Size factor experiments

Sean Maden

2022-12-08

This vignette demonstrates the principle of type-specific factor transformations for deconvolution experiments. This is is useful, for instance, when accounting for differing cell sizes in deconvolution of bulk tissue RNA-seq data.

1 Table of contents

Overview

Simulations

Conclusions

2 Overview

2.1 Terminology

We represent the deconvolution problem as:

$$Y = Z * S * P$$

Where we have:

- $Y[G, J]$: The convoluted signal matrix with dimensions $G*J$ where G is the set of markers (e.g. marker genes), and J is the set of bulk samples (e.g. from bulk tissue RNA-seq).
- $Z[G, K]$: The signature matrix having dimensions $G * K$ where G is the set of markers (e.g. marker genes), and K is the set of types (e.g. cell types).
- S : The type-specific factors, which can be represented as a vector of length K.
- P : The type-specific proportions, represented as a vector of length K.

2.2 Methods for calculating Z

We wish to predict P using some Y and Z. When the type-specific factors/sizes in S are known or can be reliably estimated, we have the following options for representing Z:

1. $Z_1 = Z$: This is the unadjusted representation of the signature signals by type.
2. $Z_2 = Z * S$: This is the signature signals adjusted, or multiplied, by the type-specific factors or sizes.

2.3 Simulation experiments

Below, we will show the impact of transformation Z on some S. We evaluate the transformation's impact on the P predictions from deconvolution. For each simulation, we generate the objects Z, S, and P, then we generate the pseudobulked Y data as the product of $Z * S * P$. Finally, we predict P using the two methods of calculating Z (see above). For these experiments, we set the total types to be two or $K = 2$, and we set the total markers to be two or $G = 2$.

We will consider the S factors to represent the true sizes of the simulated cell types, and our terminology below will reflect this. We perform three experiments where we vary the type-specific sizes of S in each. In other words, we vary the relative sizes of each type (s_1 for type 1 (or k_1) and s_2 for type 2 (or k_2) in the following ways:

(A) Cell sizes equal:

$$s_1 == s_2$$

(B) Cell sizes slightly different:

$$s_1 \approx s_2$$

(C) Cell sizes very different: size of type k_2

$$s_1 \gg s_2$$

Each experiment consists of a series of 1,000 simulations wherein the proportions of two simulated types are themselves varied in complementary fashion (see below).

2.4 Results evaluations

Below, we show the summary plots evaluating the results of simulations. There are two comparator groups, one for each distinct value of Z, Z_1 and Z_2, used in the predictions. These are labeled below according to the status of the `zs_transformation` variable, where `FALSE` indicates no transformation/Z_1, and `TRUE` indicates the transformation `Z*S`/Z_2. The former group is represented in red, and the latter is represented in teal.

We evaluate results using the root mean squared error (RMSE) across types. That is, an individual simulation yields an RMSE for each of two simulated types, and we calculate the RMSE across these types using:

$$RMSE = \sqrt{\sum_{k_i=1}^{K} (p_i - \hat{p}_i)^2}$$

Where we have:

- K : The total number of types, and here $K = 2$
- p_i : The true proportion for type k_i
- $\hat{p}_i$: The estimated proportion for type k_i, here using the non-negative least squares (NNLS) function `nnls::nnls()`.

Separate RMSE calculations are made for the cases of Z_1 and Z_2, respectively.

3 Simulations

This section shows how to use `lute` to rapidly set up and carry out a series of deconvolution simulation experiments.

3.1 Experiment setup

We will carry out 1,000 simulations for each series, which we specify by assigning the variable `num.sim`.

```
num.sim = 1e3
```

We will start with the simplest case of having two cell types (e.g. $K = 2$), and having two marker genes (e.g. $G = 2$), one "positive" marker for each type. Get the simulated marker data, corresponding to the Z matrix, using:

```
num.sim <- 1e3
lgv <- random_lgv(gindexv = c(1, 2), num.iter = num.sim)
```

We also need to specify the **true** type-specific proportions for simulations. We can do this automatically using the function `make_lpv()` to simulate the complementarity that we see for real cell type proportions.

```
lpv <- make_lpv()
```

That's all there is to the basic simulation setup.

3.2 Experiment series A

Specify equal cell sizes as `size1` and `size2` ($s_1 = s_2$), then store these as a list `lsv` having the same length as the object `lgv` above. These list lengths correspond to the total simulations to be run in the series.

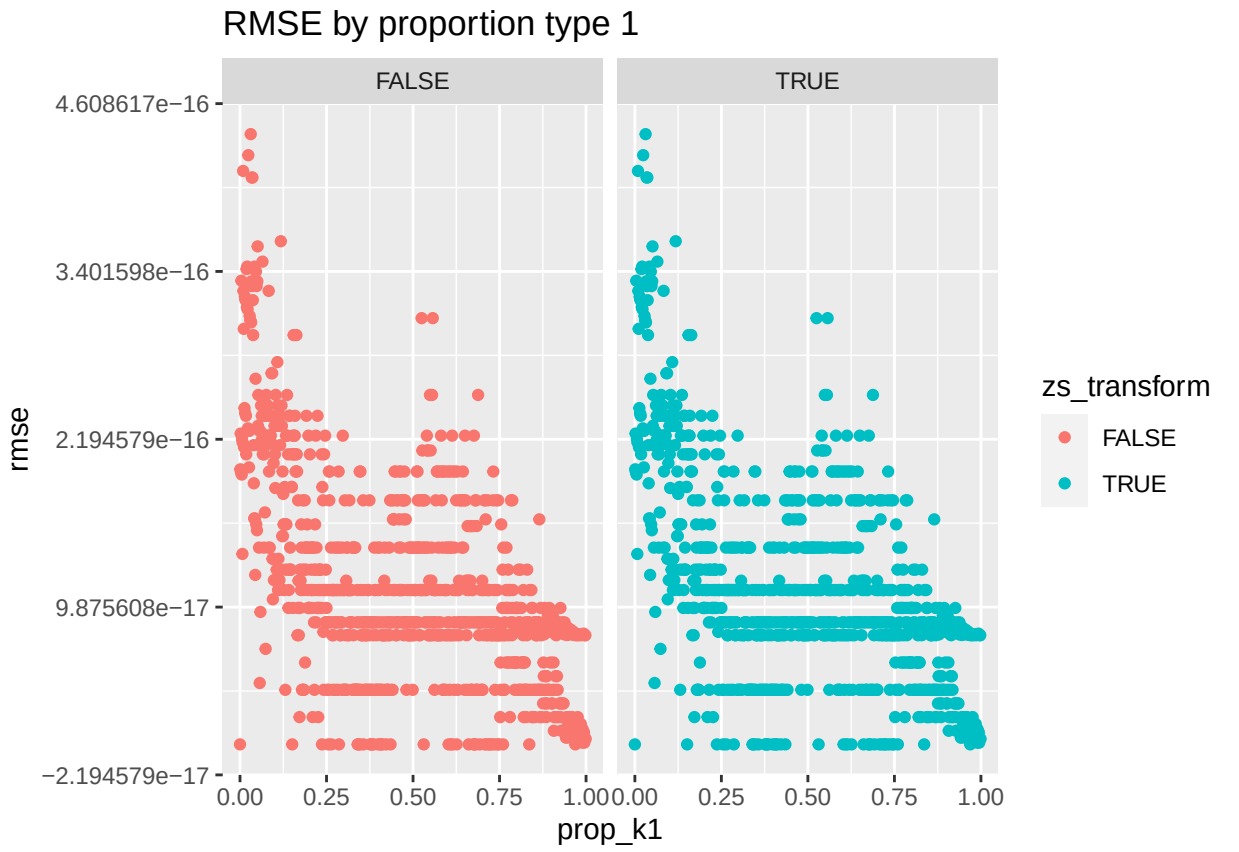
```
size1 <- size2 <- 1
lsv <- lapply(seq(num.sim), function(ii){c(size1, size2)})
```

We now run deconvolution simulations using the `decon_analysis()` function and providing the marker gene signals `lgv`, cell sizes `lsv`, and proportions `lpv` (see `?decon_analysis` for details).

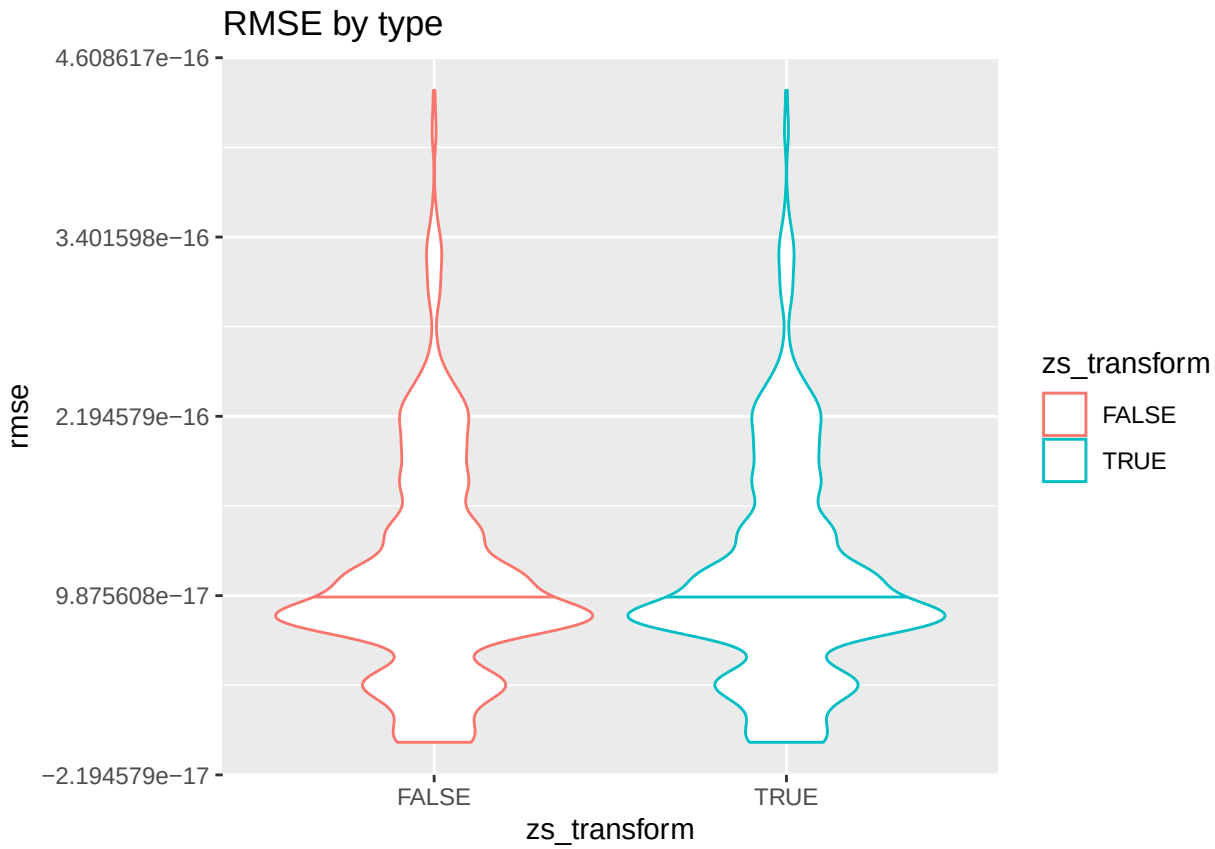
```
lres <- decon_analysis(lgv = lgv, lpv = lpv, lsv = lsv)
```

We can view summary plots of the simulations with:

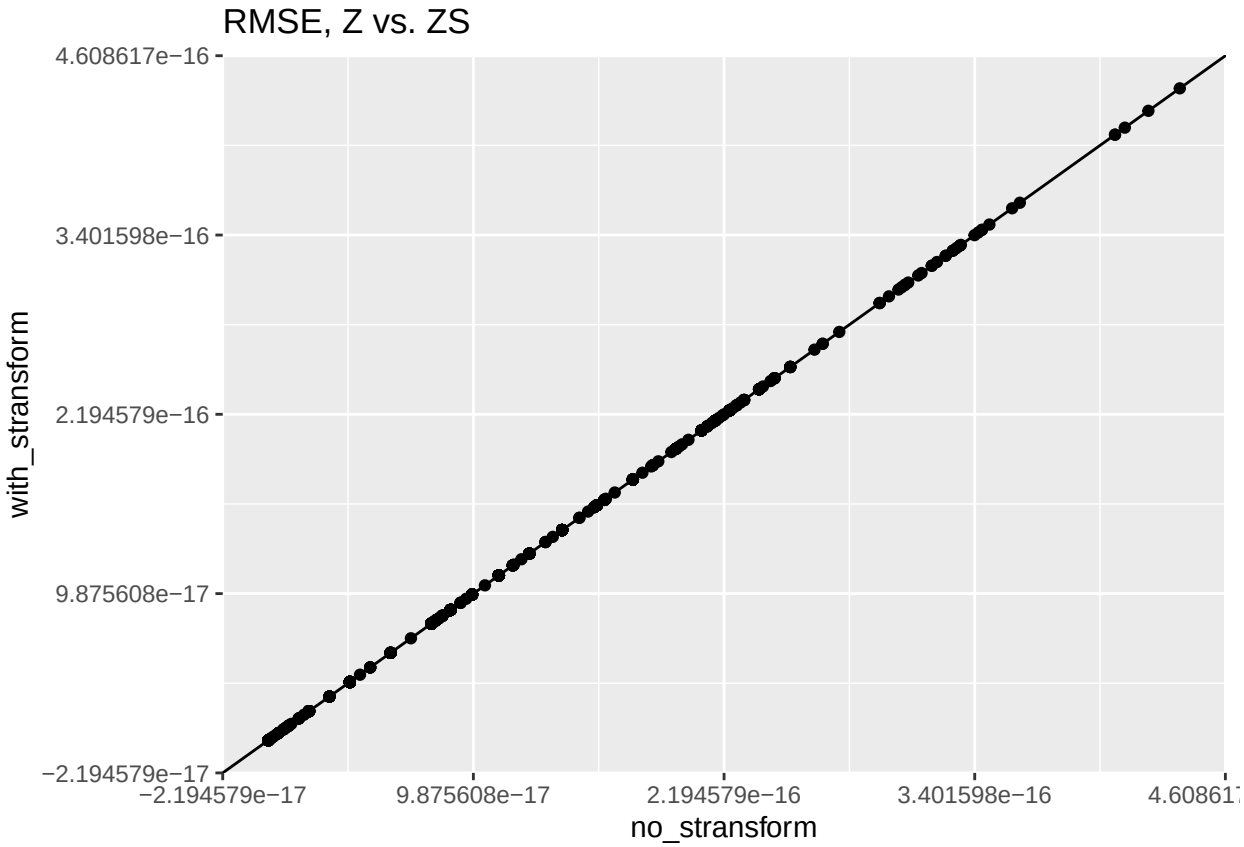
```
lres$lgg$ggpt1
```




```
lres$lgg$ggvp
```



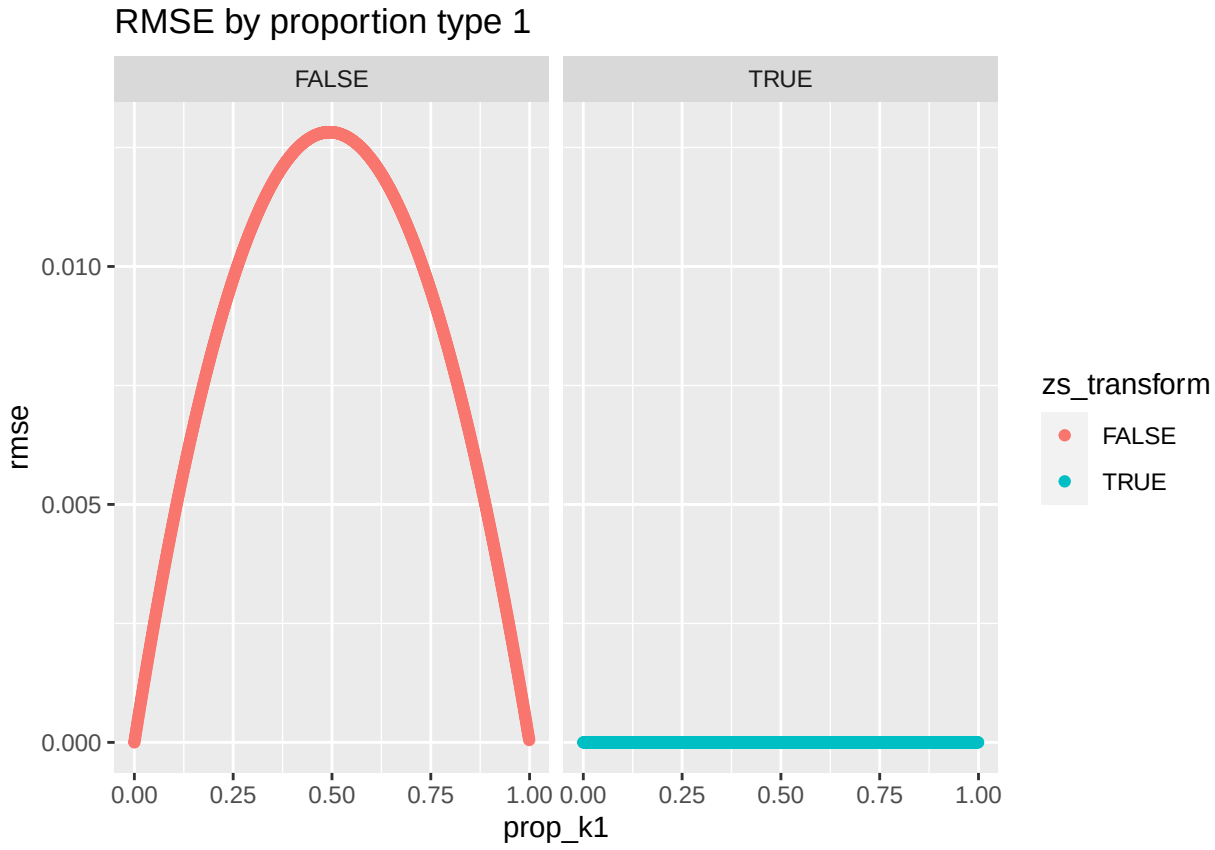

```
lres$lgg$ggpt2
```



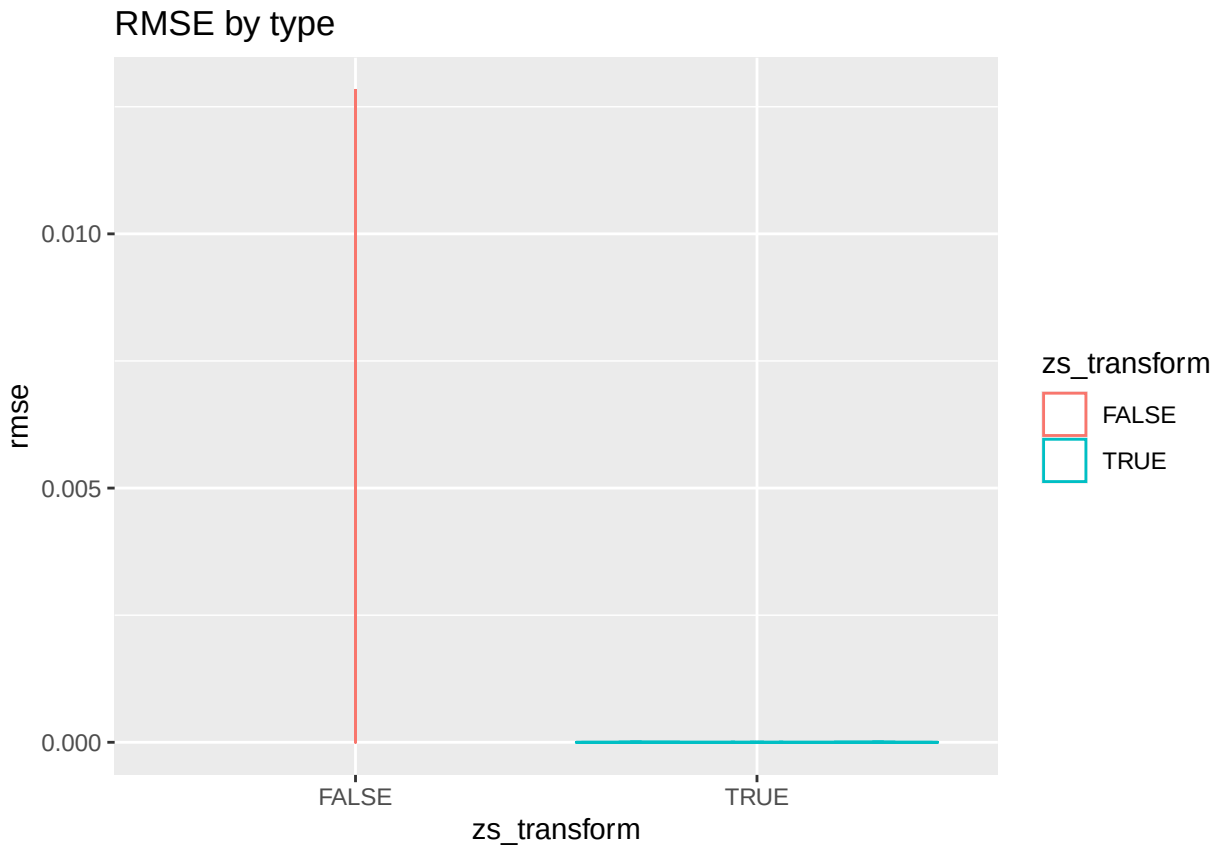
3.3 Experiment series B

For this series, we set the type-specific sizes to be slightly different ($s_1 \approx s_2$). We will set type 1's size to be 100 and type 2's size to be 95, then run a new simulation series. Finally, we will view the summary plots as before.

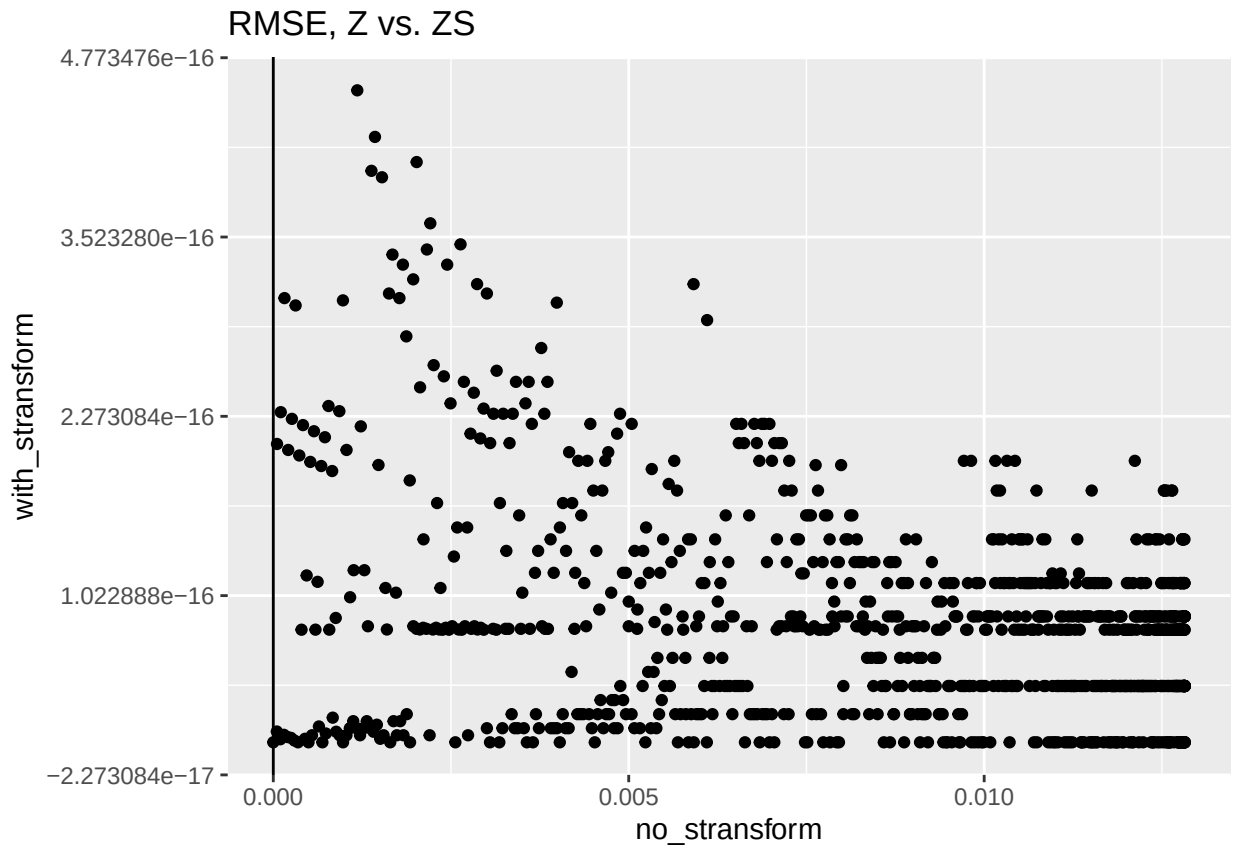
```
size1 <- 100
size2 <- 95
lsv <- lapply(seq(num.sim), function(ii){c(size1, size2)})
lres <- decon_analysis(lgv = lgv, lpv = lpv, lsv = lsv)
lres$lgg$ggpt1
```




```
lres$lgg$ggvp
```



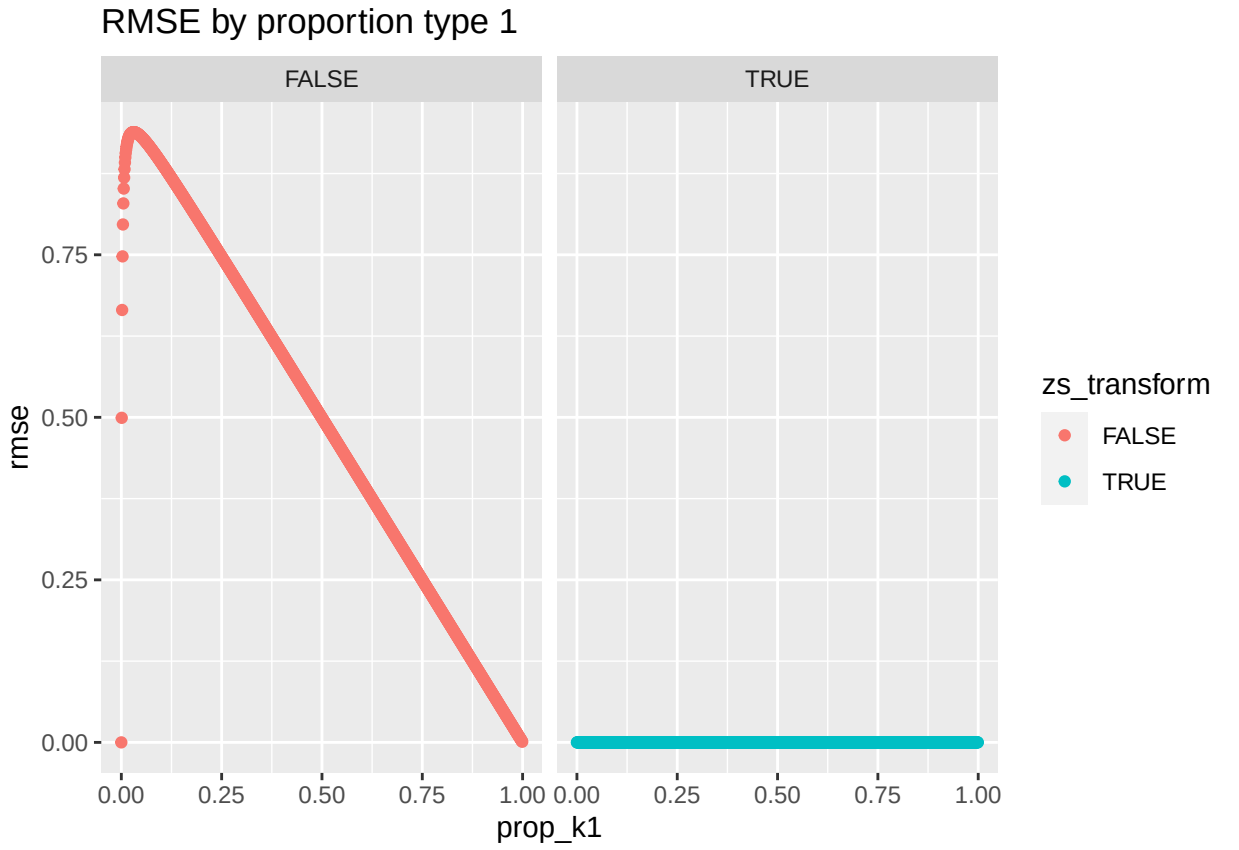

```
lres$lgg$ggpt2
```



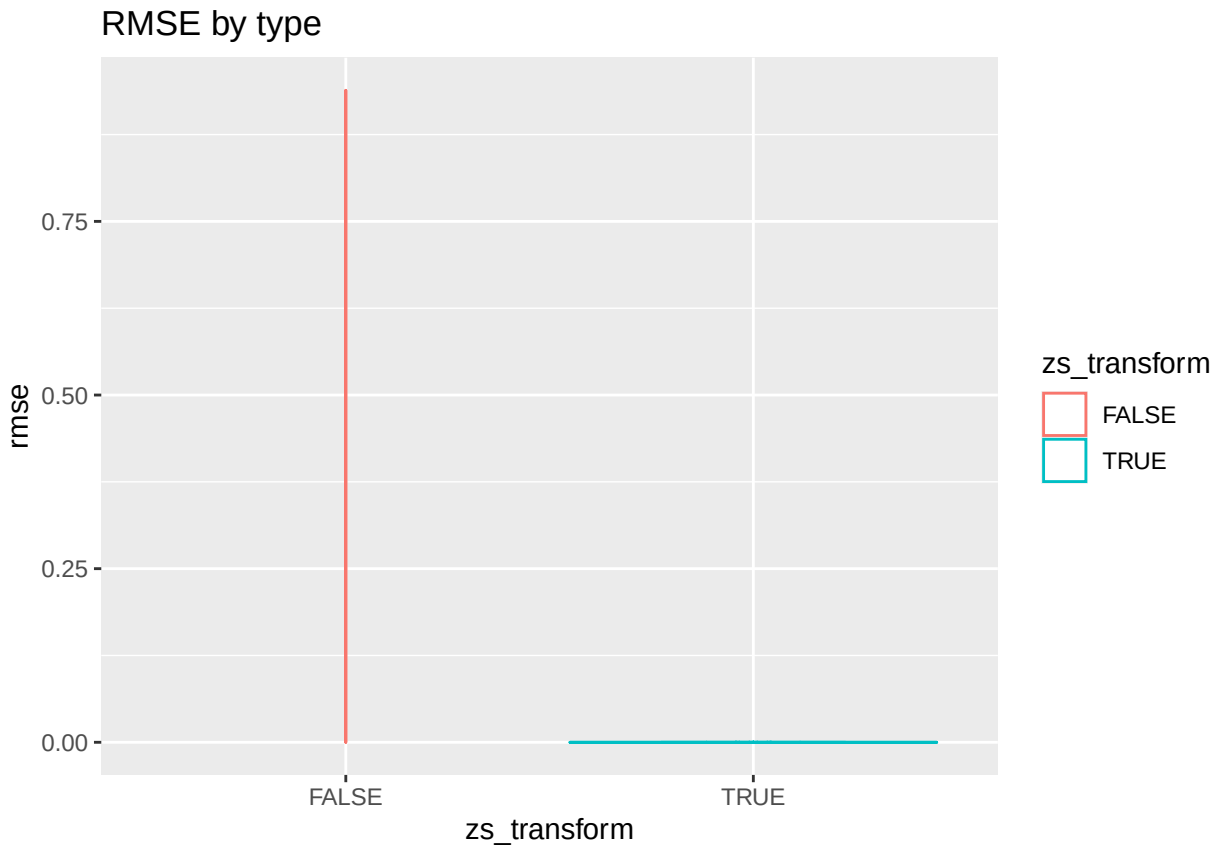
3.4 Experiment series C

For this series, we make type 1 much larger than type 2 ($s_1 \gg s_2$). We will set type 1 to have size 1000 and type 2 to have size 1, then run and summarize simulations as above.

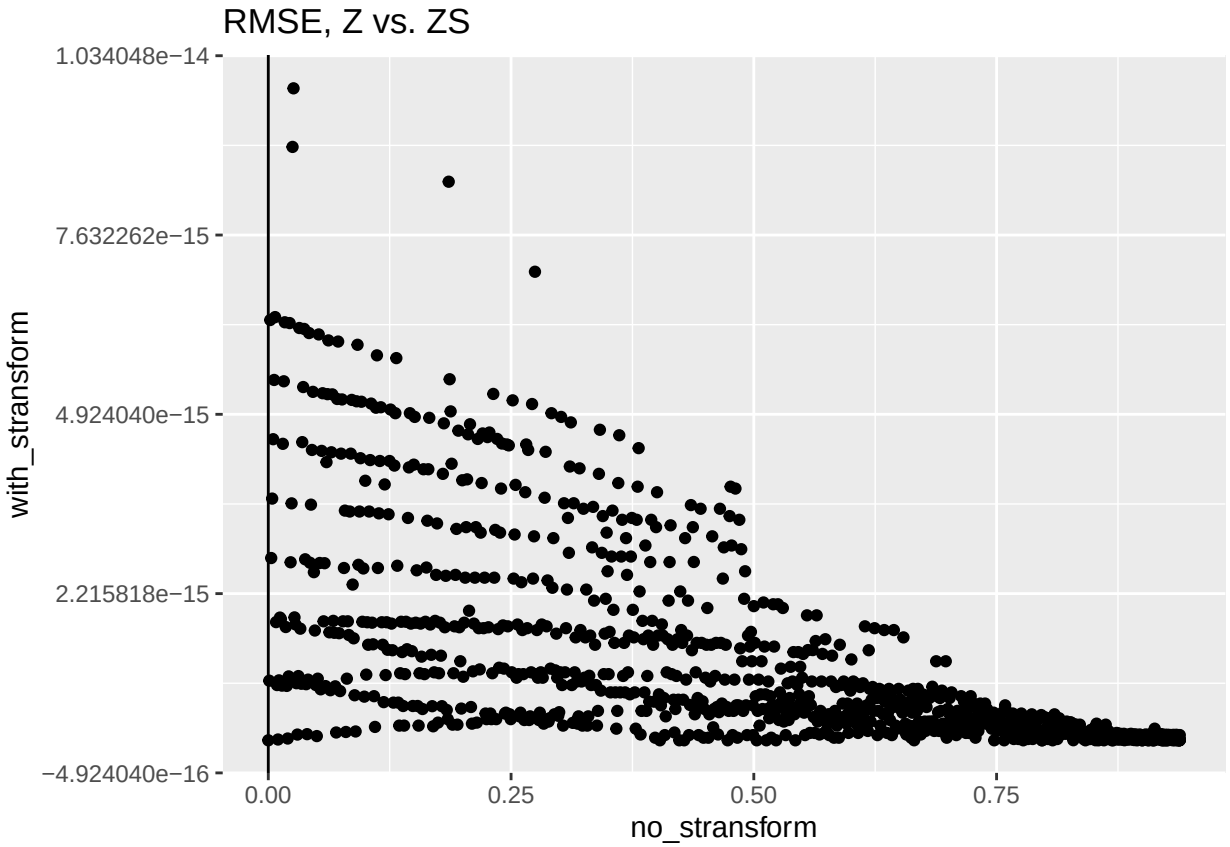
```
size1 <- 1e3
size2 <- 1
lsv <- lapply(seq(num.sim), function(ii){c(size1, size2)})
lres <- decon_analysis(lgv = lgv, lpv = lpv, lsv = lsv)
lres$lgg$ggpt1
```




```
lres$lgg$ggvp
```




```
lres$lgg$ggpt2
```



4 Conclusions

In this vignette, we showed a simple experiment evaluating the impact of a transformation on type-specific factor transformation on deconvolution simulation outcomes. We measured outcomes using RMSE across types, and comparing data in visualizations for the case of predictions using either the (1) unadjusted signals matrix Z, or (2) the adjusted signals matrix $Z * S$ (see above for details).

In experiment series A, we see a validation of the design (a.k.a. "sanity check") where results are virtually identical for the two conditions when cell sizes are equal. In series B, when cell sizes differ just slightly, we see noticeably lower RMSE for most simulations in the case of (2), denoted as`zs_transform == TRUE` or the teal color in the plots. In series C, when cell sizes are drastically different, we see even lower RMSE for (2)/`zs_transform == TRUE`. These findings show the importance of correcting on a type-specific factor when present in the convoluted signals matrix Y, such as when quantifying expression in bulk tissue RNA-seq data.