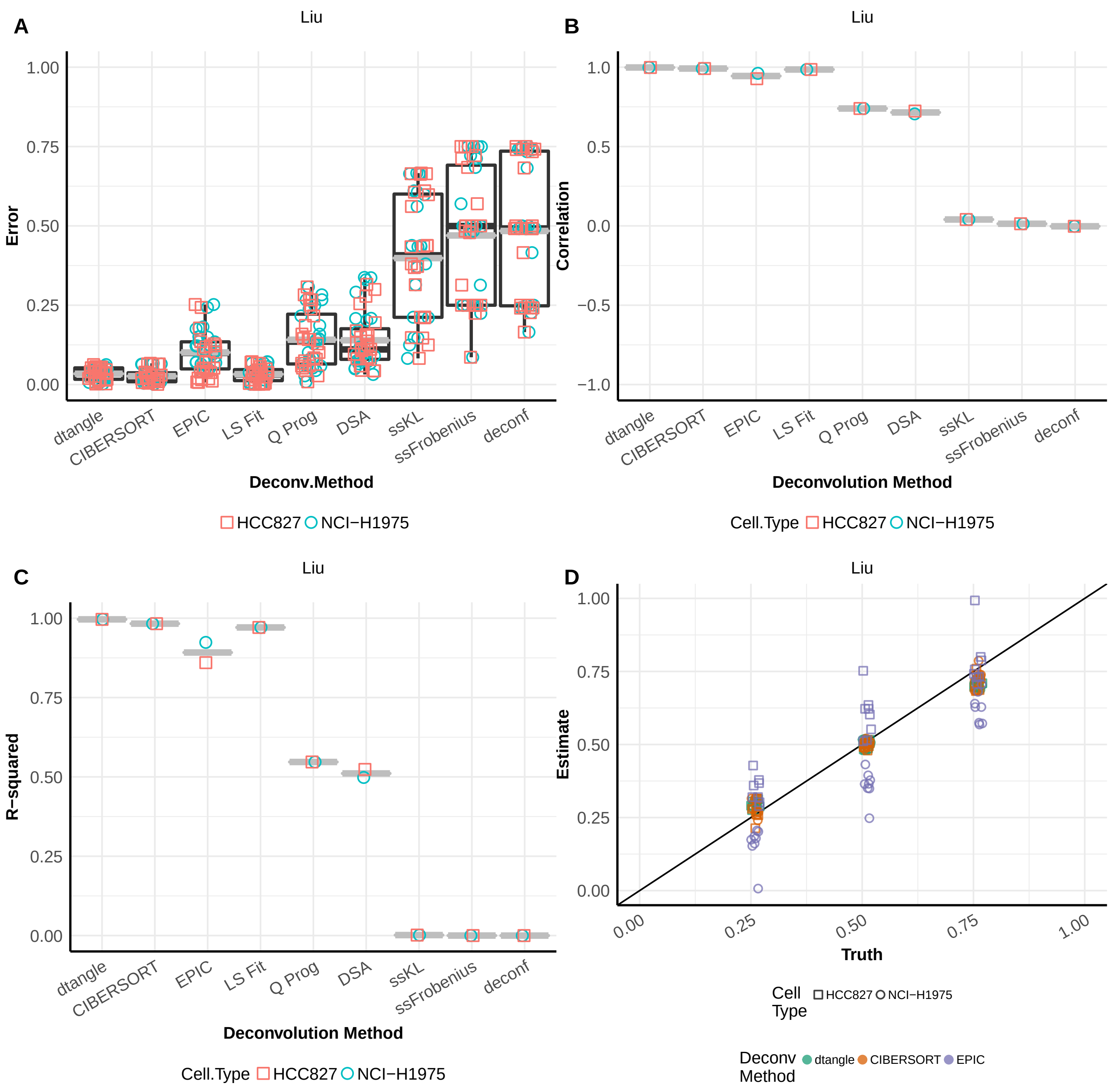


Fig. S18: Similar to Figure S13 but for the Liu data-set.

Dataset: Newman PBMC

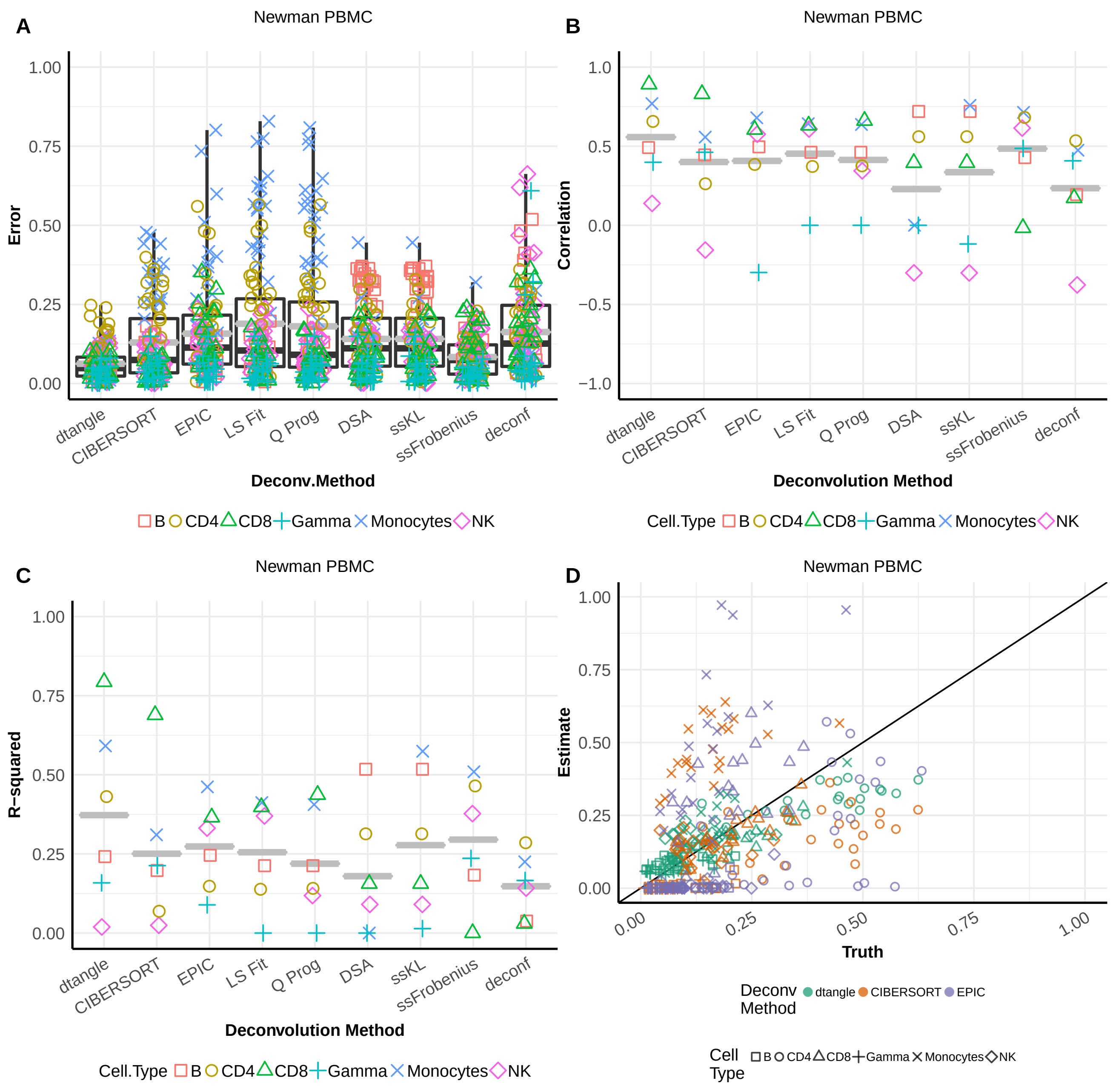


Fig. S19: Similar to Figure S13 but for the Newman PBMC data-set.

Dataset: Newman FL

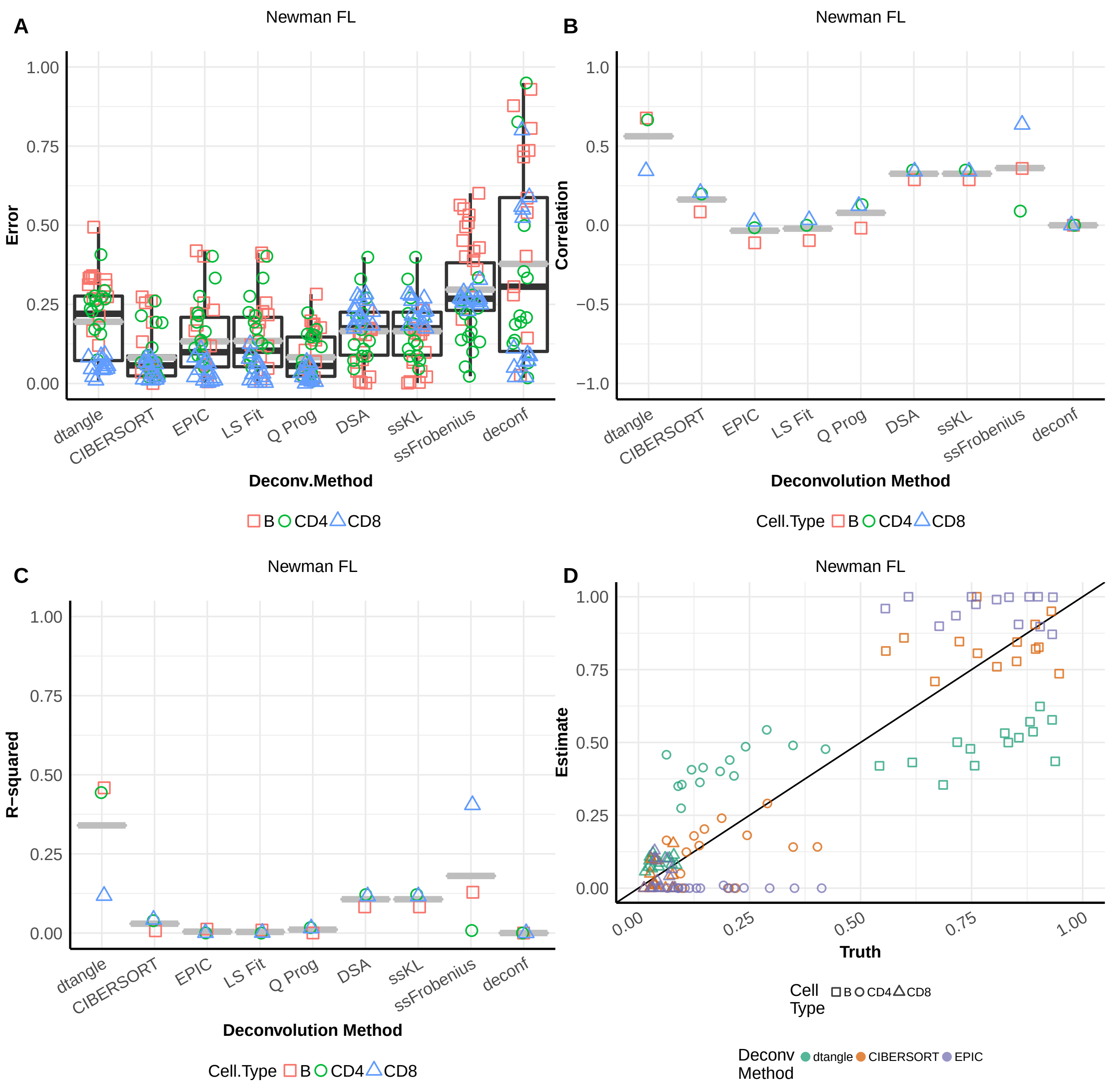


Fig. S20: Similar to Figure S13 but for the Newman FL data-set.

Dataset: Parsons

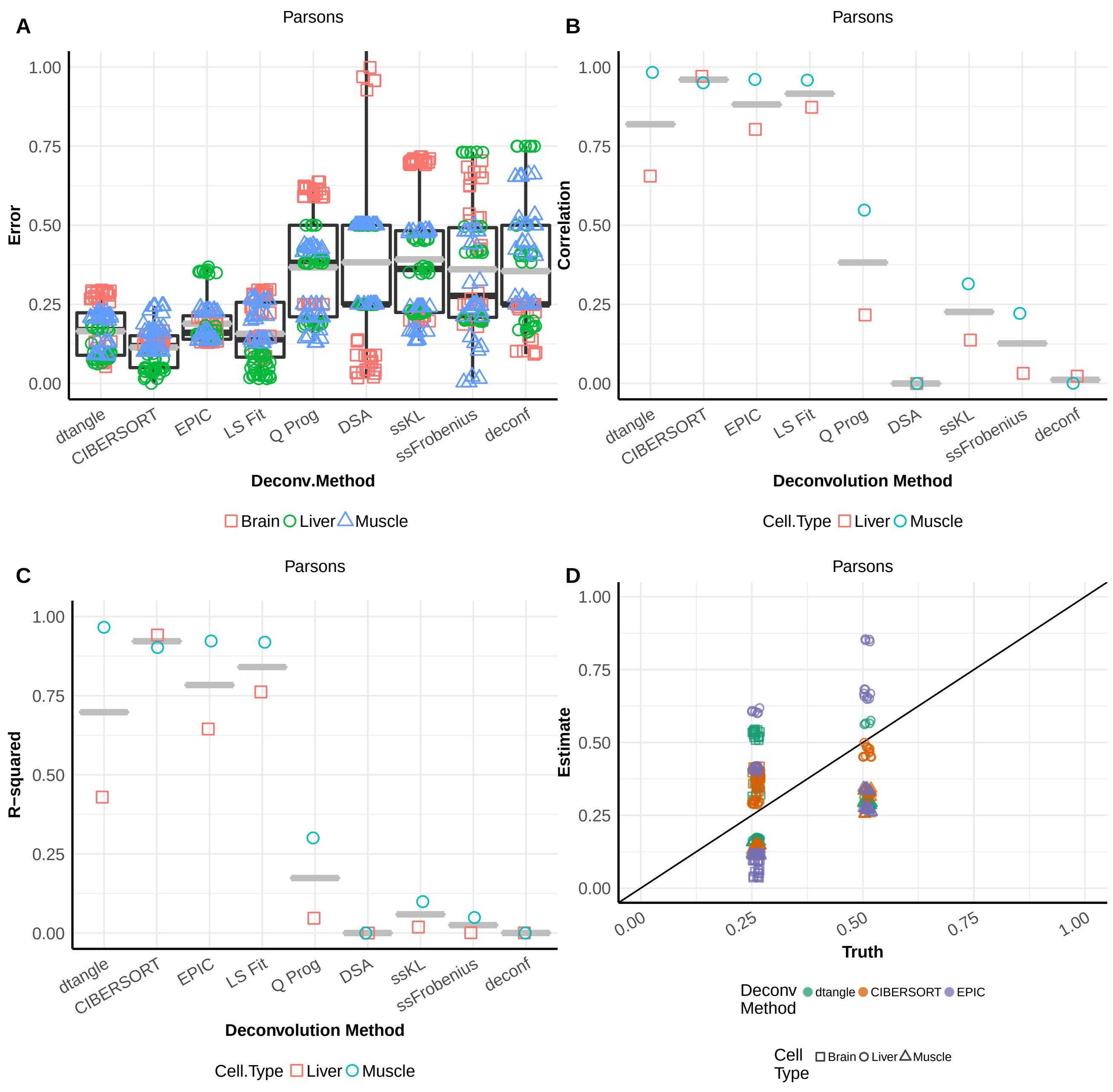


Fig. S21: Similar to Figure S13 but for the Parsons data-set.

Dataset: Shen-Orr

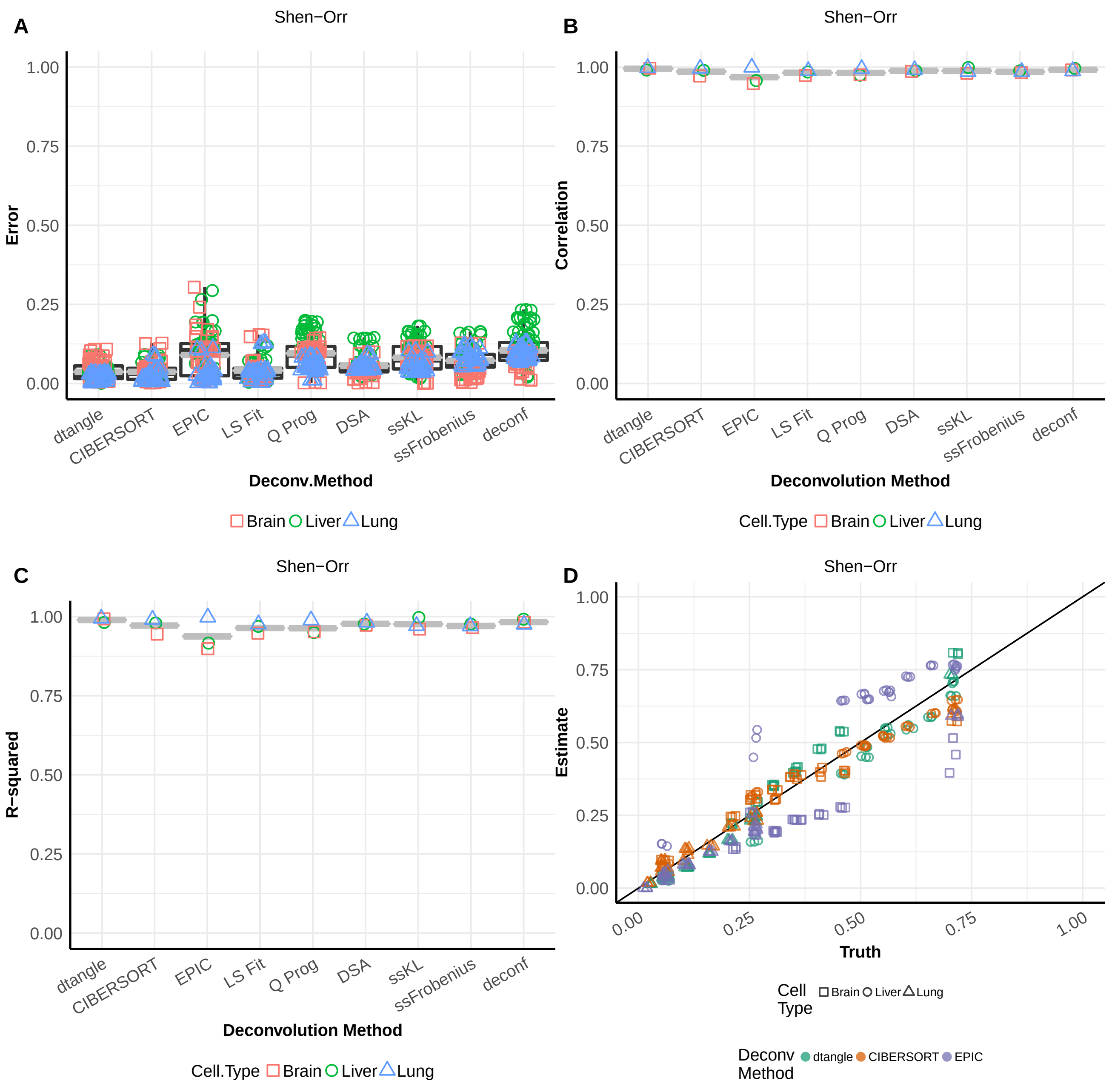


Fig. S22: Similar to Figure S13 but for the Shen-Orr data-set.

Dataset: Shi

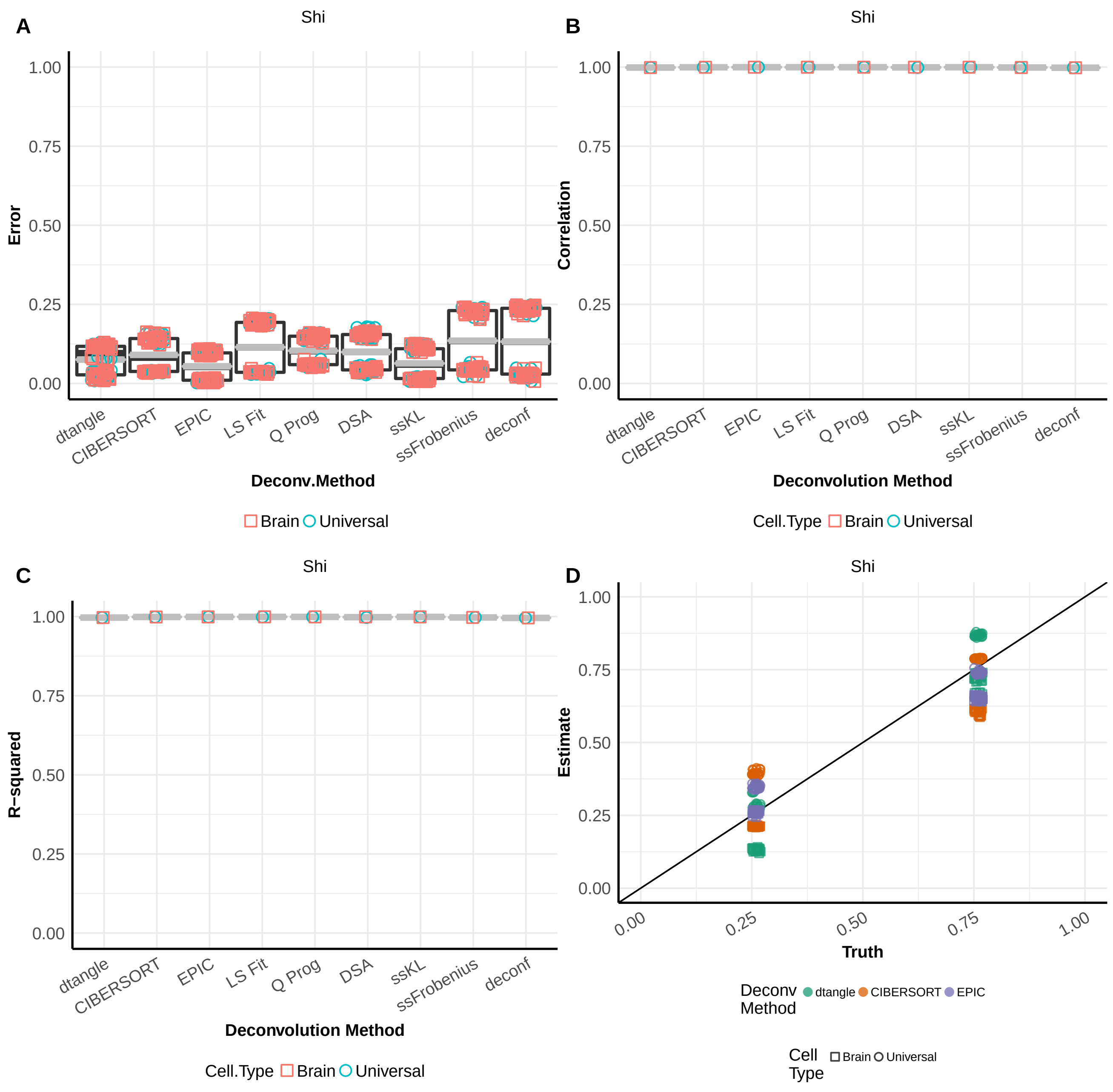


Fig. S23: Similar to Figure S13 but for the Shi data-set.

Dataset: Newman PBMC

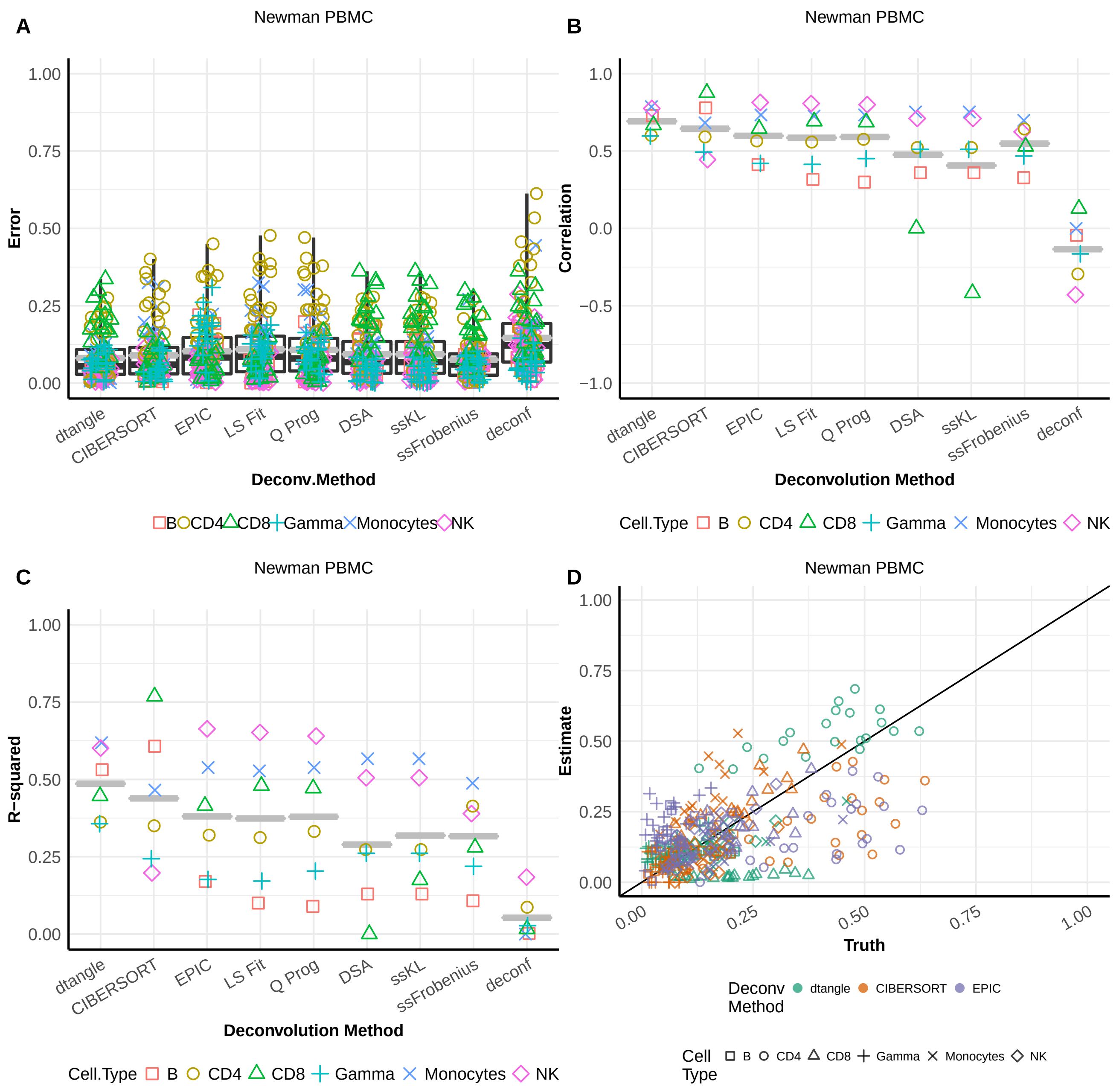


Fig. S24: Same as Figure S19 but using references, mixtures samples, and marker genes directly from Newman paper supplement.

Dataset: Newman FL

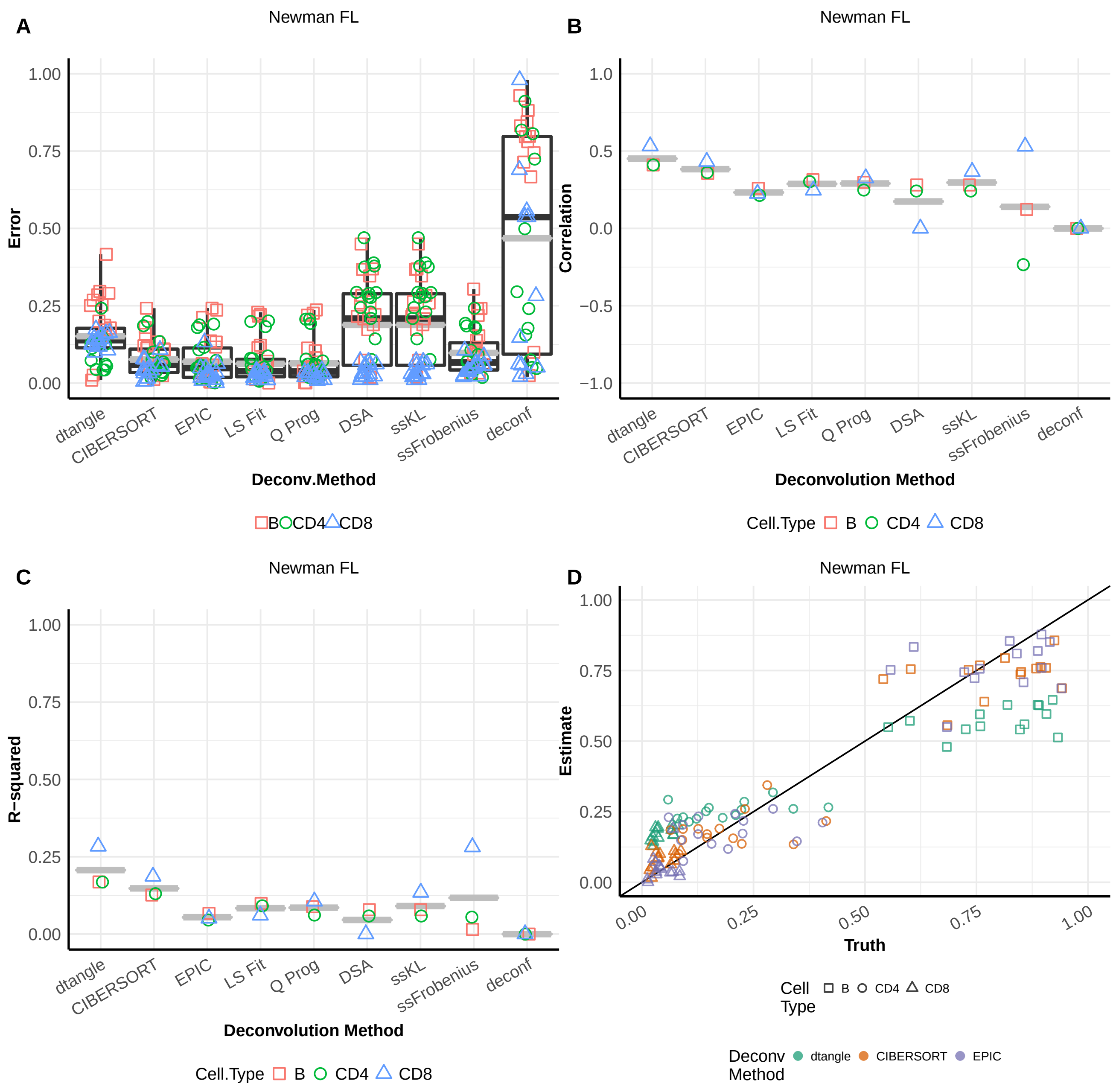


Fig. S25: Same as Figure S20 but using references and marker genes directly from Newman paper supplement.

Lyme Disease Example

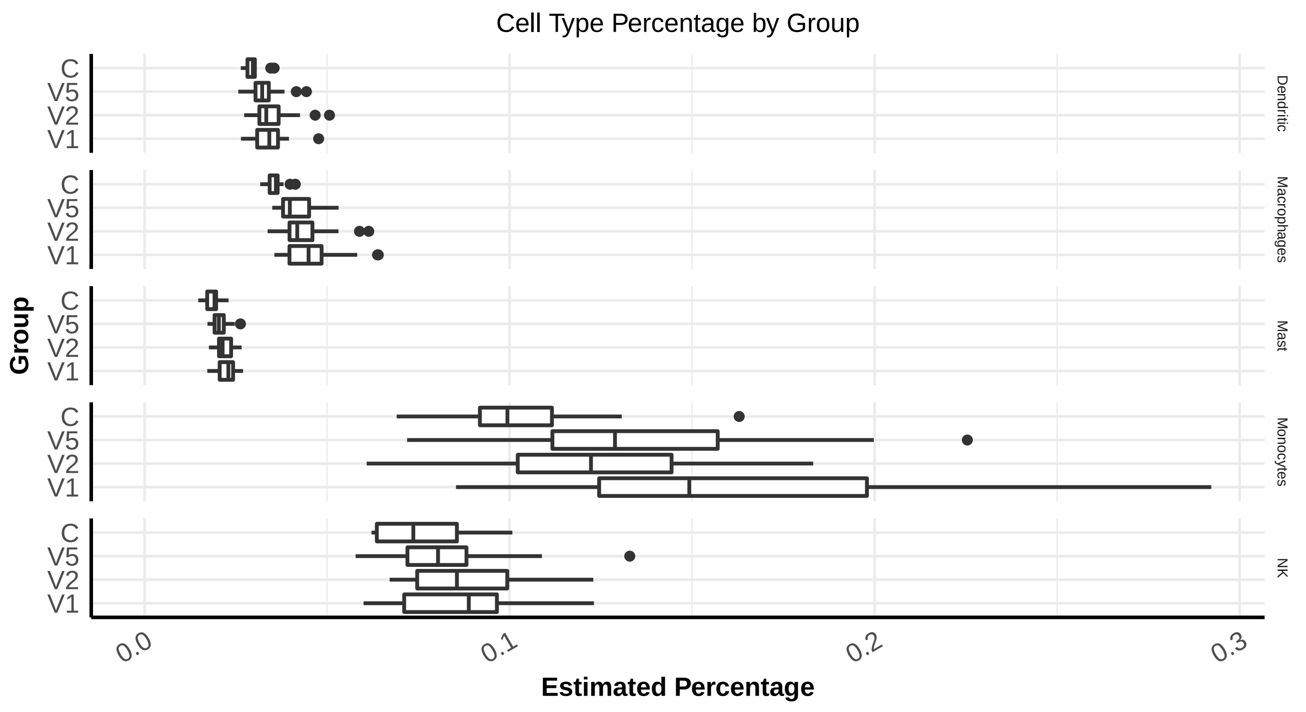
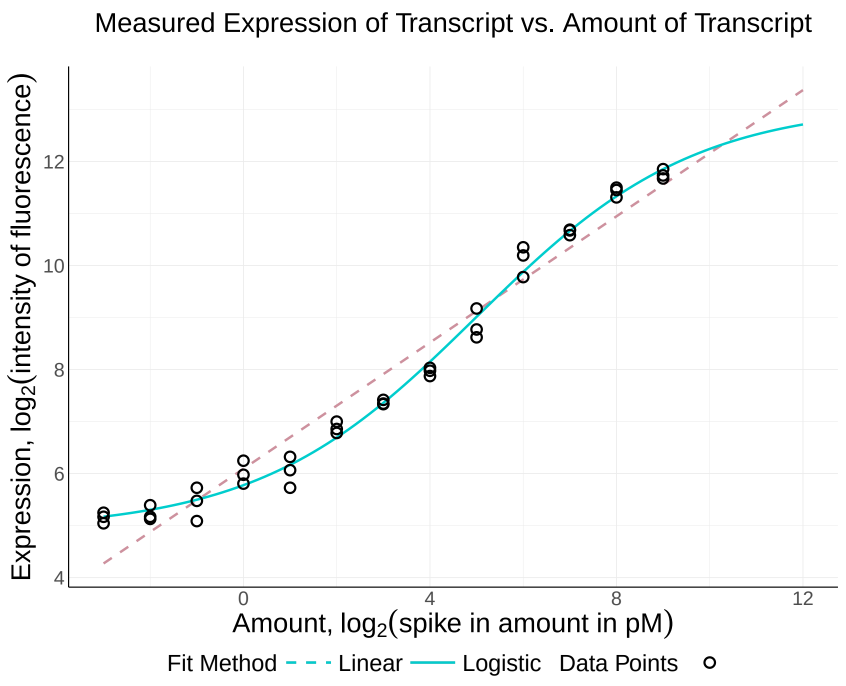
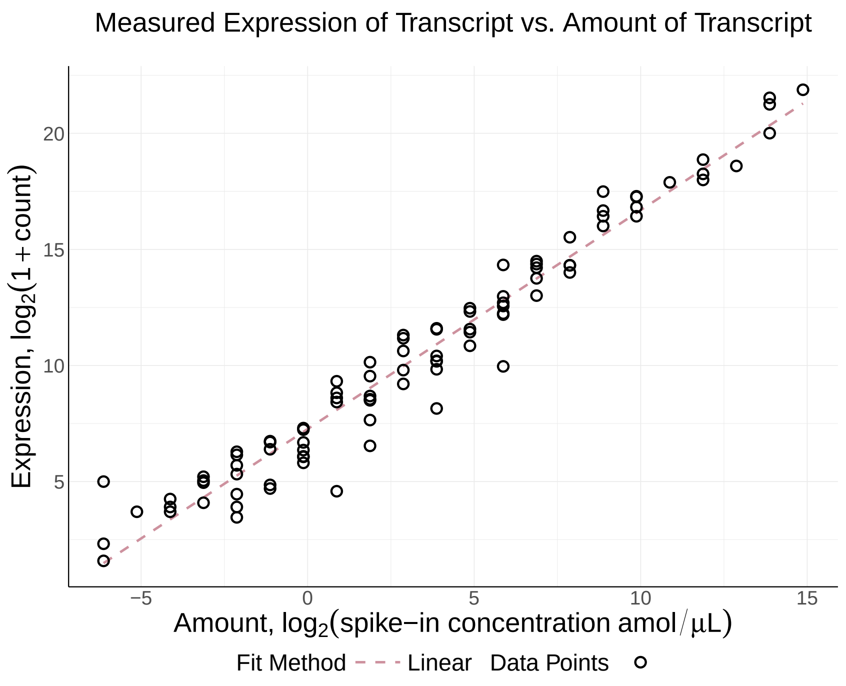


Fig. S26: Estimated cell type proportions over time.



(a) $\log_2$ measured expression versus $\log_2$ concentration of a probe for gene TNFRSF1B. The relationship is approximated well by a linear model. While we have plotted amount against measured expression for one particular gene the results are generalizable to all genes. Points are plotted for the 13 experiments where the gene is spiked-in at a amount above zero and for each of the three technical replicates of each experiment. Along with the data points are plotted a linear and logistic least squares fit. The linear fit is a simple linear regression of measured expression on amount and the logistic fit is the least squares fit of a generalized logistic function of the form $\beta_0 + \beta_1 / (1 + \exp(\beta_2 x + \beta_3))$.



(b) $\log_2$ measured expression versus $\log_2$ concentration of ERCC spike-in controls in RNA-seq data. This relationship is highly linear. A linear least squares regression fit is plotted as a line.

Fig. S27

Simulation: Artificial Cell Type with Low Gaussian Error

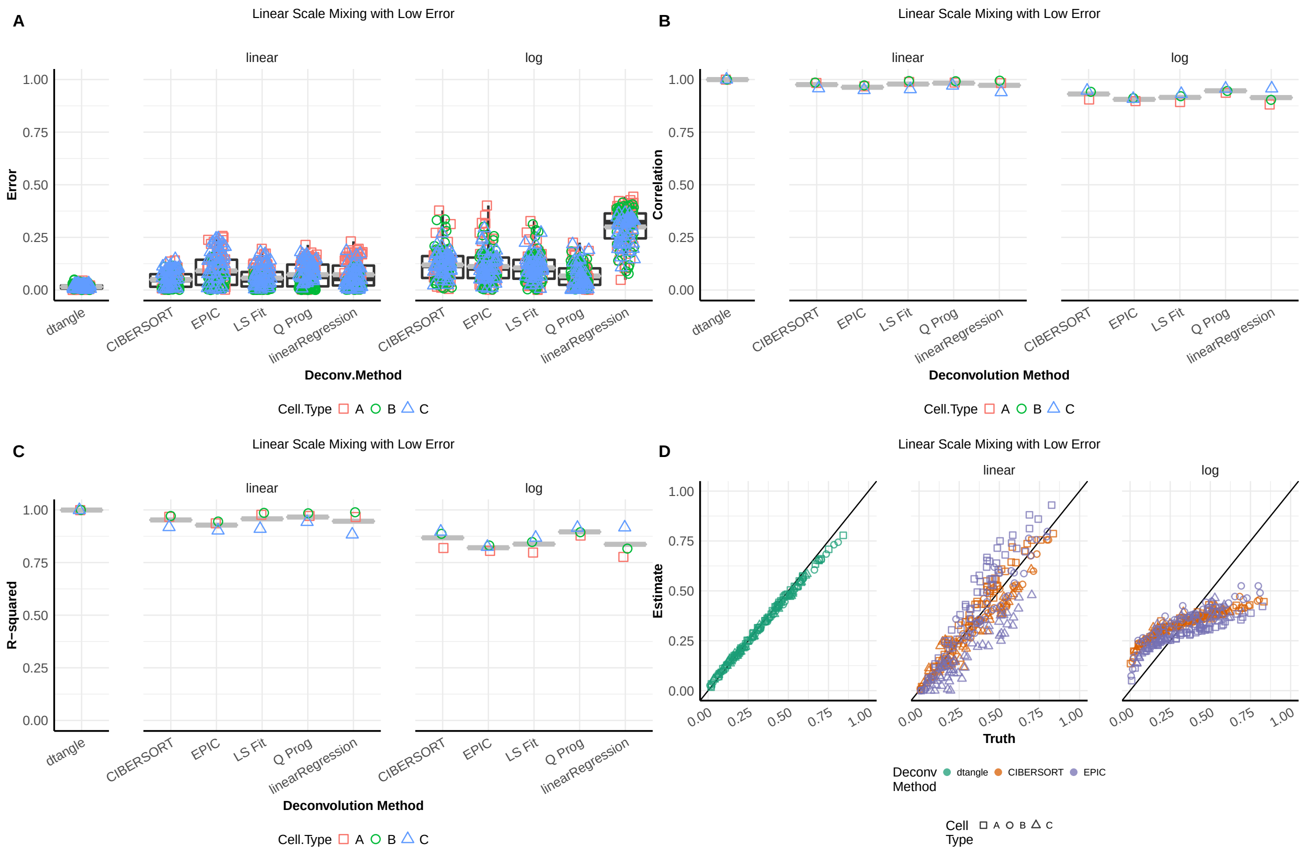


Fig. S28: Partial deconvolution methods performance on simulated gaussian data with **low error**. Computation for methods other than dtangle was done for data both on the $\log_2$ scale and the linear un-transformed scale. Slope (γ) for dtangle is set to one. Top 10% of 25% most variable genes used for deconvolution. Marker genes determined by median differences across reference samples. (A) Boxplots of error for each algorithm. y-axis is the absolute value of the error of the estimates from the true mixing proportions. Black line is the median absolute error, grey line is the mean absolute error. (B) Boxplots of correlation. For each cell type the correlation of the true mixing proportions against the estimated mixing proportions is calculated. If the s.d. of the estimates is zero, we say the correlation is zero. If the s.d. of the true proportions is zero we do not calculate the correlation. (C) Scatter plots of estimated mixing proportions again true mixing proportions for dtangle, CIBERSORT and EPIC. Orange line is a $45°$ line through zero.

Simulation: Artificial Cell Type with High Gaussian Error

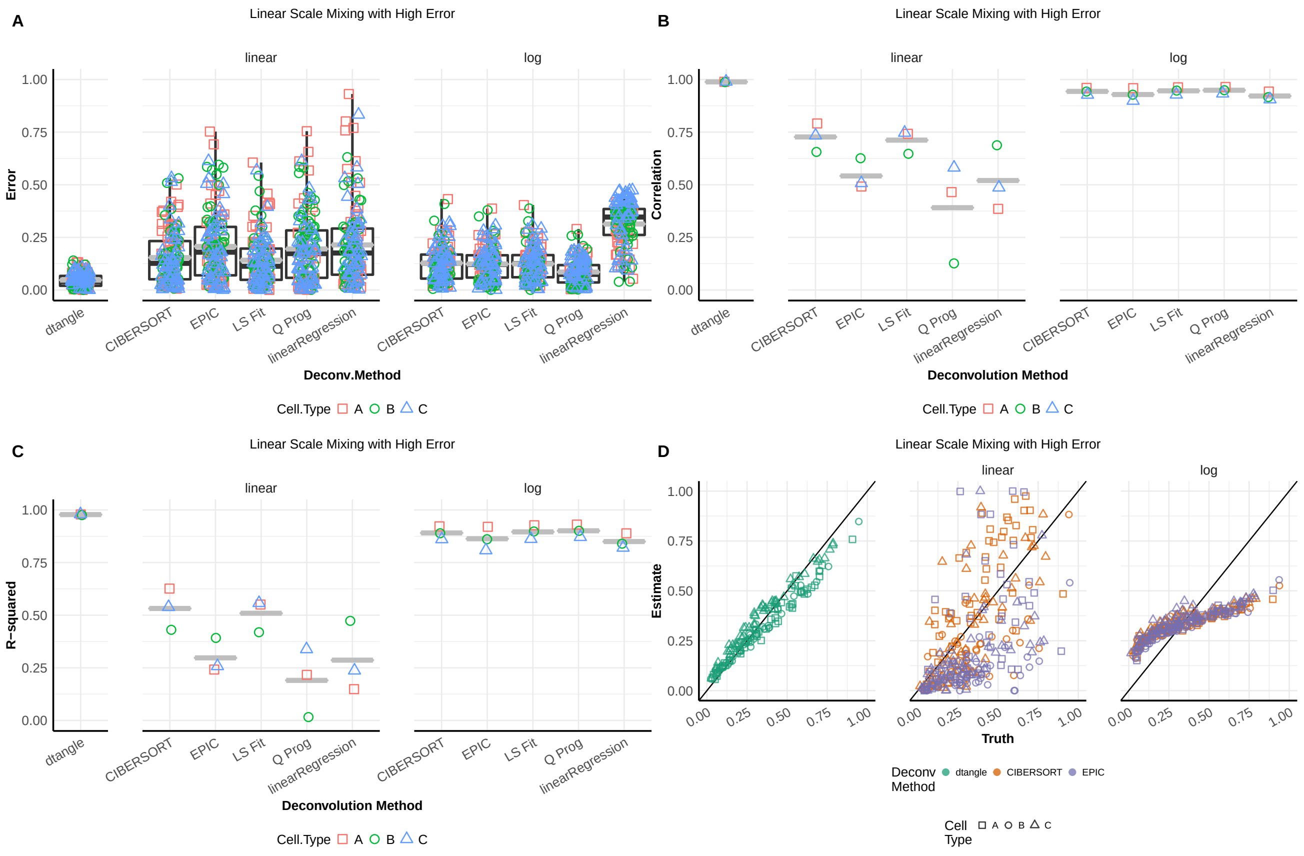


Fig. S29: Similar to Figure S28 but with a high error variance used in simulation.

Simulation: Artificial Cell Type with Low Gaussian Error and Outliers

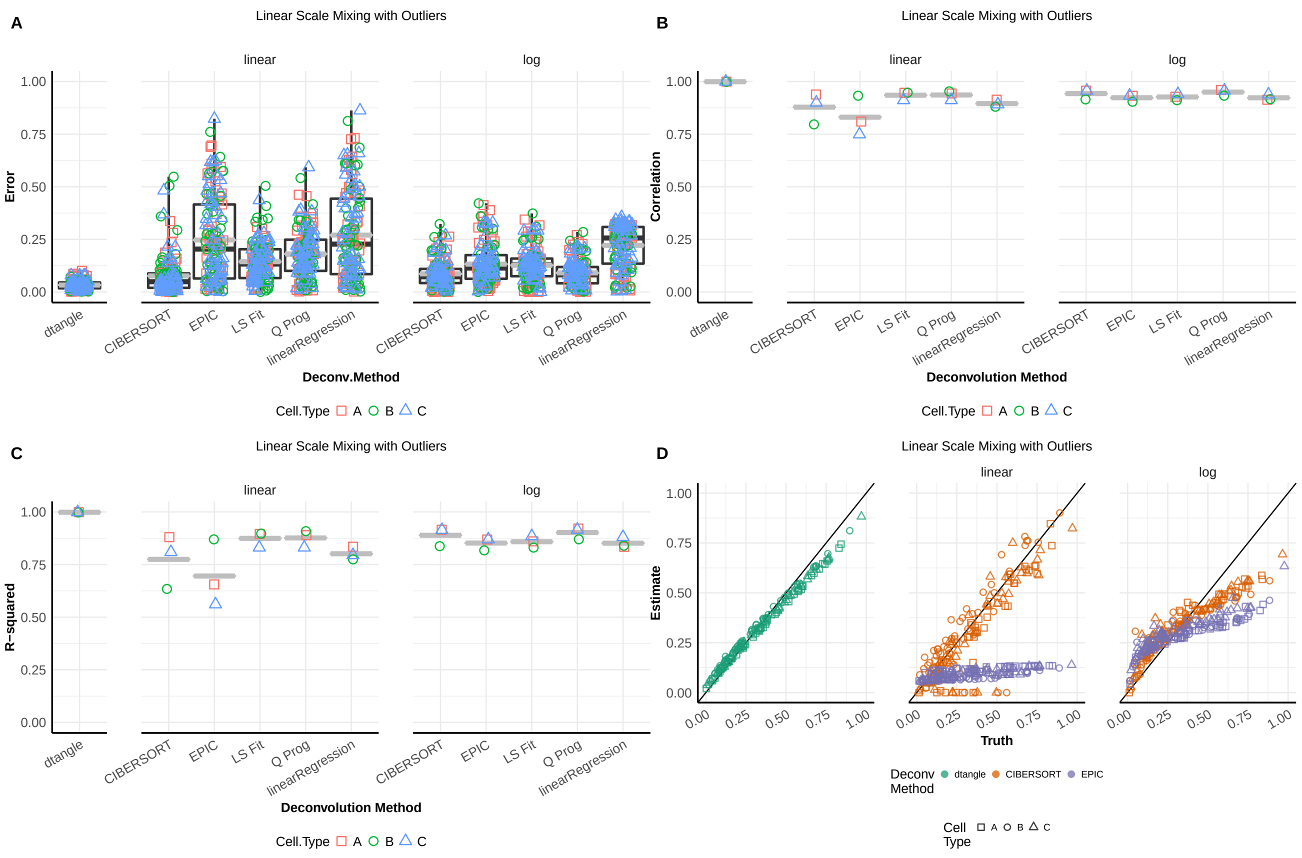


Fig. S30: Similar to Figure S28 but with outliers added to the simulated data.

Simulation: Artificial Cell Type with Gaussian Error by Number of Marker Genes

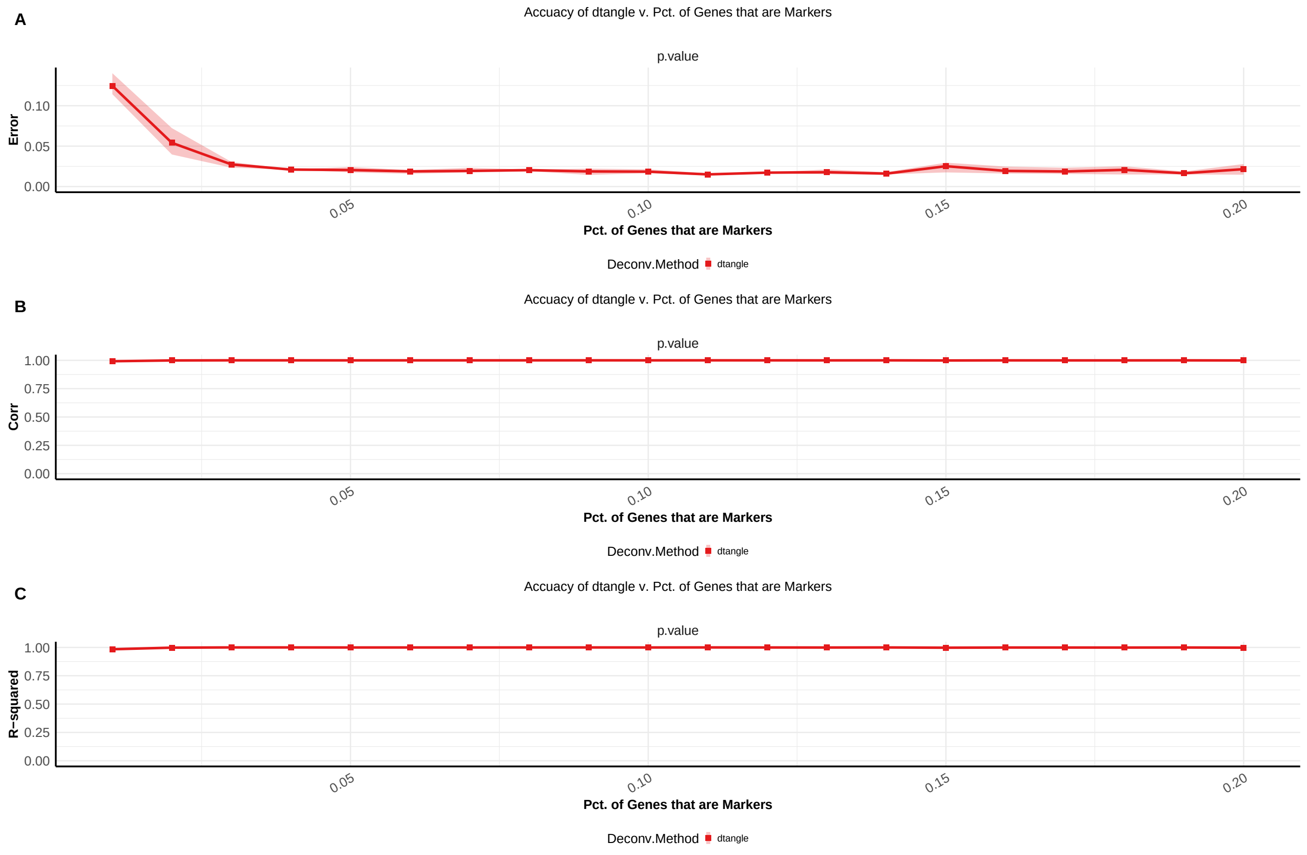


Fig. S31: Accuracy of dtangle by the number of marker genes present in gaussian simulated data with low error variance. y-axis is accuracy measured by (A) grand mean of the absolute value of the error of the true proportions from the estiamted proportions and (B) mean correlation within each cell type. The x-axis is the percentage of the data set that is comprised of marker genes as defined by dtangle.

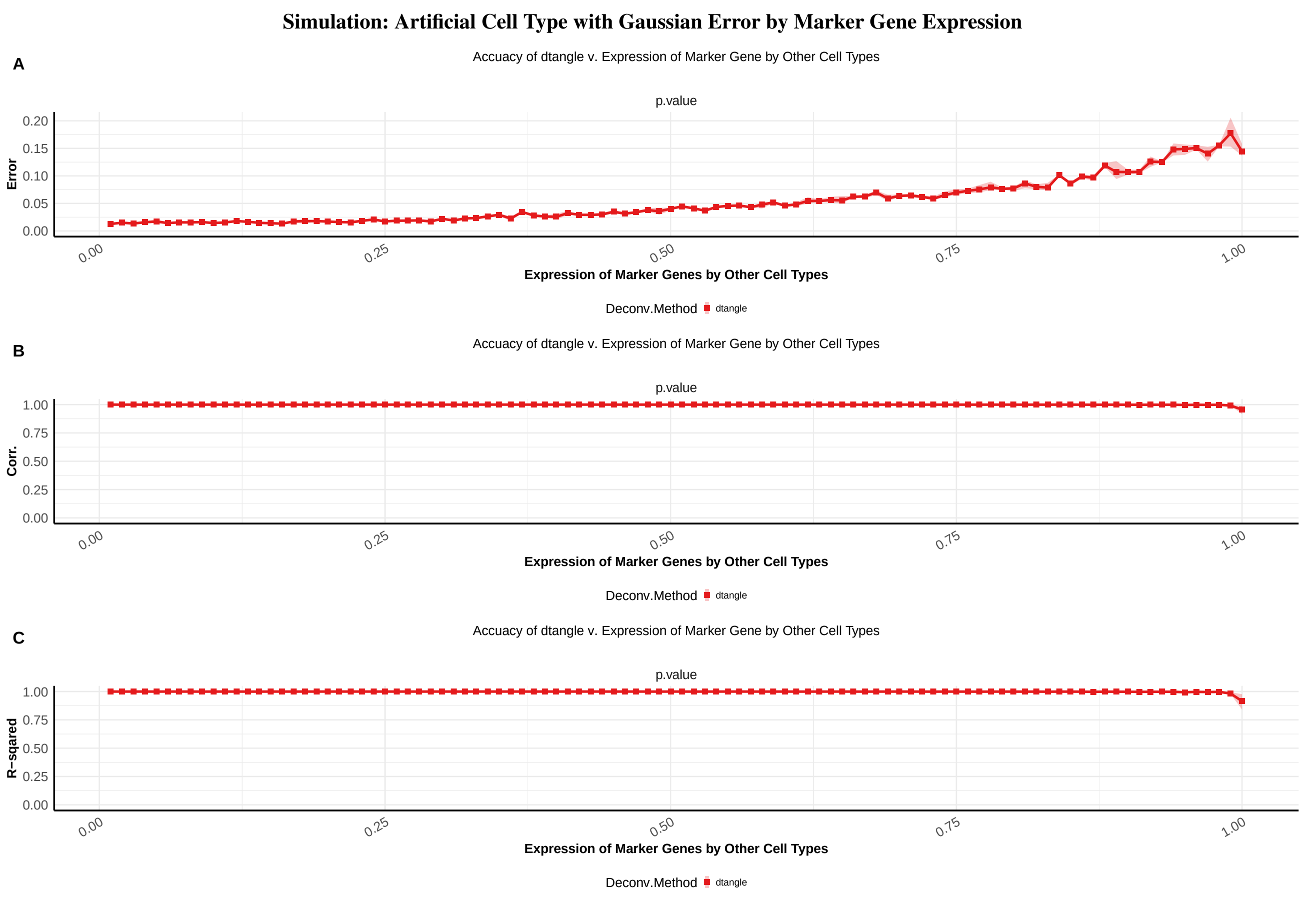


Fig. S32: Accuracy of dtangle by expression level of marker genes in gaussian simulated data with low error variance. y-axis is accuracy measured by (A) grand mean of the absolute value of the error of the true proportions from the estiamted proportions and (B) mean correlation within each cell type. The x-axis is the quantile of the over-all data at which marker genes are expressed in all other cell types.

Simulation: Artificial Cell Type with Poisson Error

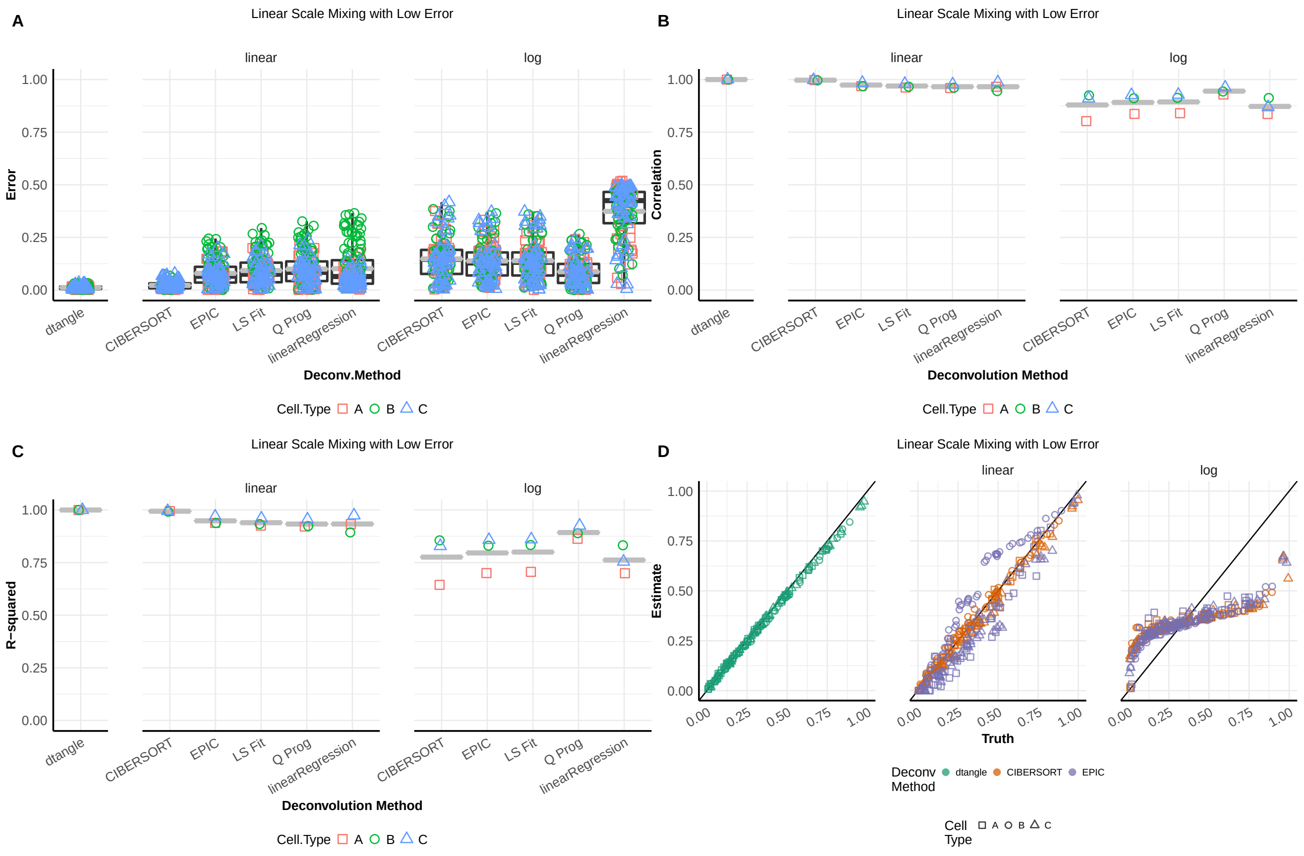


Fig. S33: Similar to Figure S28 but using a poisson error.

Simulation: Artificial Cell Type with Poisson Error and Outliers

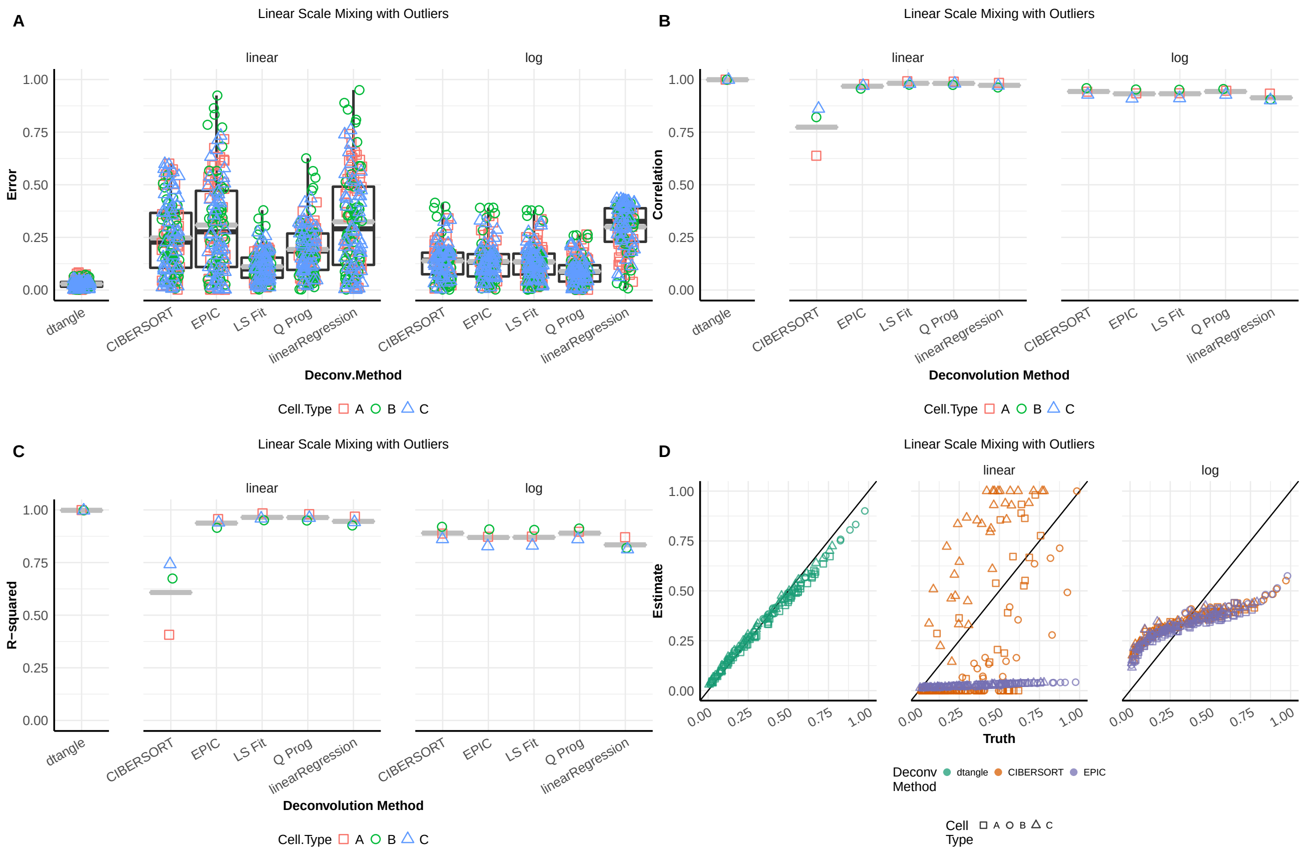


Fig. S34: Similar to Figure S33 but with outliers added to the simulated data.

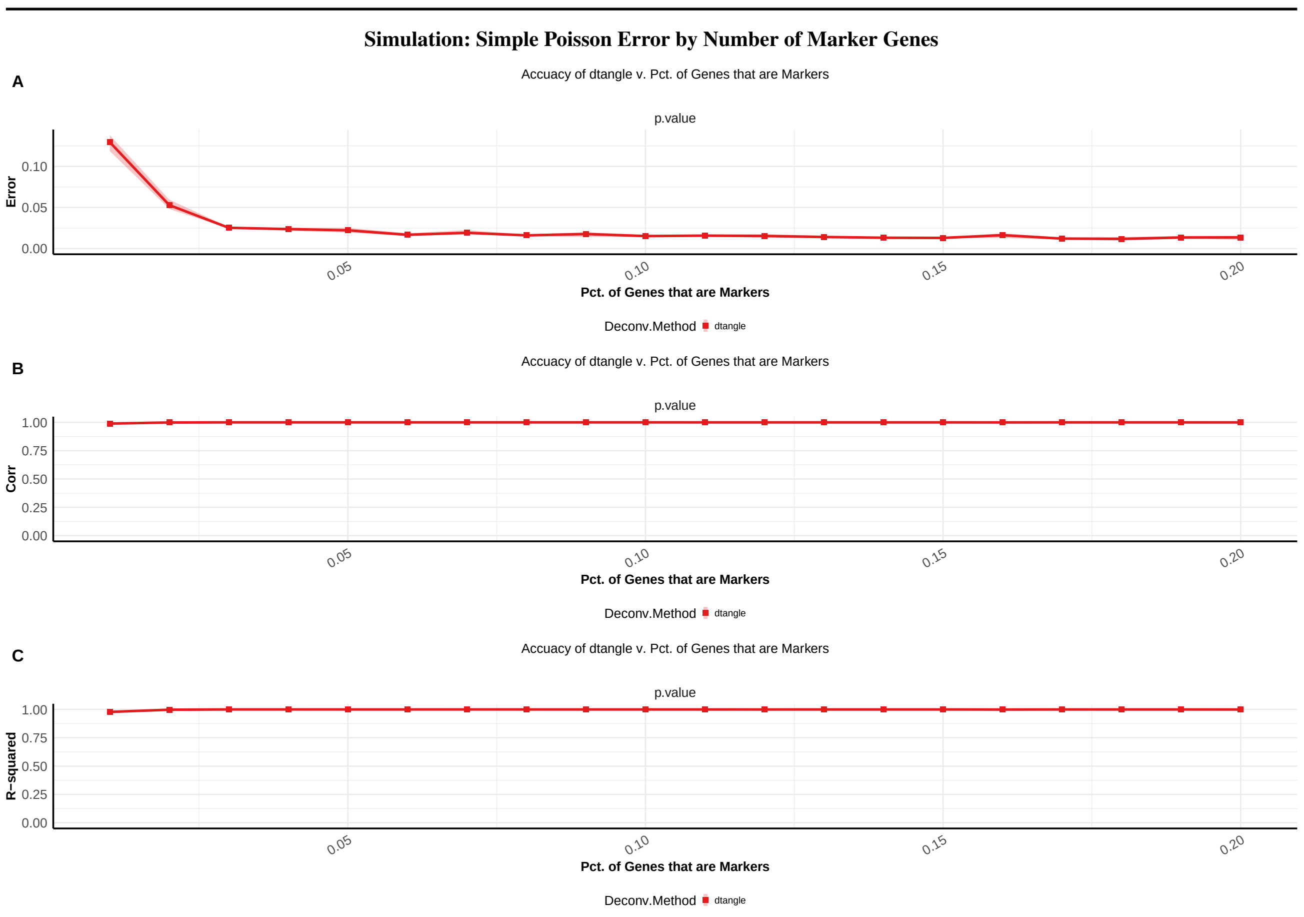


Fig. S35: Similar to Figure S31 but using a poisson error.

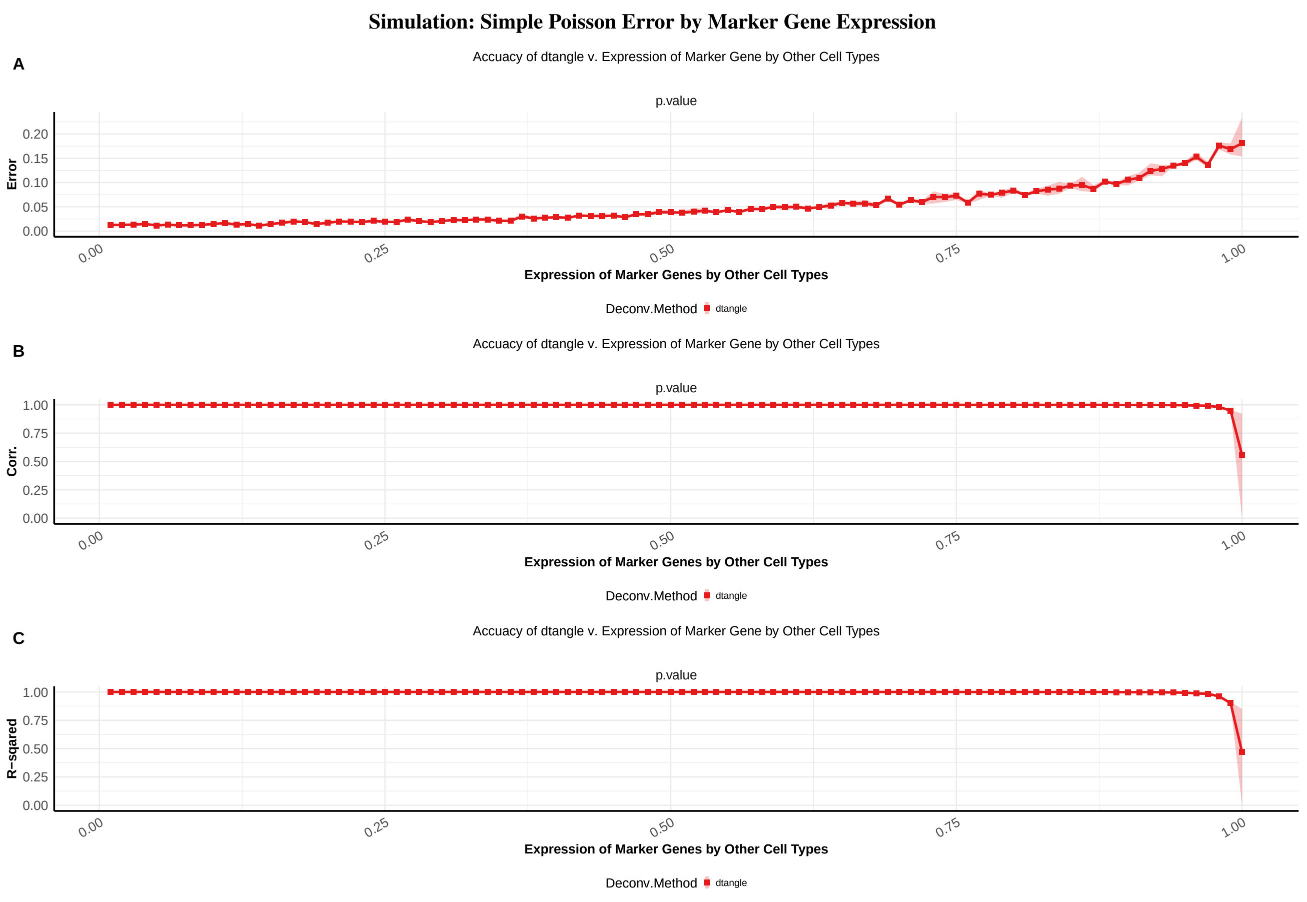


Fig. S36: Similar to Figure S32 but using a poisson error.

Simulation: Parsons with Low Gaussian Error

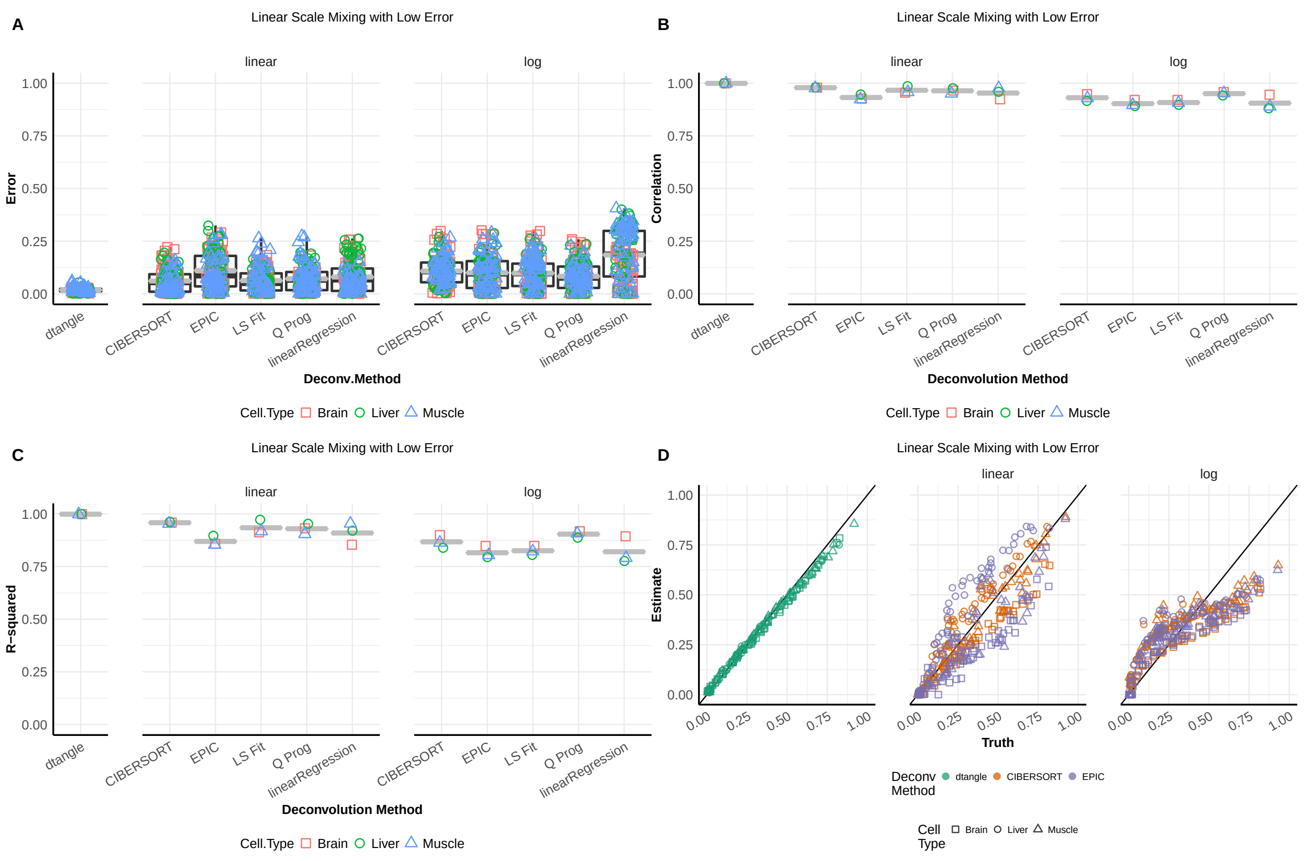


Fig. S37: Similar to Figure S28 but simulation was done by in-silico mixtures of reference cell type profiles from the Parsons data set.

Simulation: Parsons with High Gaussian Error

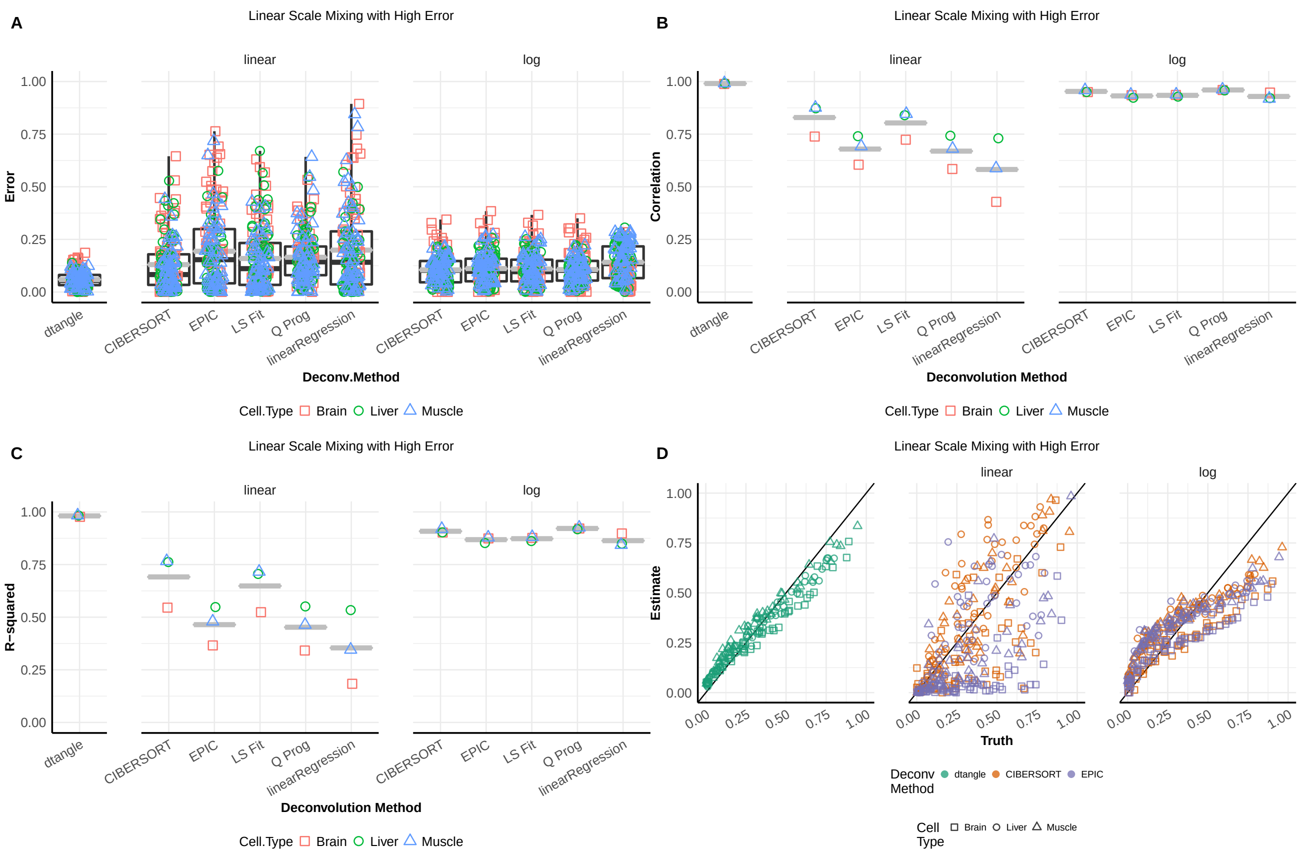


Fig. S38: Similar to Figure S37 but with a high error variance used in simulation.

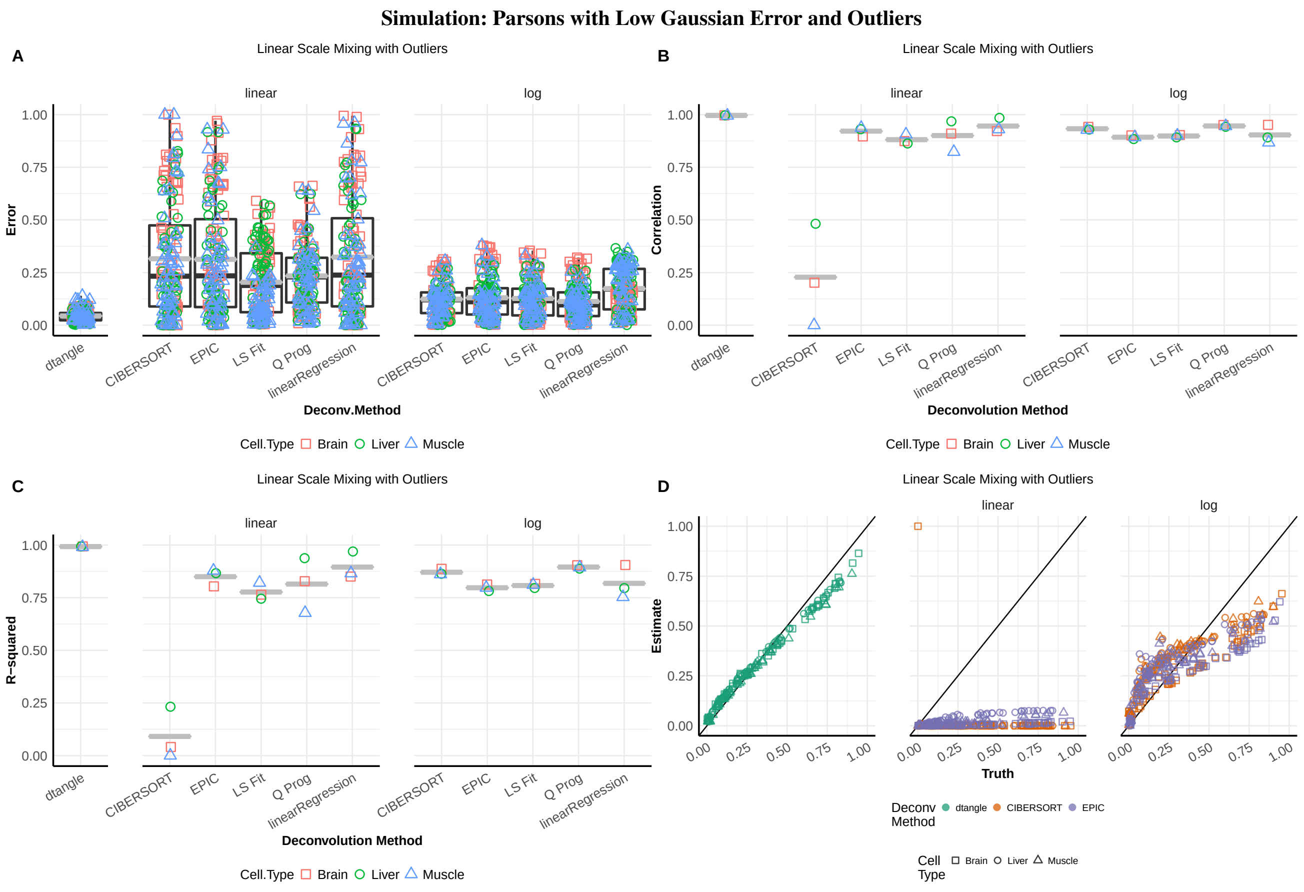


Fig. S39: Similar to Figure S37 but with outliers added to the simulated data.

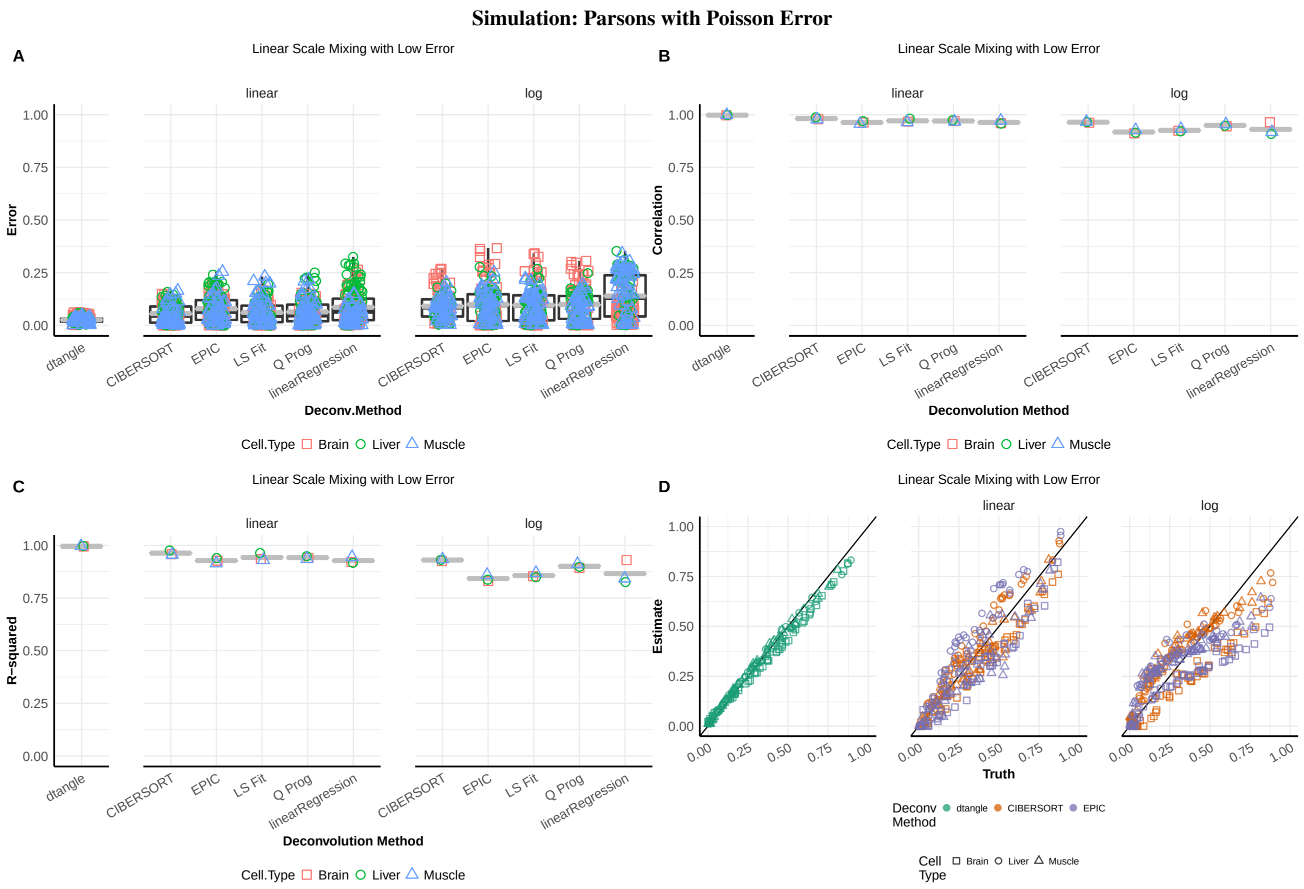


Fig. S40: Similar to Figure S37 but using poisson error.

Simulation: Parsons with Poisson Error and Outliers

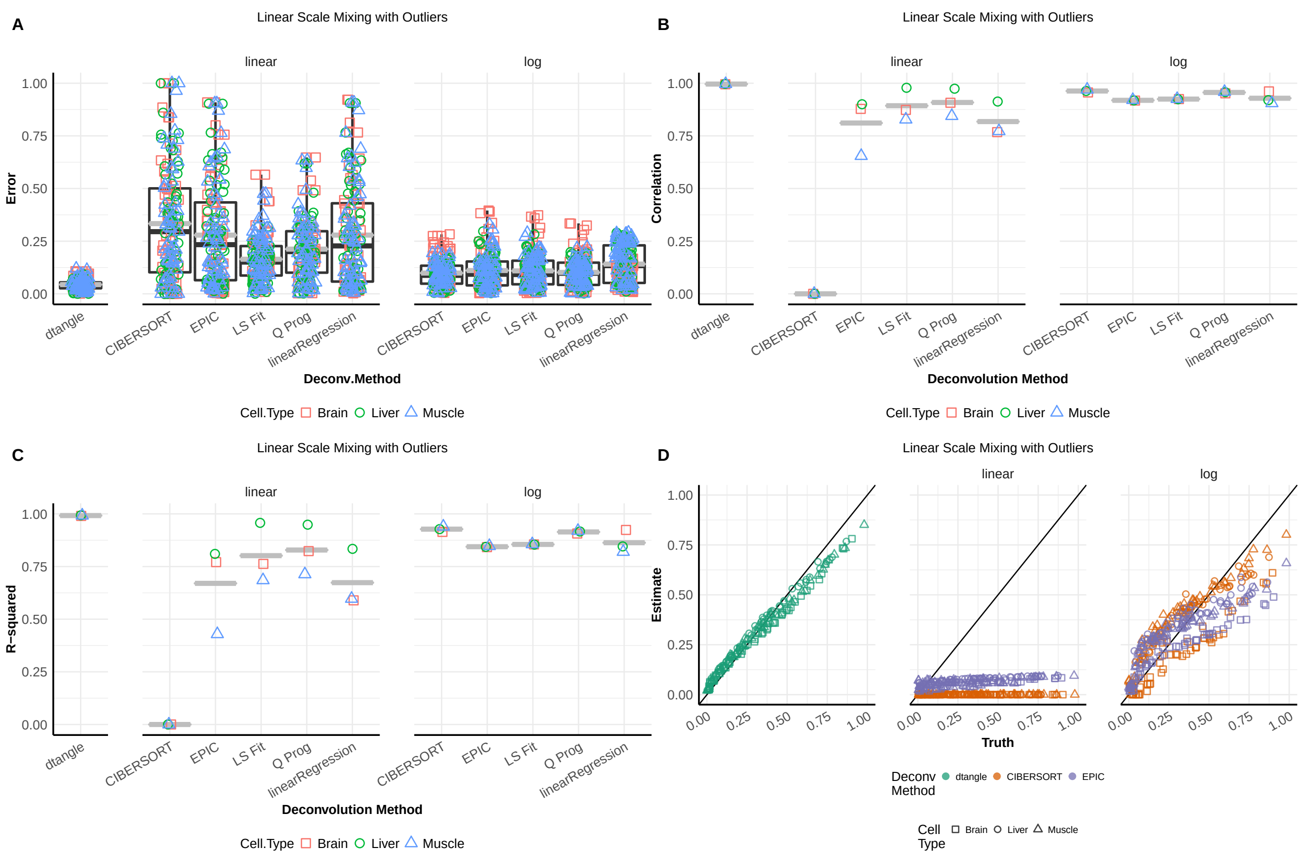


Fig. S41: Similar to Figure S40 but with outliers added to the simulated data.

Simulation: Linsley with Low Gaussian Error

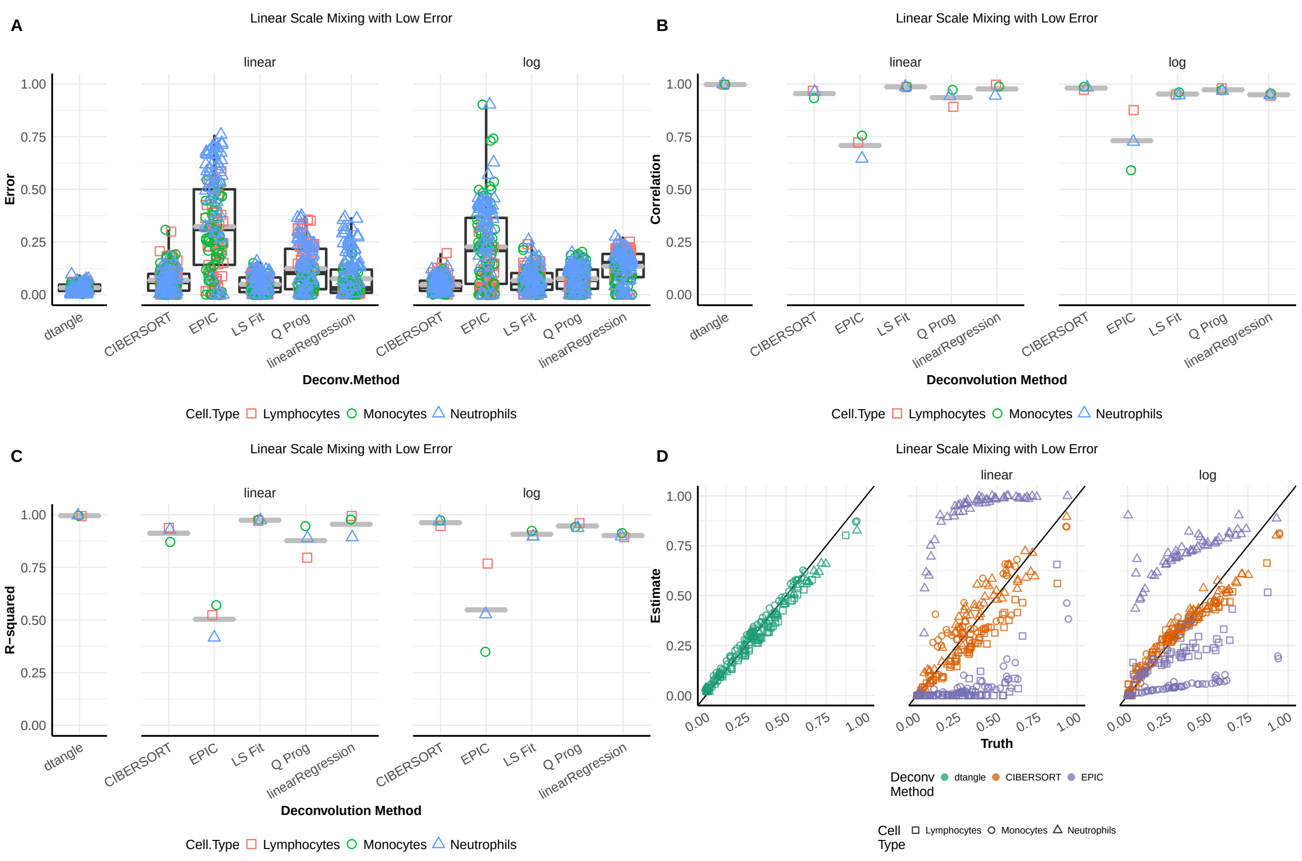


Fig. S42: Similar to Figure S28 but simulation was done by in-silico mixtures of reference cell type profiles from the Linsley data set.

Simulation: Linsley with High Gaussian Error

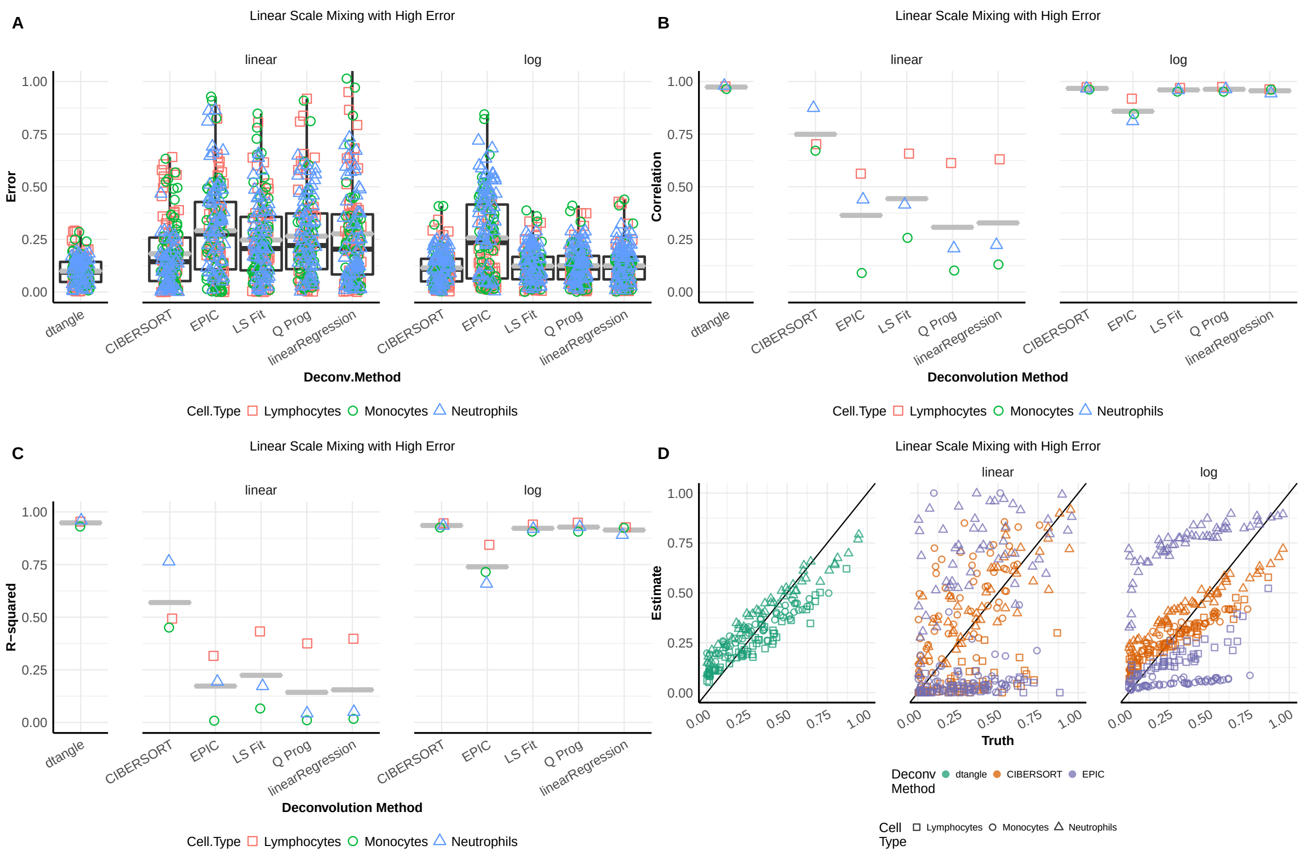


Fig. S43: Similar to Figure S42 but with a high error variance used in simulation.

Simulation: Linsley with Low Gaussian Error and Outliers

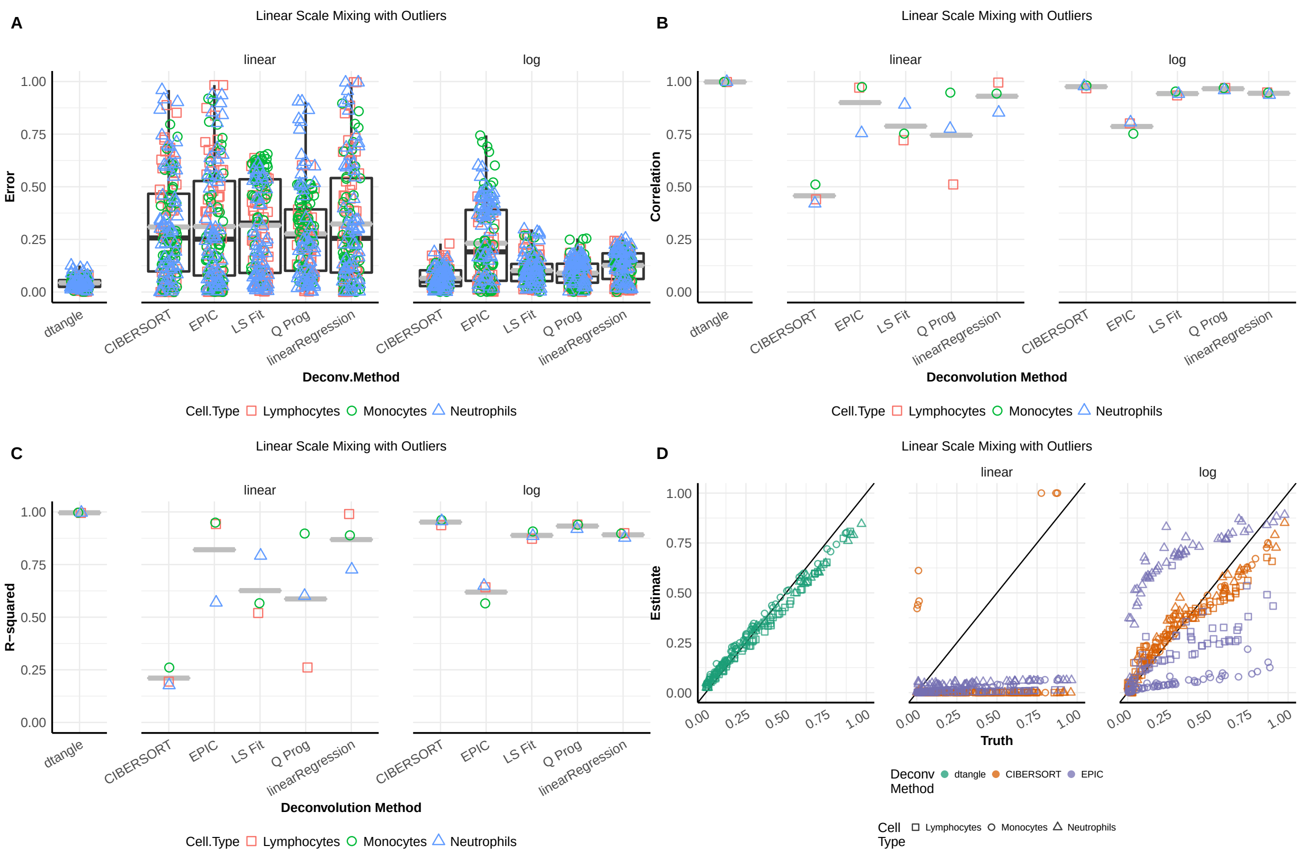


Fig. S44: Similar to Figure S42 but with outliers added to the simulated data.

Simulation: Linsley with Poisson Error

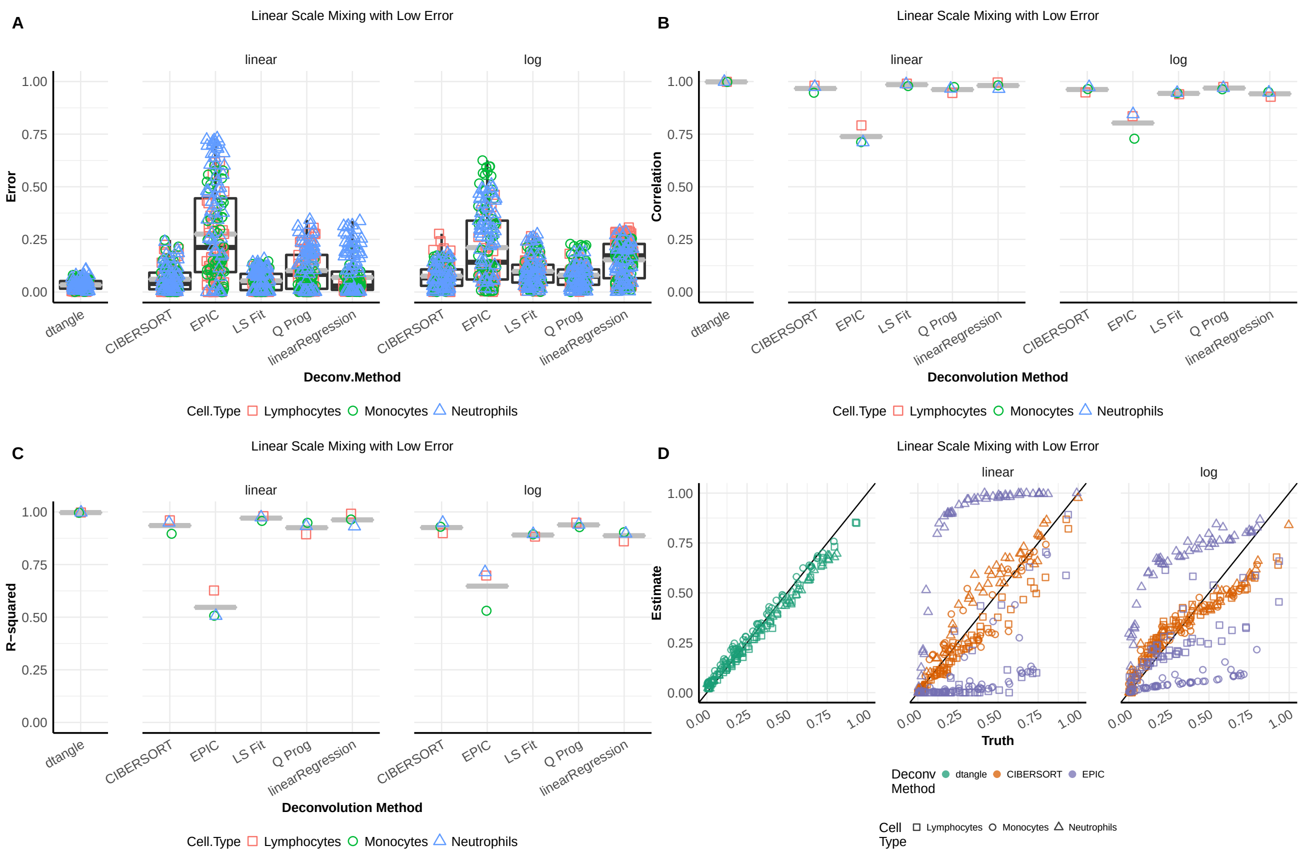


Fig. S45: Similar to Figure S42 but using poisson error.

Simulation: Linsley with Poisson Error and Outliers

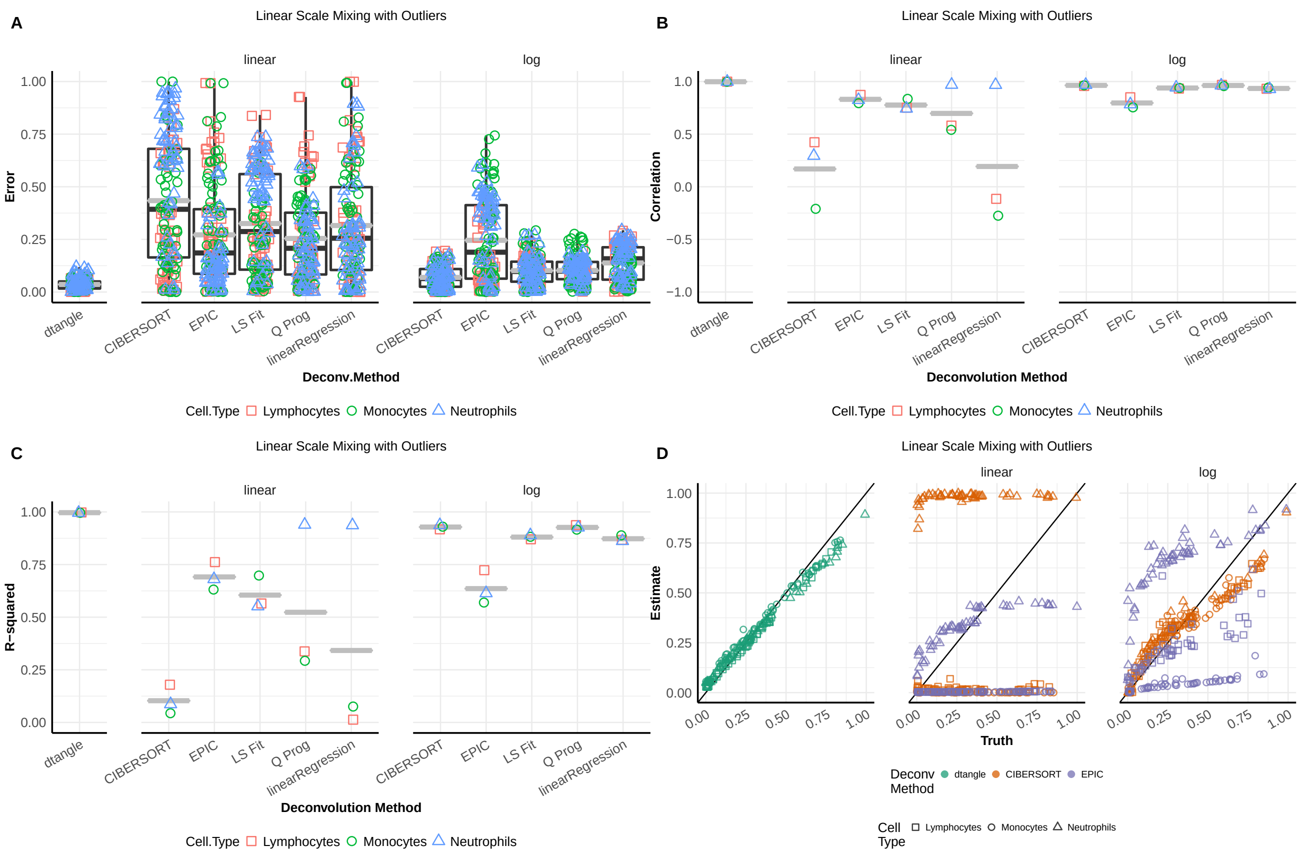


Fig. S46: Similar to Figure S45 but with outliers added to the simulated data.