

donor_bias

Sean Maden

2022-12-21

This vignette walks through a series of deconvolution analyses evaluating the impact of between-donor variations on results.

Experiment series 1: $\text{ndonors} = 2$

Let's run a series of experiments keeping $\text{ndonor} = 2$. We start with the simplest example of $k = 2$ types and $G = 2$ total markers, one for each type. We will vary the offset from small (2) to medium (5) to large (10). This offset magnitude should be proportional to observed between-donor variances.

```
ndonor = 2
gindexv = c(1, 2)
offsetv <- c(2, 5, 10)
num.sim <- 50
lexpt <- lapply(offsetv, function(offi){
  title.append <- paste0("Offset = ", offi, "\n")
  donor_marker_experiment(gindexv = c(1,2), num.sim = num.sim, ndonor = ndonor,
    plot.title.append = title.append,
    sd.offset.pos = offi,
    sd.offset.neg = offi)
})
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:Biobase':
```

```
##
```

```
##      combine
```

```
## The following objects are masked from 'package:GenomicRanges':
```

```
##
```

```
##      intersect, setdiff, union
```

```
## The following object is masked from 'package:GenomeInfoDb':
```

```
##
```

```
##      intersect
```

```
## The following objects are masked from 'package:IRanges':
##
## collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
## first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
## combine, intersect, setdiff, union

## The following object is masked from 'package:matrixStats':
##
## count

## The following objects are masked from 'package:stats':
##
## filter, lag

## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

```
names(lexpt) <- paste0("offset:", offsetv)
```

Randomized marker data

```
knitr::kable(lexpt$`offset:2`$marker.table, align = "c")
```

donor1	donor2	donor.combn.all.mean	donor.combn.all.median	type	marker	marker.type
28	24	26.0	26.0	type1	marker1	type1
4	4	4.0	4.0	type1	marker2	type2
3	3	3.0	3.0	type2	marker1	type1
23	20	21.5	21.5	type2	marker2	type2

```
knitr::kable(lexpt$`offset:5`$marker.table, align = "c")
```

donor1	donor2	donor.combn.all.mean	donor.combn.all.median	type	marker	marker.type
32	23	27.5	27.5	type1	marker1	type1
8	7	7.5	7.5	type1	marker2	type2
6	6	6.0	6.0	type2	marker1	type1
27	19	23.0	23.0	type2	marker2	type2

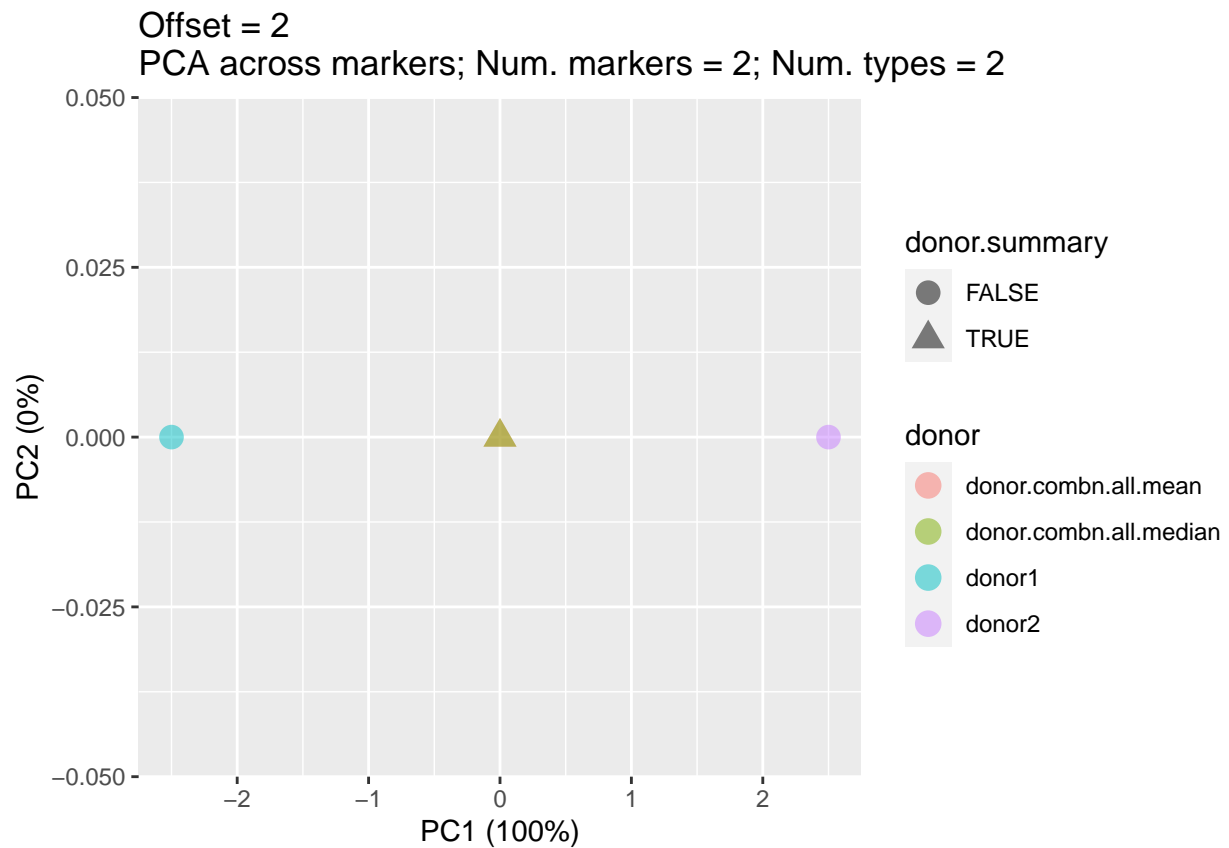
```
knitr::kable(lexpt$`offset:10`$marker.table, align = "c")
```

donor1	donor2	donor.combn.all.mean	donor.combn.all.median	type	marker	marker.type
39	21	30.0	30.0	type1	marker1	type1
14	13	13.5	13.5	type1	marker2	type2
20	19	19.5	19.5	type2	marker1	type1
40	22	31.0	31.0	type2	marker2	type2

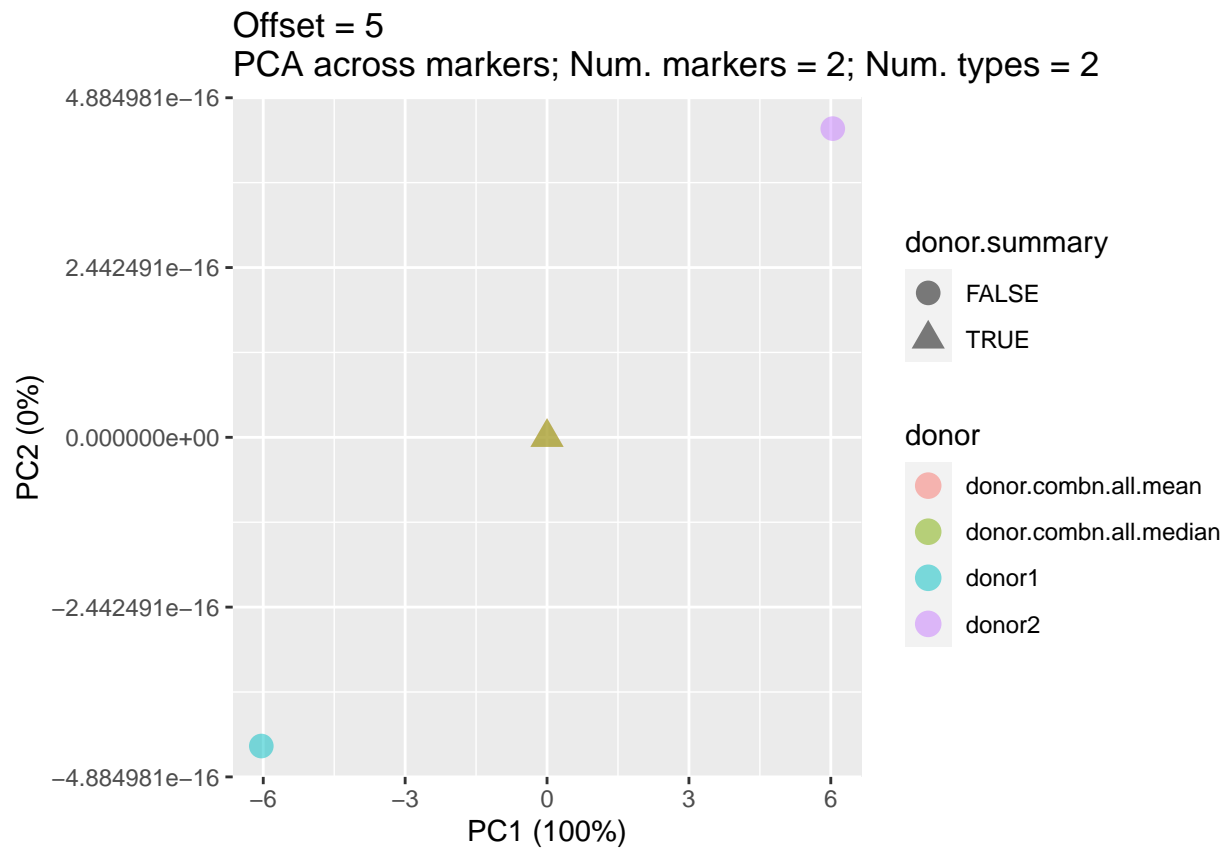
Marker bias plots

PC1 vs. PC2 of donor bias

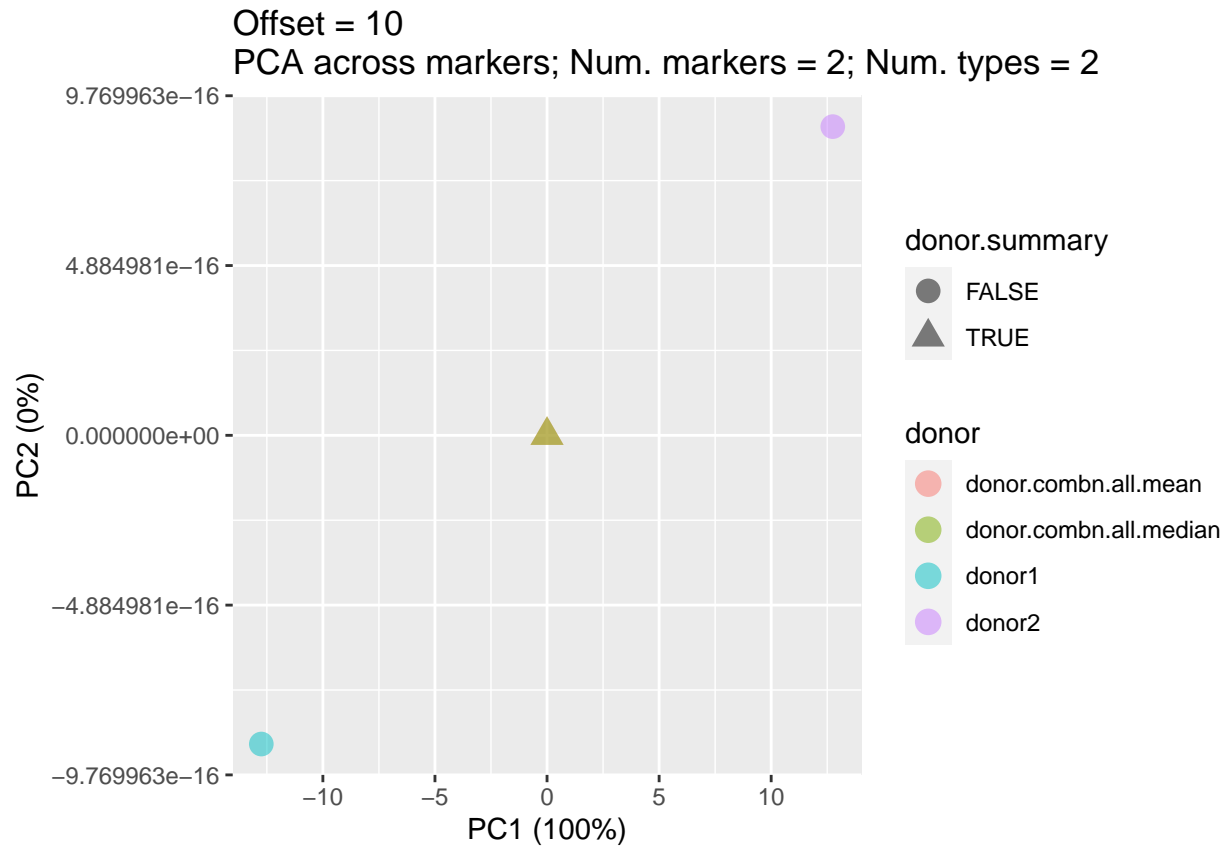
```
lexpt$`offset:2`$lpca.markers$pca.bydonor$scatterplot.pc1.pc2
```



```
lexpt$`offset:5`$lpca.markers$pca.bydonor$scatterplot.pc1.pc2
```

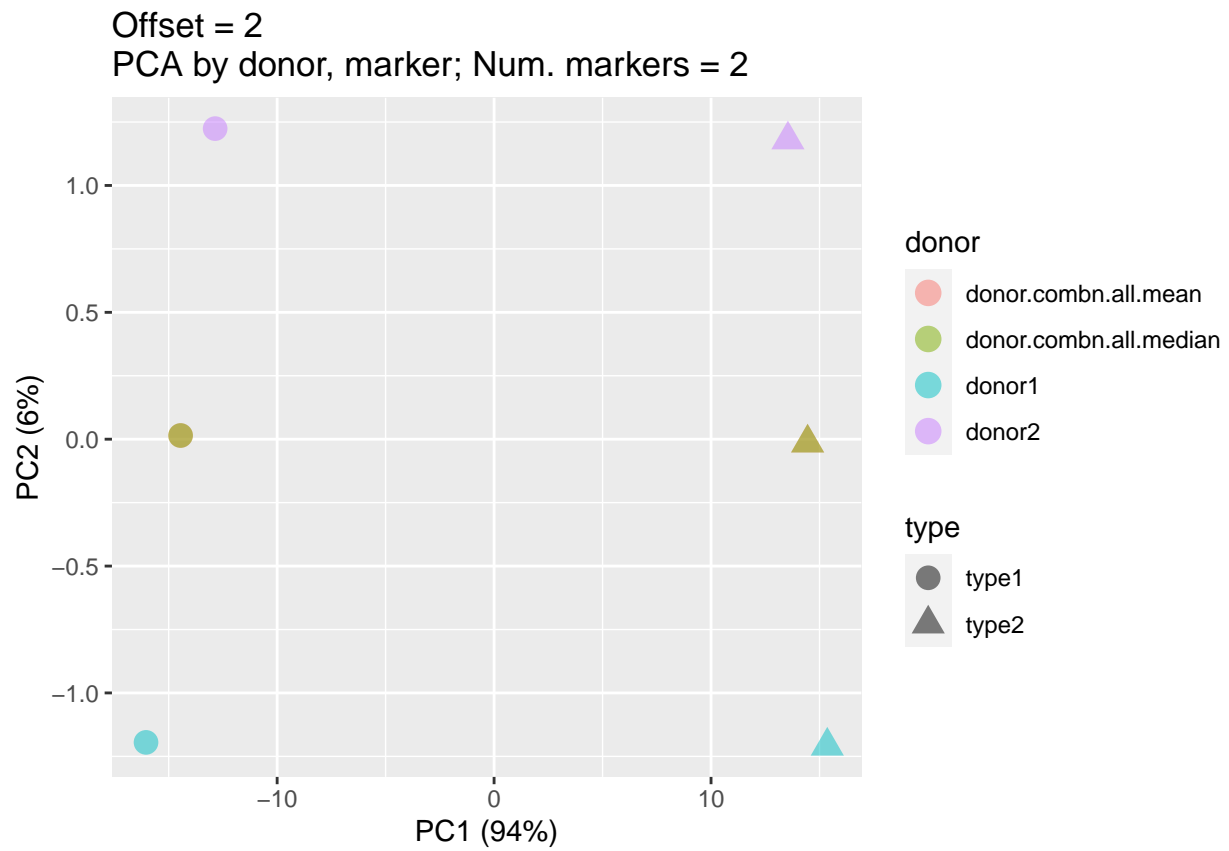


```
lexpt$`offset:10`$lpca.markers$pca.bydonor$scatterplot.pc1.pc2
```



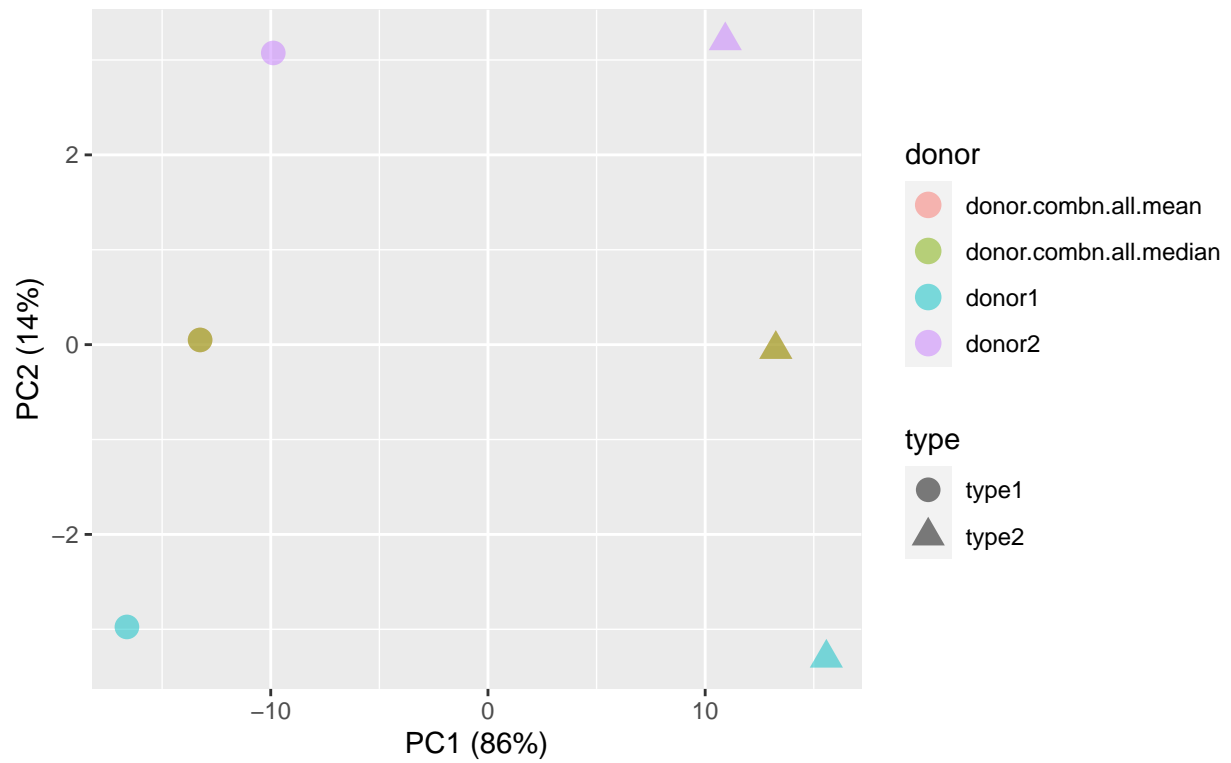
PC1 vs. PC2 of donor, marker variances

```
lexpt$`offset:2`$lpca.markers$pca.bydonortype$scatterplot.pc1.pc2
```

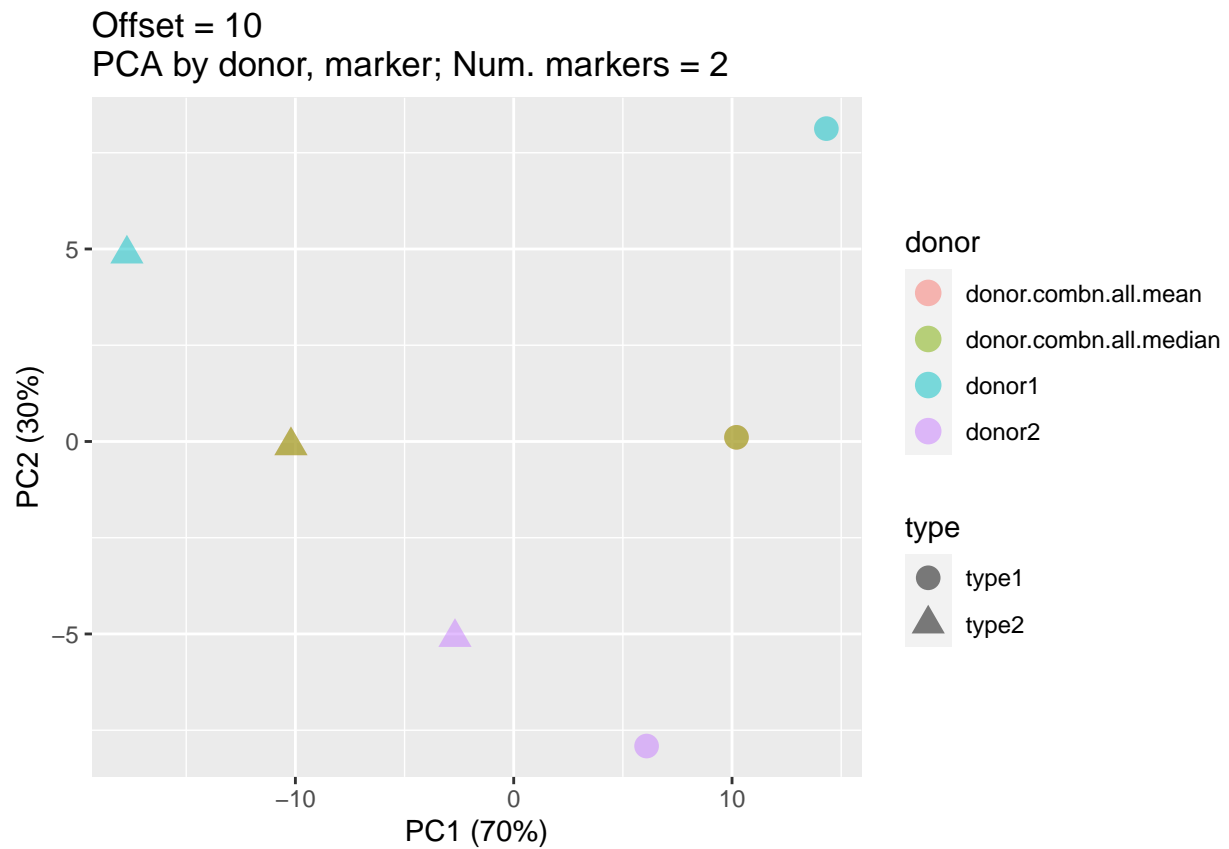


```
lexpt$`offset:5`$lpca.markers$pca.bydonortype$scatterplot.pc1.pc2
```

Offset = 5
 PCA by donor, marker; Num. markers = 2



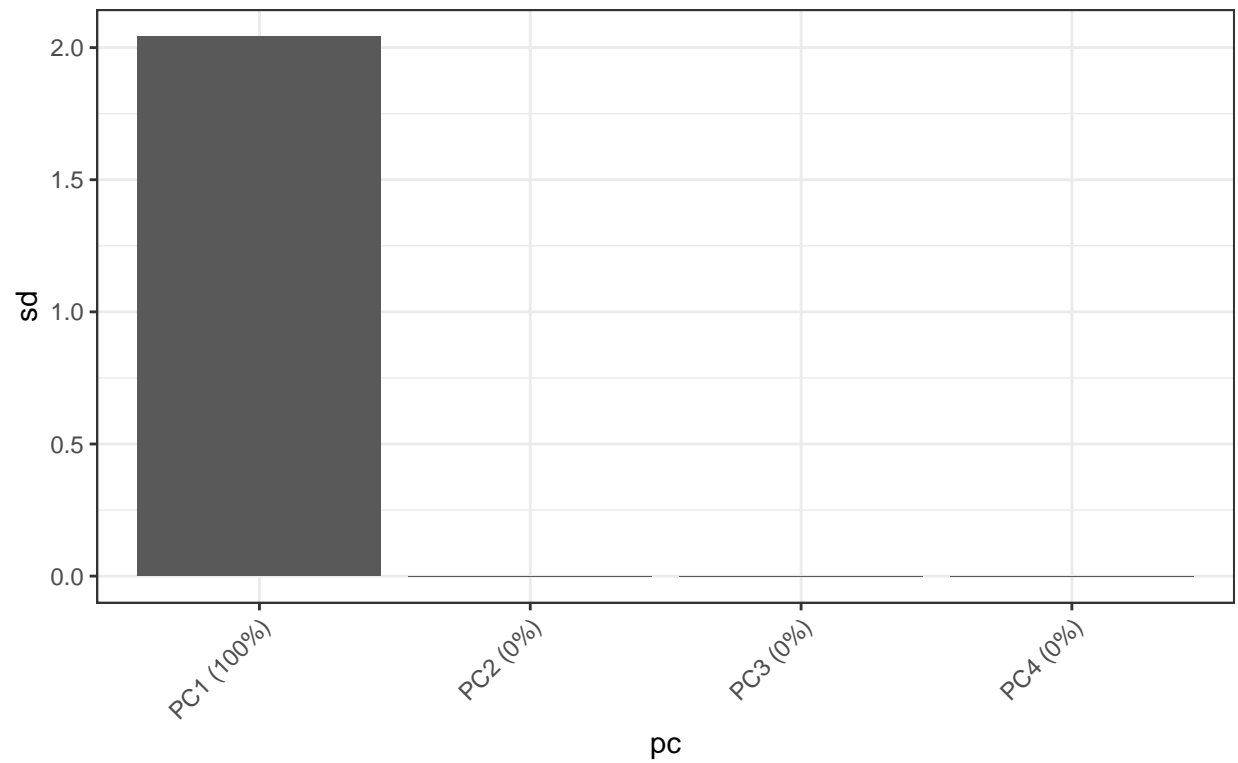
```
lexpt$`offset:10`$lpca.markers$pca.bydonortype$scatterplot.pc1.pc2
```



Screeplots, PCA by donor

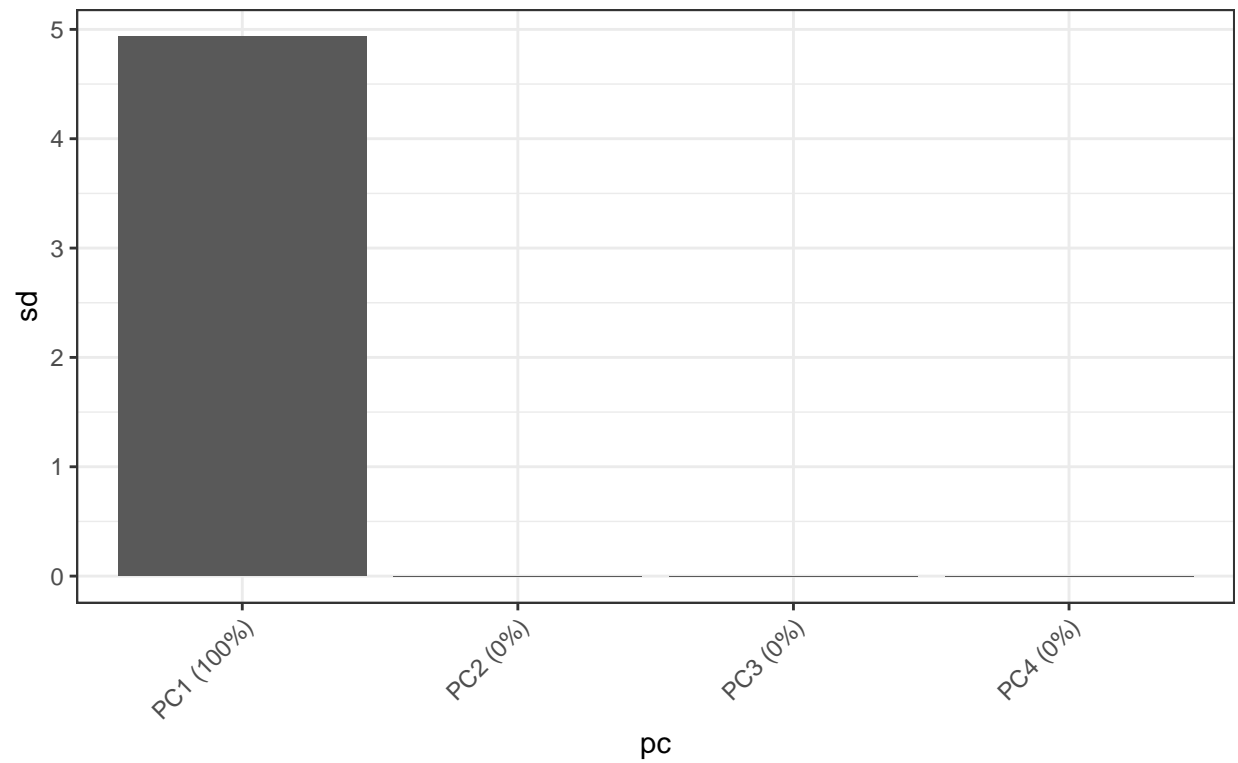
```
lexpt$`offset:2`$lpca.markers$pca.bydonor$screeplot
```


Offset = 2
Screeplot; Num. markers = 4

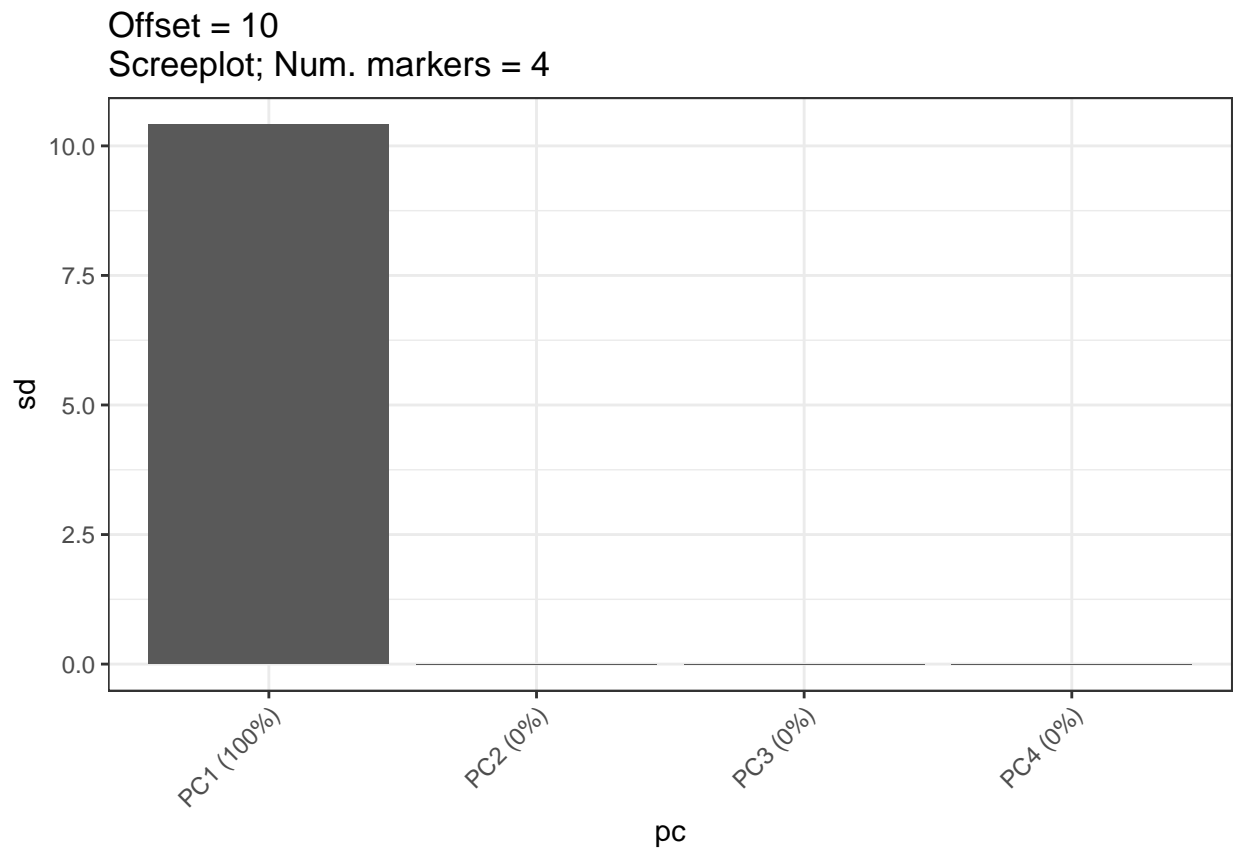


```
lexpt$`offset:5`$lpca.markers$pca.bydonor$screeplot
```

Offset = 5
Screeplot; Num. markers = 4



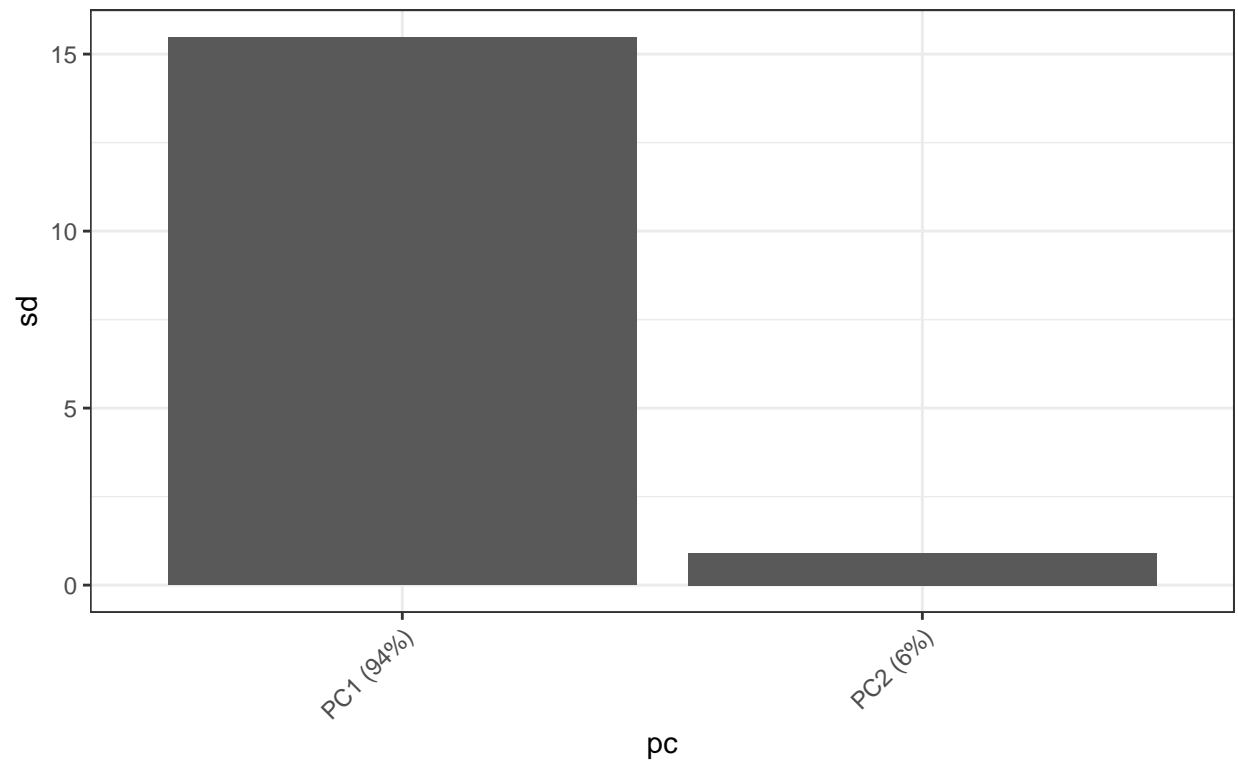
```
lexpt$`offset:10`$lpca.markers$pca.bydonor$screeplot
```



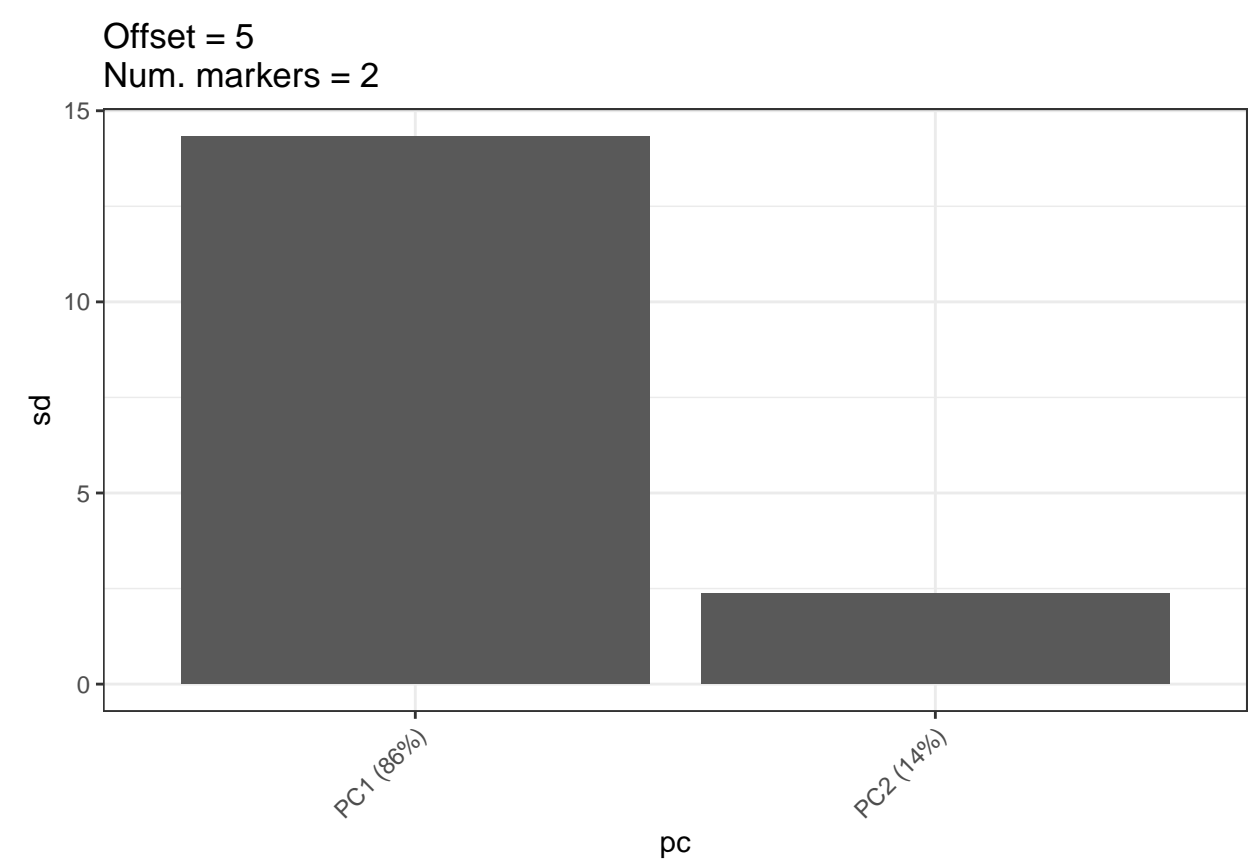
Screeplots, PCA by donor;type

```
lexpt$`offset:2`$lpca.markers$pca.bydonortype$screeplot
```

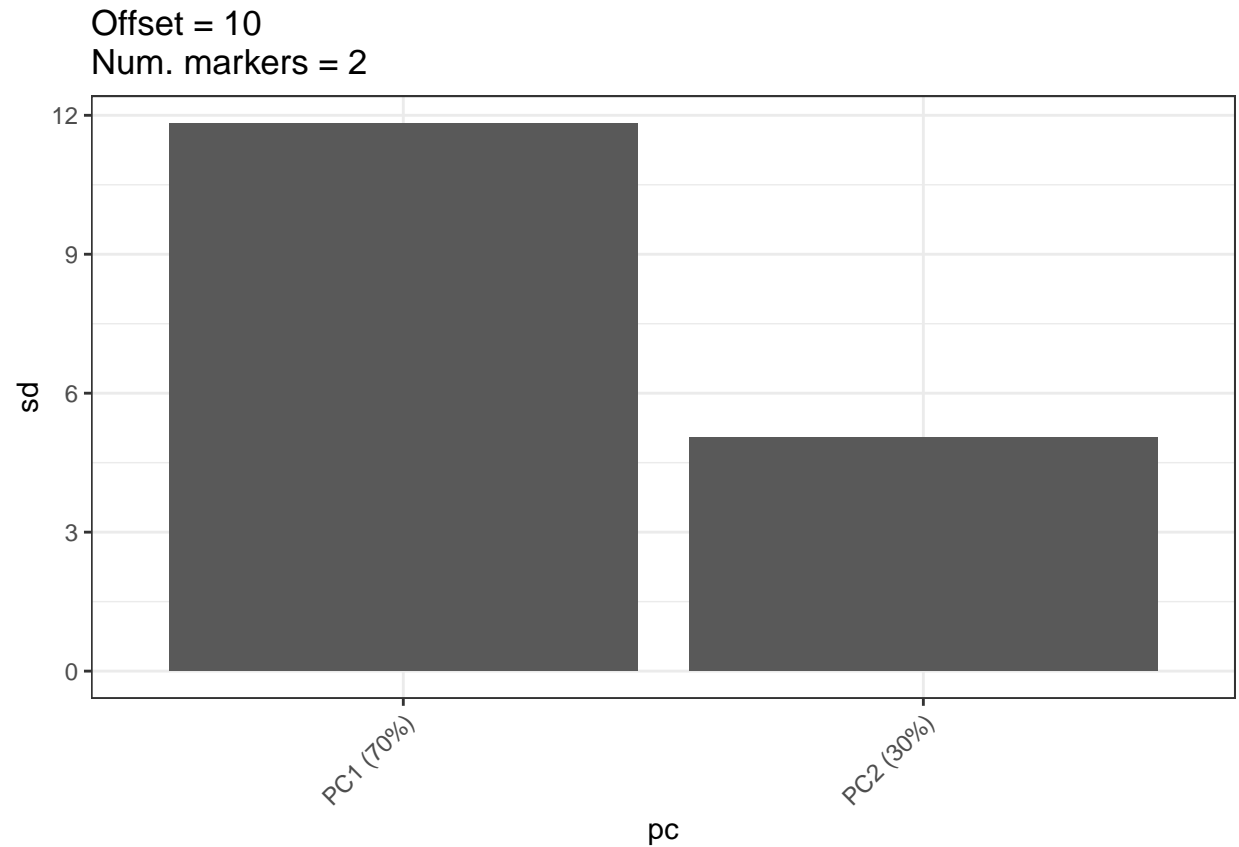
Offset = 2
Num. markers = 2



```
lexpt$`offset:5`$lpca.markers$pca.bydonortype$screeplot
```



```
lexpt$`offset:10`$l pca.markers$pca.bydonortype$screepplot
```



Deconvolution results – all tests

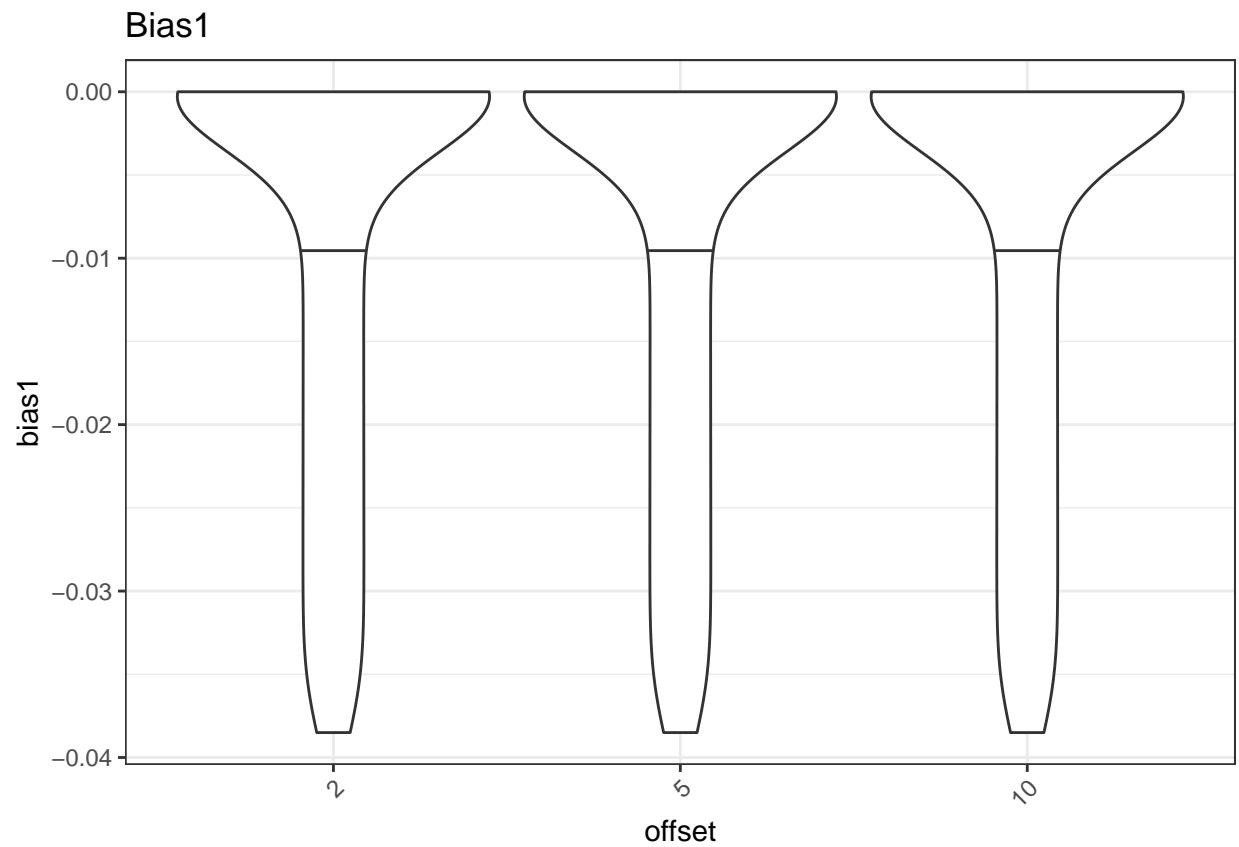
Results table

```
# get all results
dfres <- do.call(rbind, lapply(offsetv, function(offi){
  lres <- lexpt[grepl(paste0("offset:",offi,"$"), names(lexpt))][[1]]$decon.results
  dfresi <- do.call(rbind, lapply(lres, function(resi){resi$dfres}))
  dfresi <- as.data.frame(dfresi)
  dfresi$offset <- offi
  dfresi
})))
```

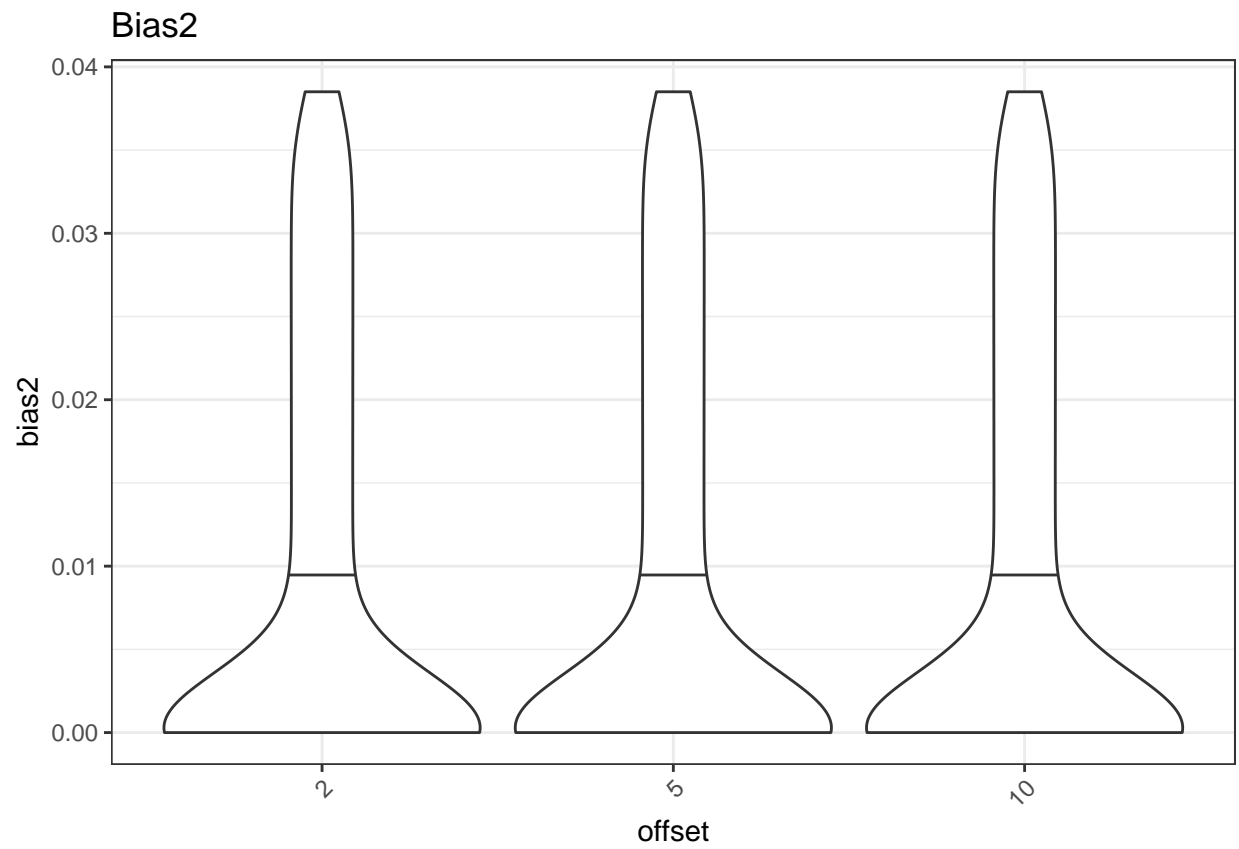
Bias violin plots

```
dfres$offset <- as.factor(dfres$offset)

ggplot(dfres, aes(x = offset, y = bias1)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias1") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

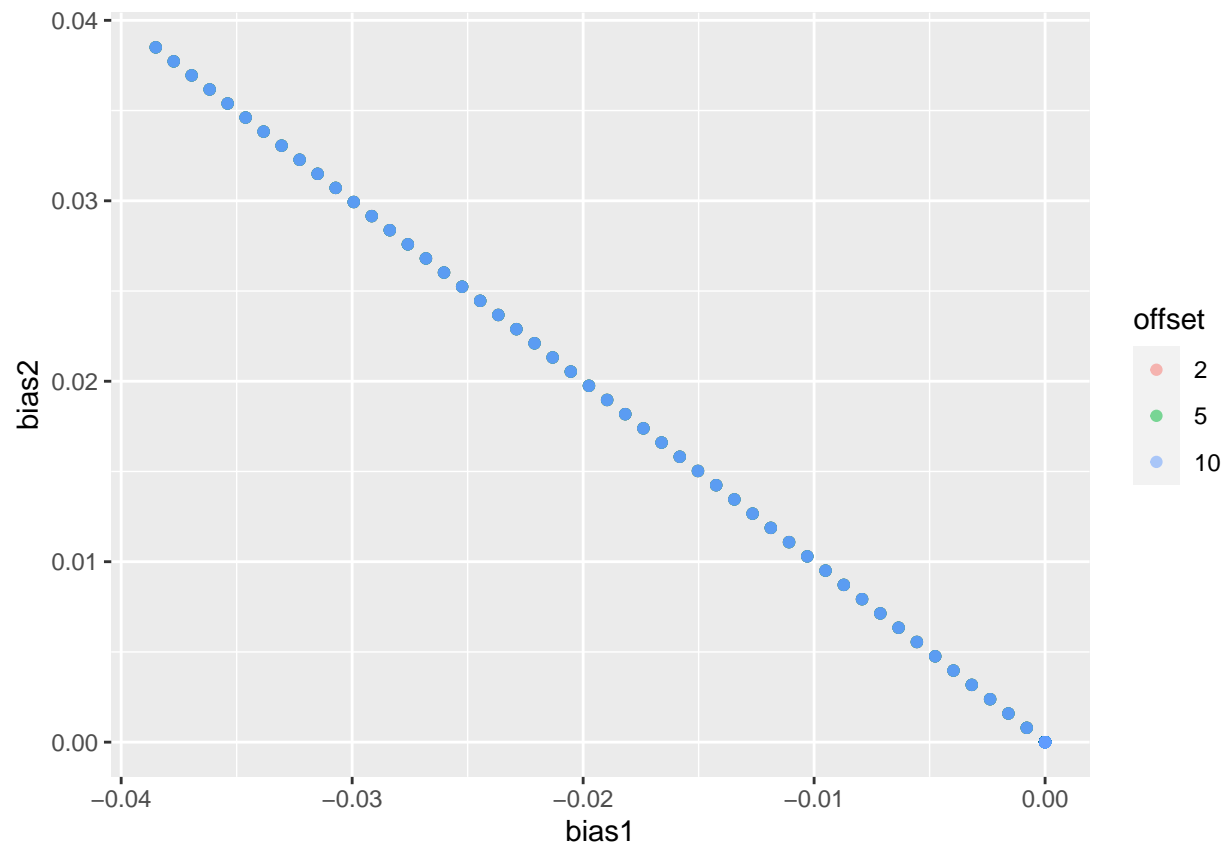


```
ggplot(dfres, aes(x = offset, y = bias2)) + geom_violin(draw_quantiles = 0.5) +  
  theme_bw() + ggtitle("Bias2") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

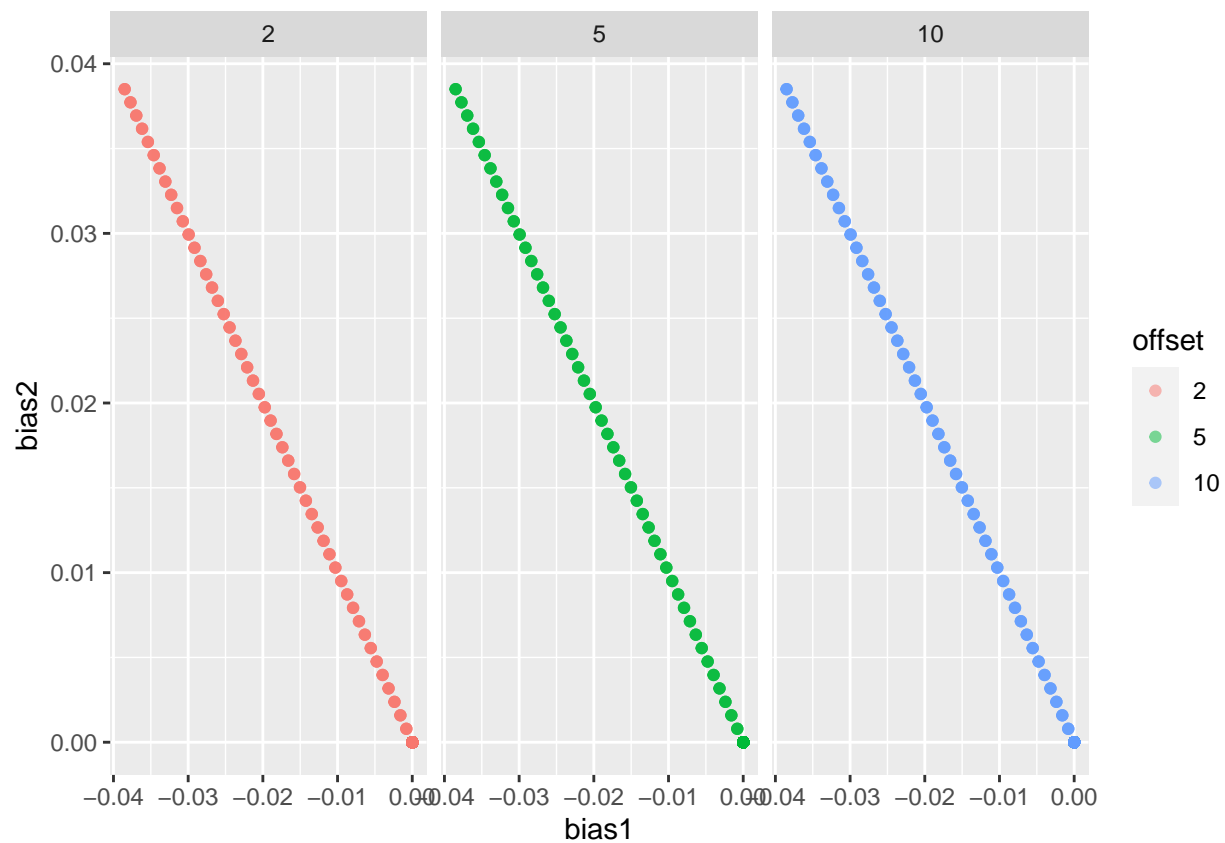


Bias scatterplots

```
ggpt <- ggplot(dfres, aes(x = bias1, y = bias2, color = offset)) + geom_point(alpha = 0.5)  
ggpt
```

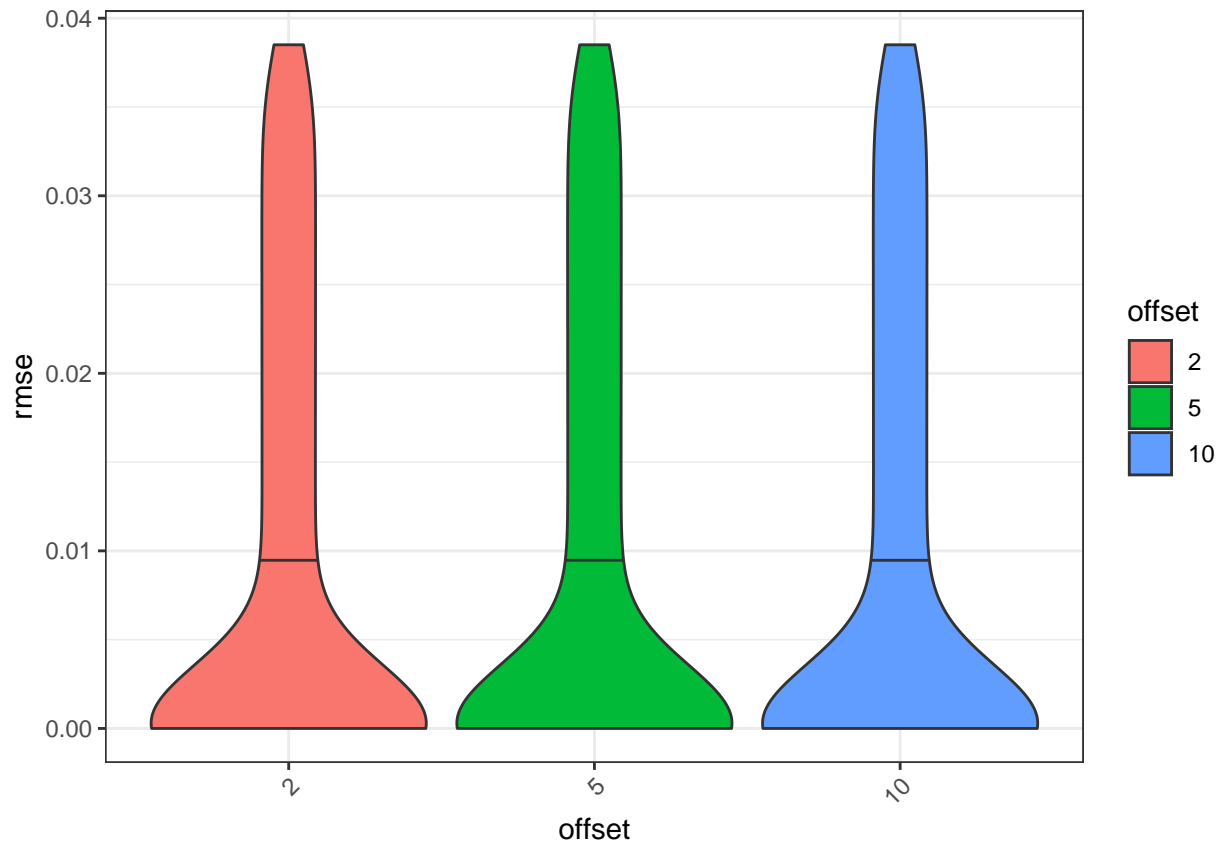



```
ggpt + facet_wrap(~offset)
```



RMSE violin plot

```
ggplot(dfres, aes(x = offset, y = rmse, fill = offset)) +
  geom_violin(draw_quantiles = 0.5) + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Deconvolution results – without S transformations

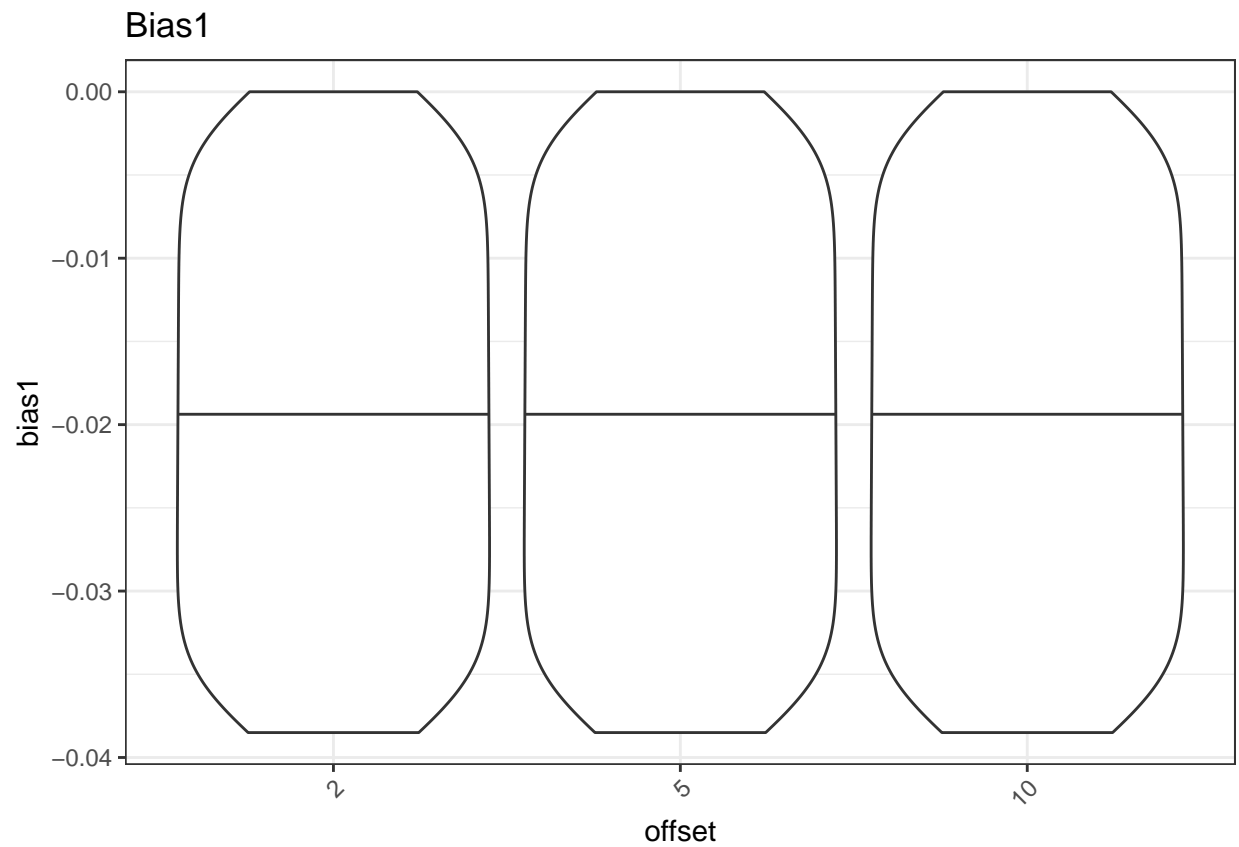
Results table

```
dfres.all <- dfres
dfres <- dfres.all[dfres.all$zs_transform==FALSE,]
```

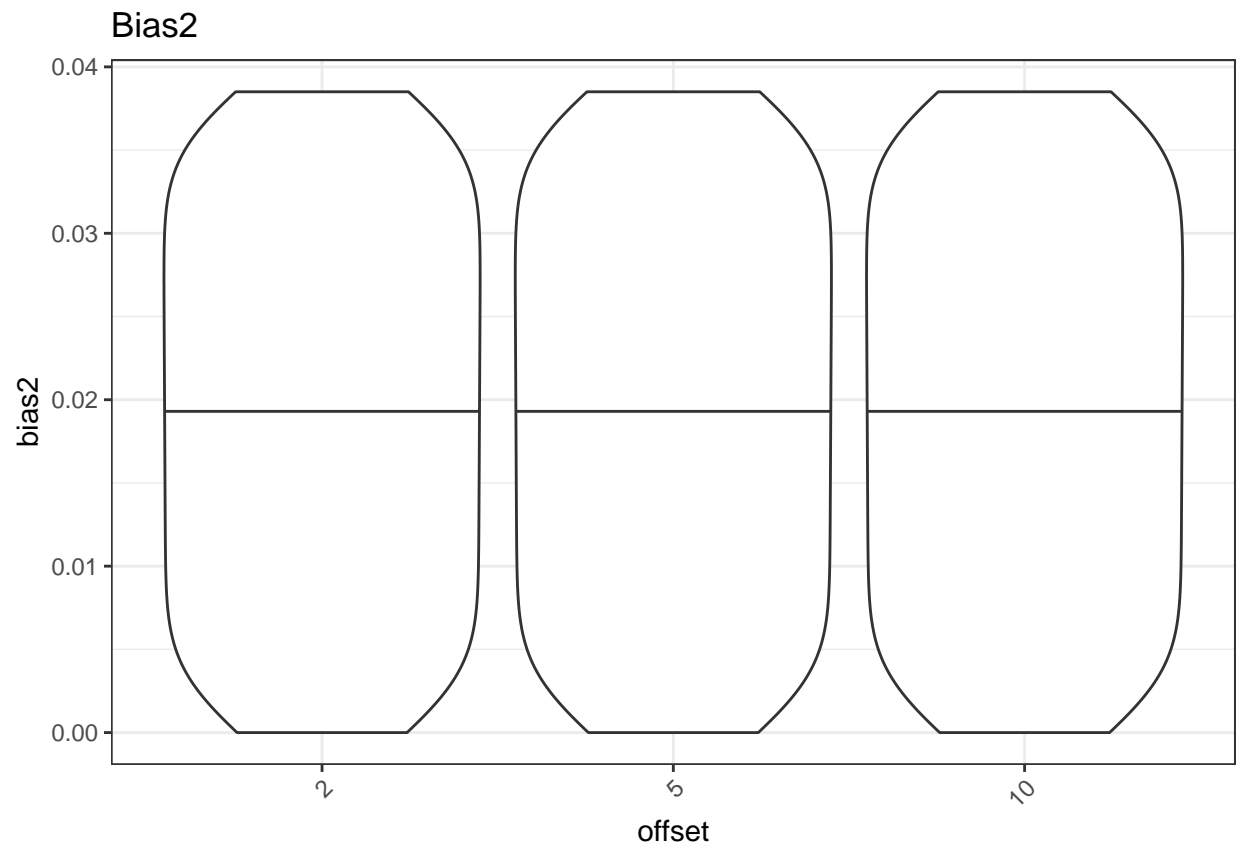
Bias violin plots

```
dfres$offset <- as.factor(dfres$offset)

ggplot(dfres, aes(x = offset, y = bias1)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias1") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

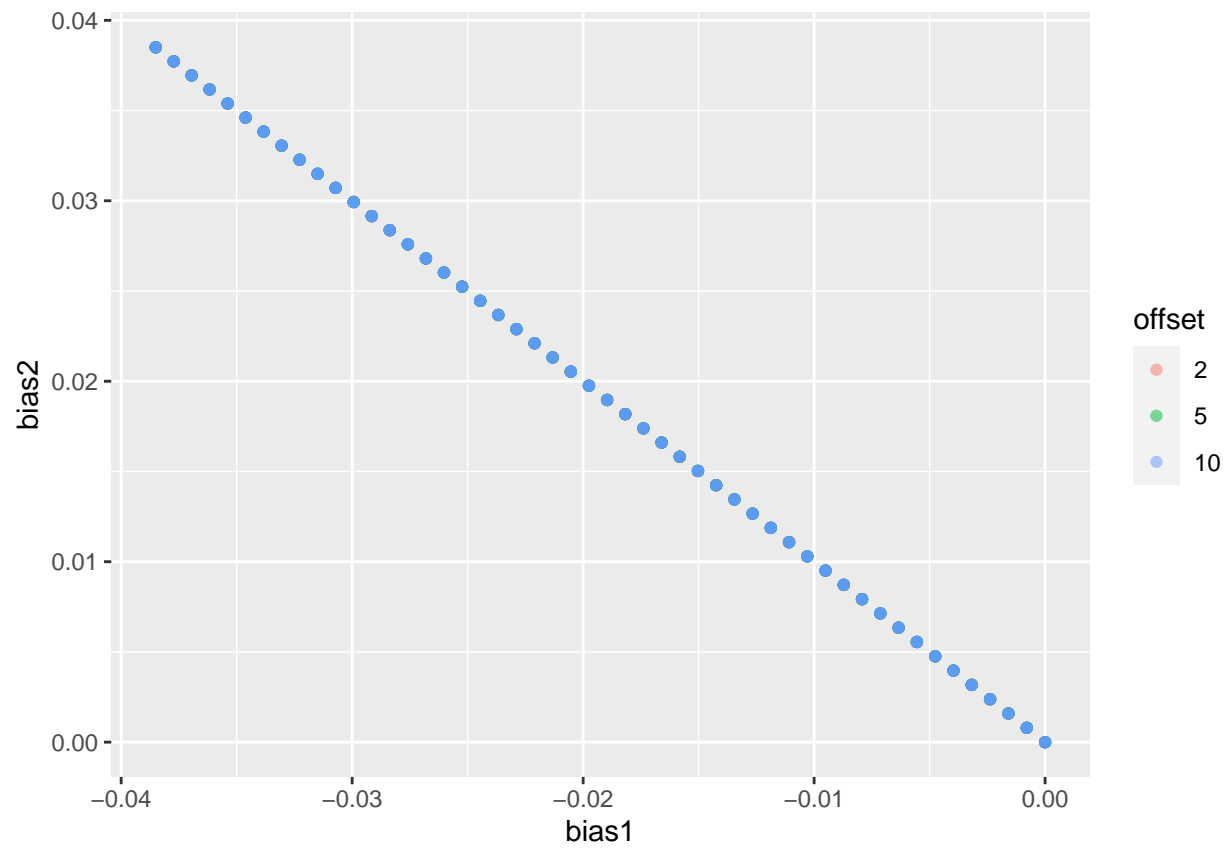


```
ggplot(dfres, aes(x = offset, y = bias2)) + geom_violin(draw_quantiles = 0.5) +  
  theme_bw() + ggtitle("Bias2") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

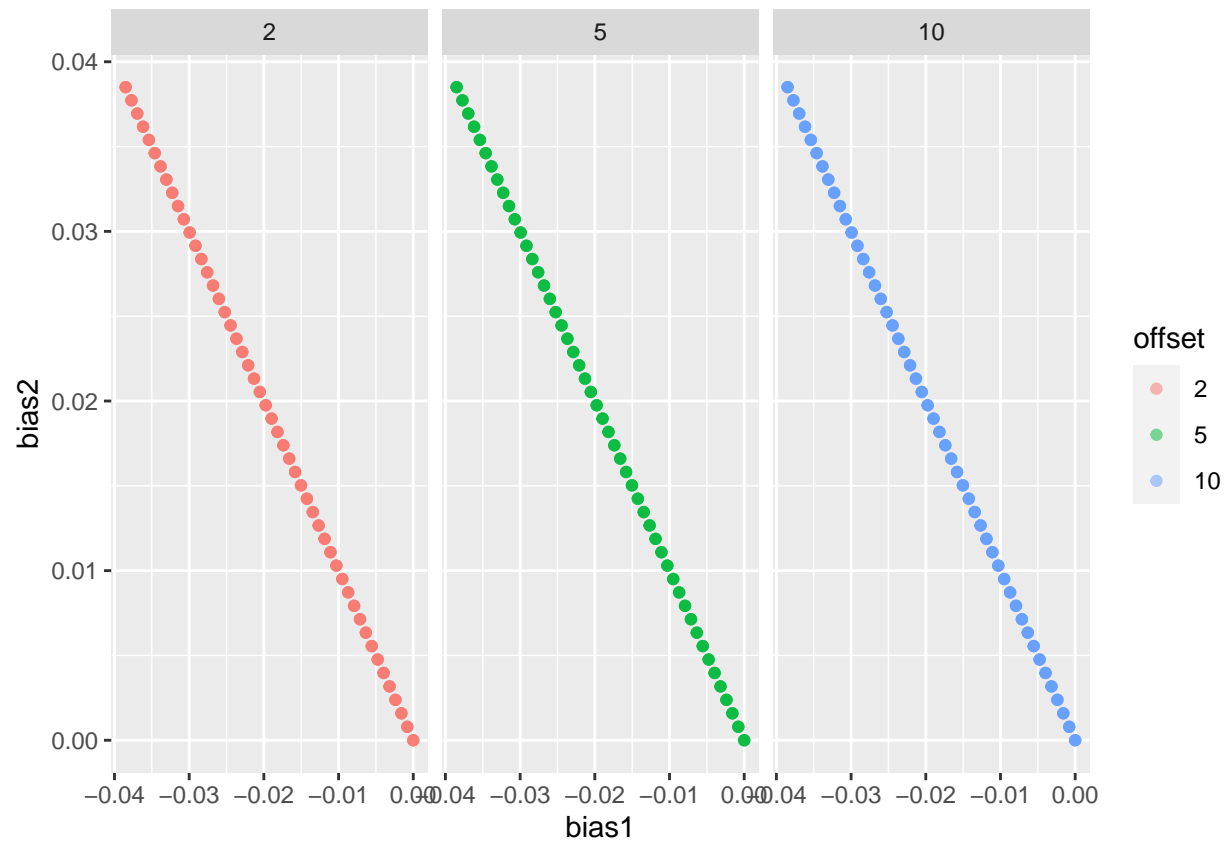


Bias scatterplots

```
ggpt <- ggplot(dfres, aes(x = bias1, y = bias2, color = offset)) + geom_point(alpha = 0.5)  
ggpt
```

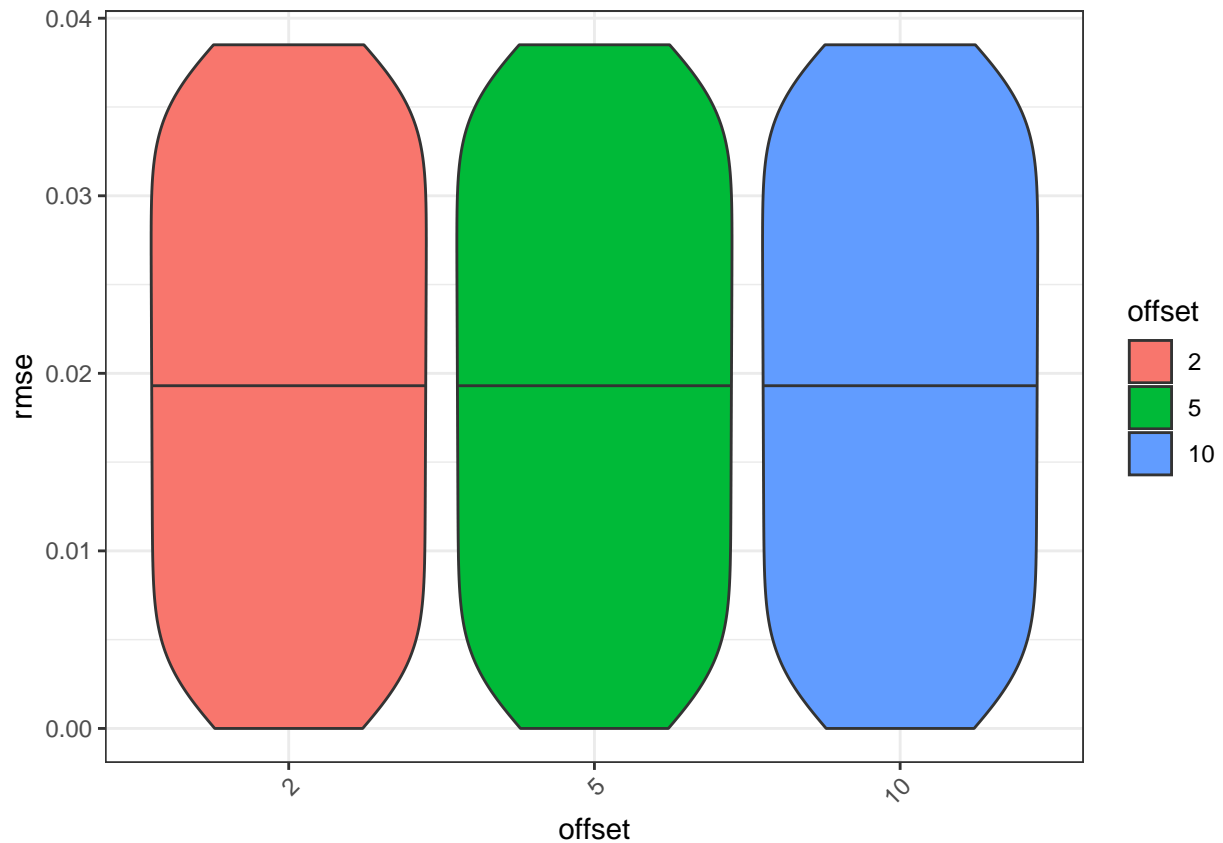


```
ggpt + facet_wrap(~offset)
```



RMSE violin plot

```
ggplot(dfres, aes(x = offset, y = rmse, fill = offset)) +
  geom_violin(draw_quantiles = 0.5) + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



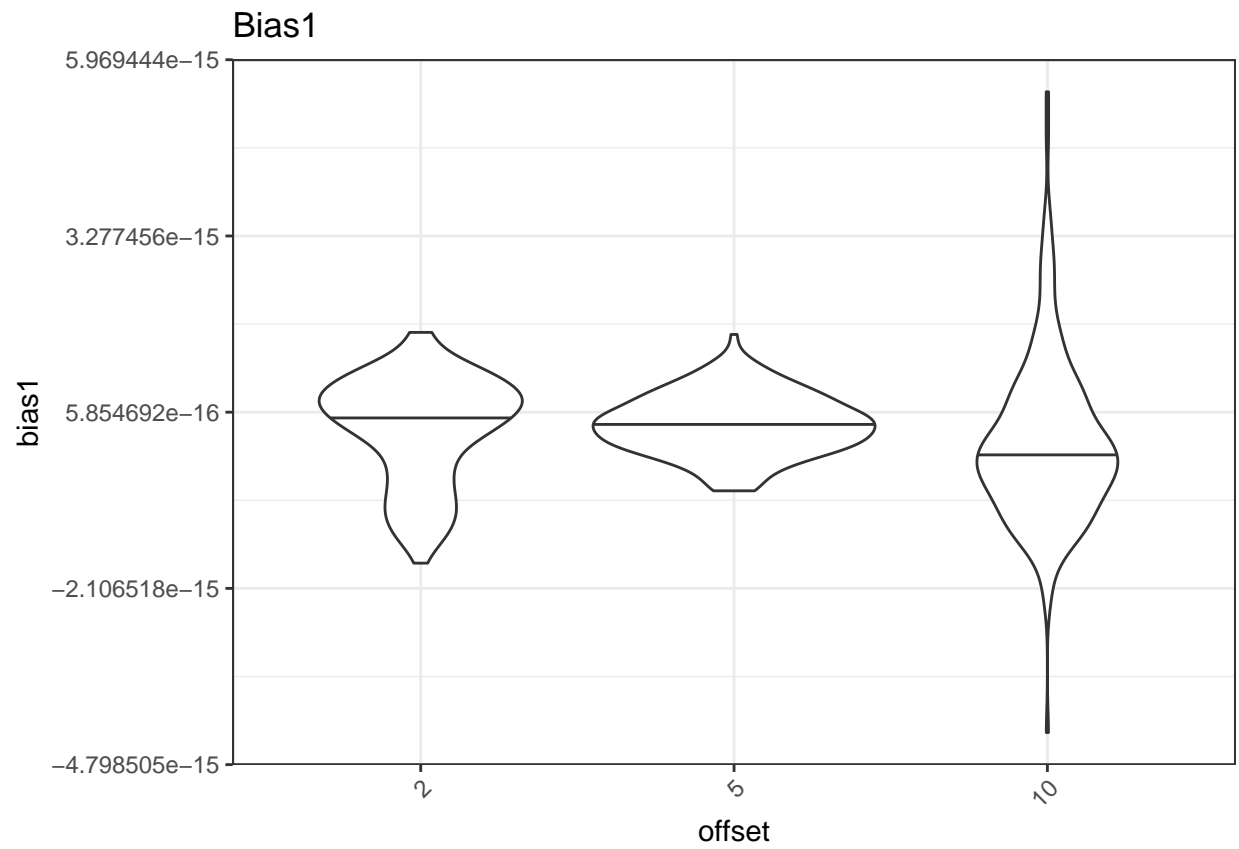
Deconvolution results – with S transformations

```
dfres <- dfres.all[dfres.all$zs_transform==TRUE,]
```

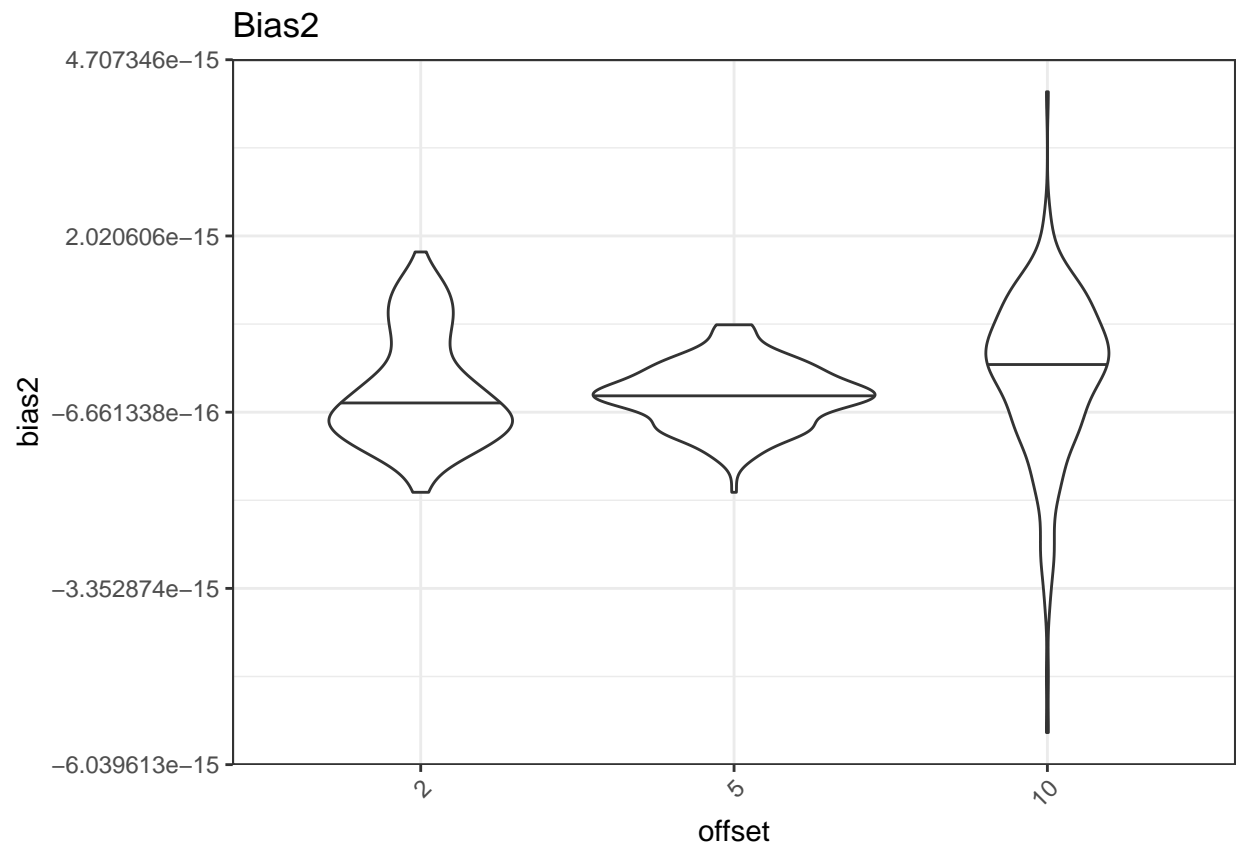
Bias violin plots

```
dfres$offset <- as.factor(dfres$offset)

ggplot(dfres, aes(x = offset, y = bias1)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias1") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

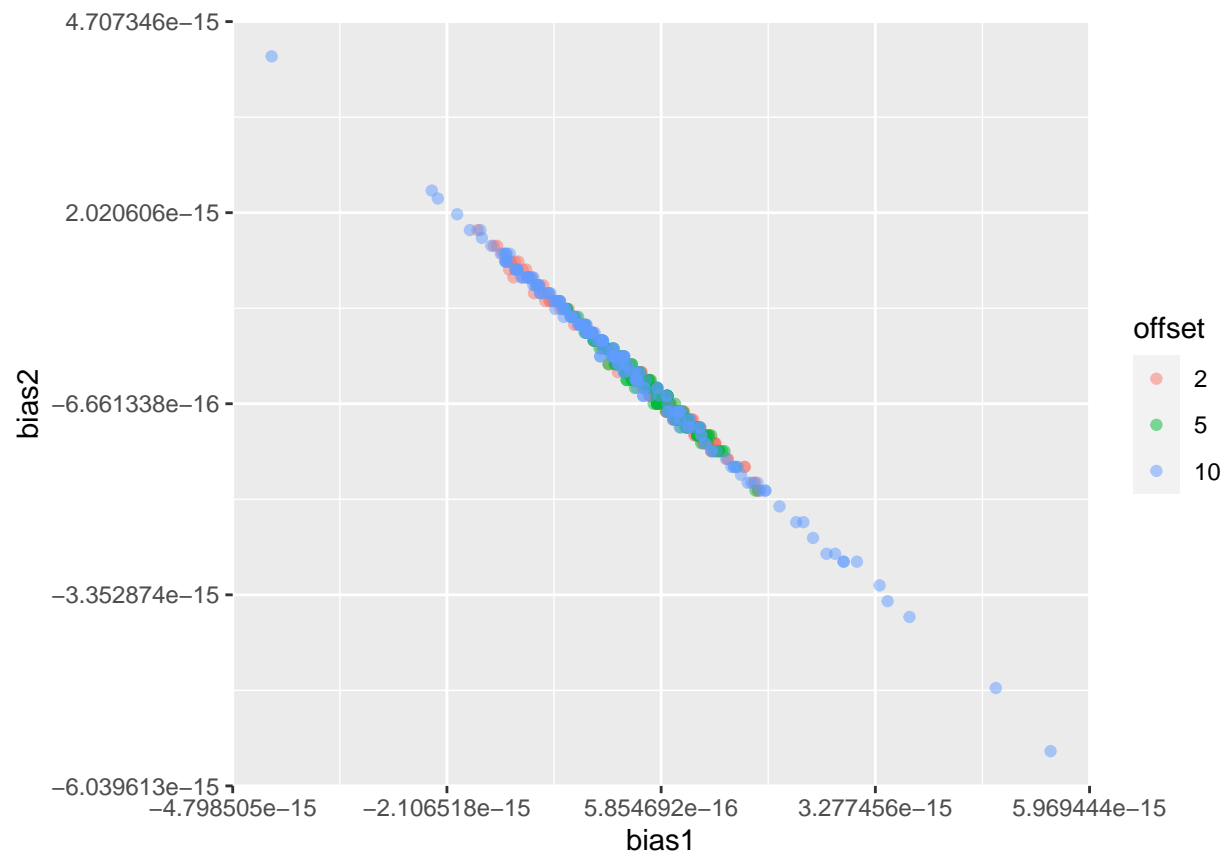



```
ggplot(dfres, aes(x = offset, y = bias2)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias2") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

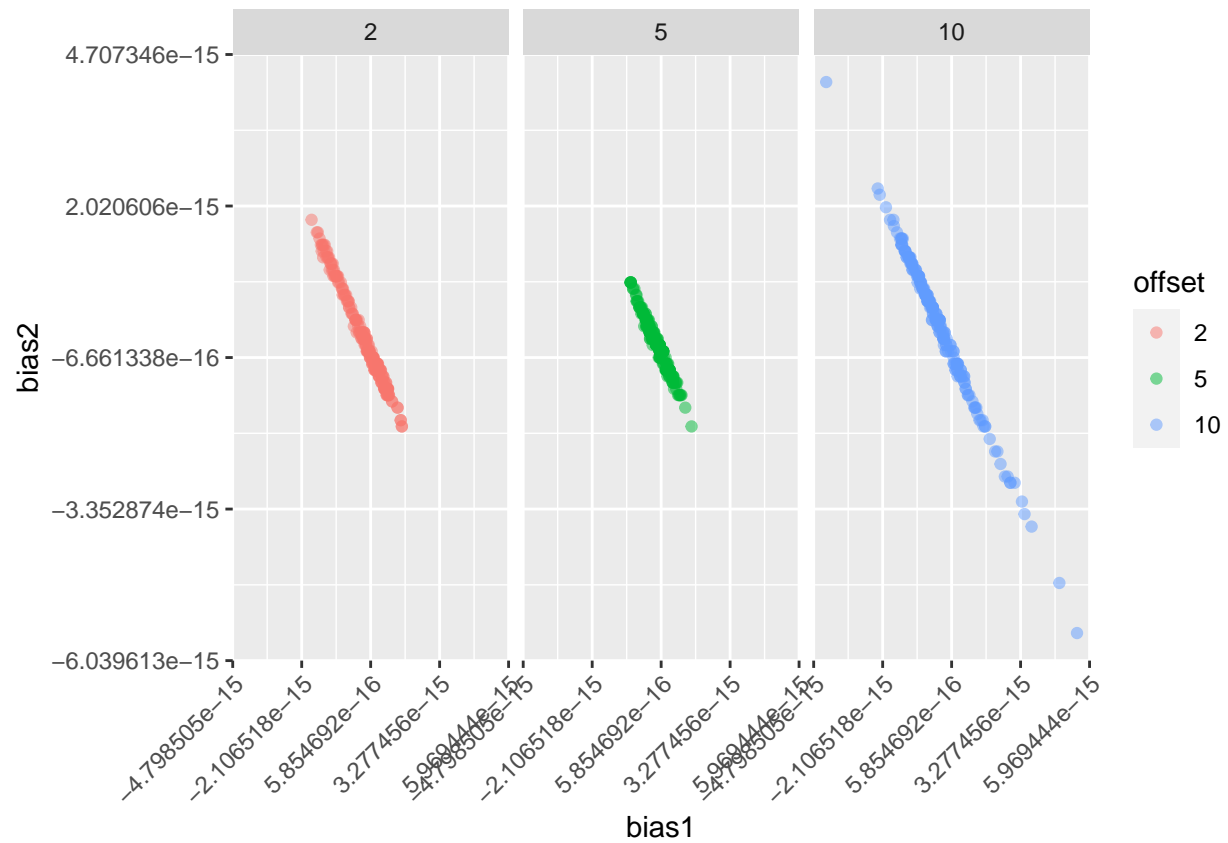


Bias scatterplots

```
ggpt <- ggplot(dfres, aes(x = bias1, y = bias2, color = offset)) + geom_point(alpha = 0.5)
ggpt
```

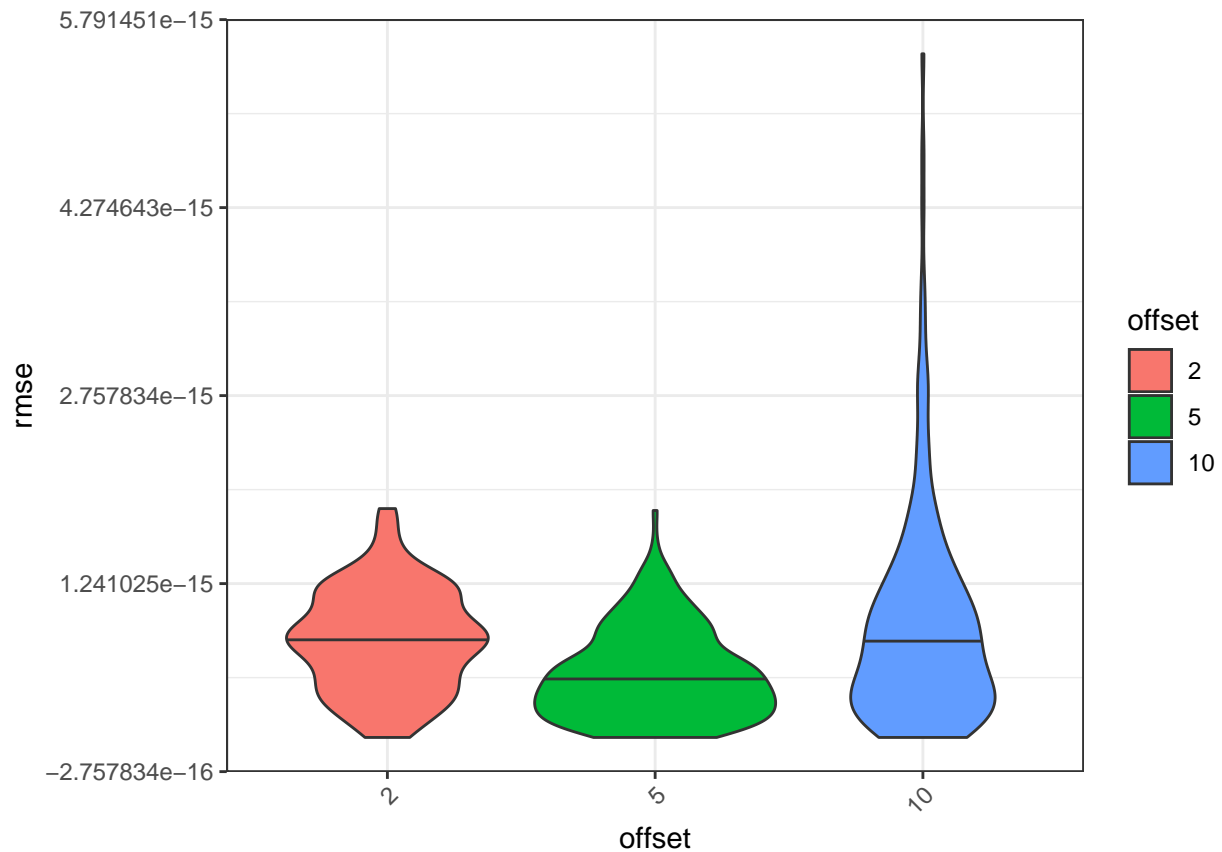


```
ggpt + facet_wrap(~offset) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



RMSE violin plot

```
ggplot(dfres, aes(x = offset, y = rmse, fill = offset)) +
  geom_violin(draw_quantiles = 0.5) + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Experiment series 1: ndonors = 10

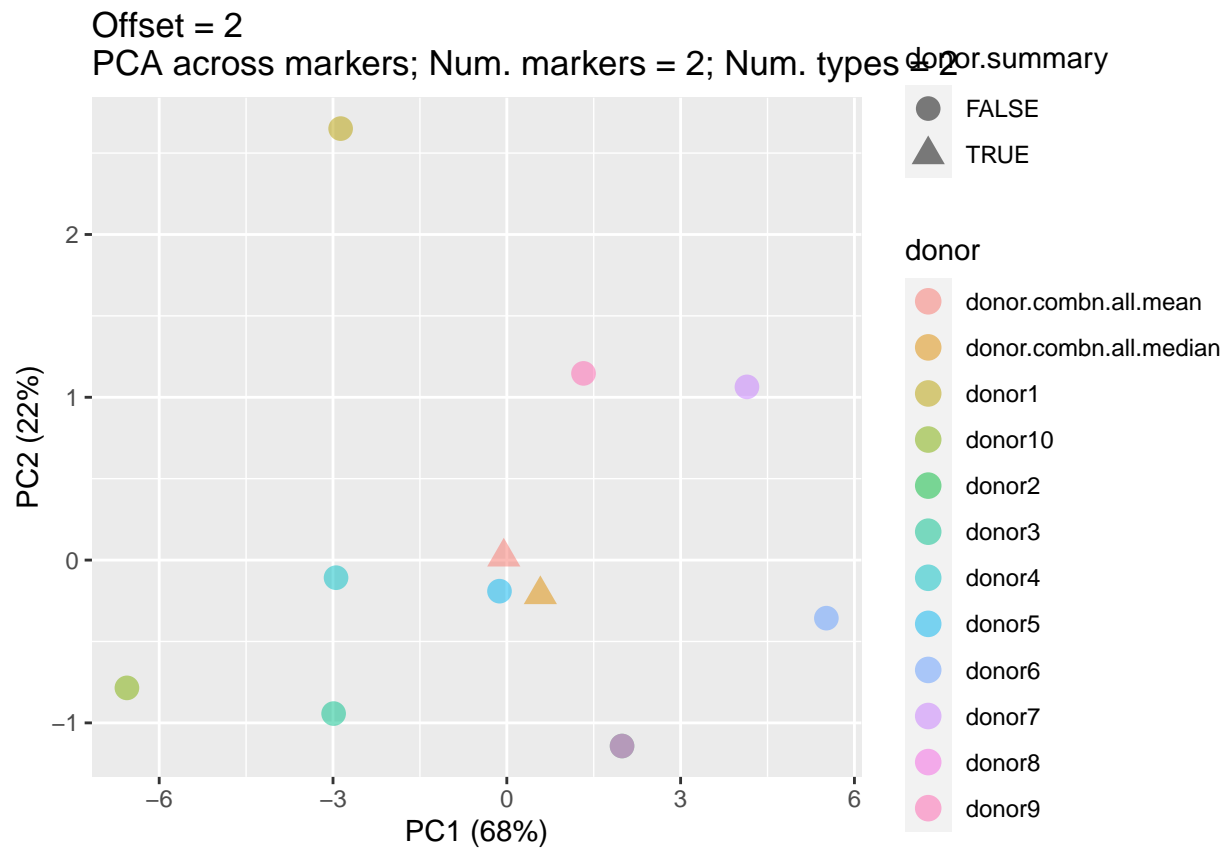
Building on the simple 2-donors case above, we now simulate 10 donors with the same progression in random offset increase.

```
ndonor = 10
gindexv = c(1, 2)
offsetv <- c(2, 5, 10)
num.sim <- 50
lexpt <- lapply(offsetv, function(offi){
  title.append <- paste0("Offset = ", offi, "\n")
  donor_marker_experiment(gindexv = c(1,2), num.sim = num.sim, ndonor = ndonor,
    plot.title.append = title.append,
    sd.offset.pos = offi,
    sd.offset.neg = offi)
})
names(lexpt) <- paste0("offset:", offsetv)
```

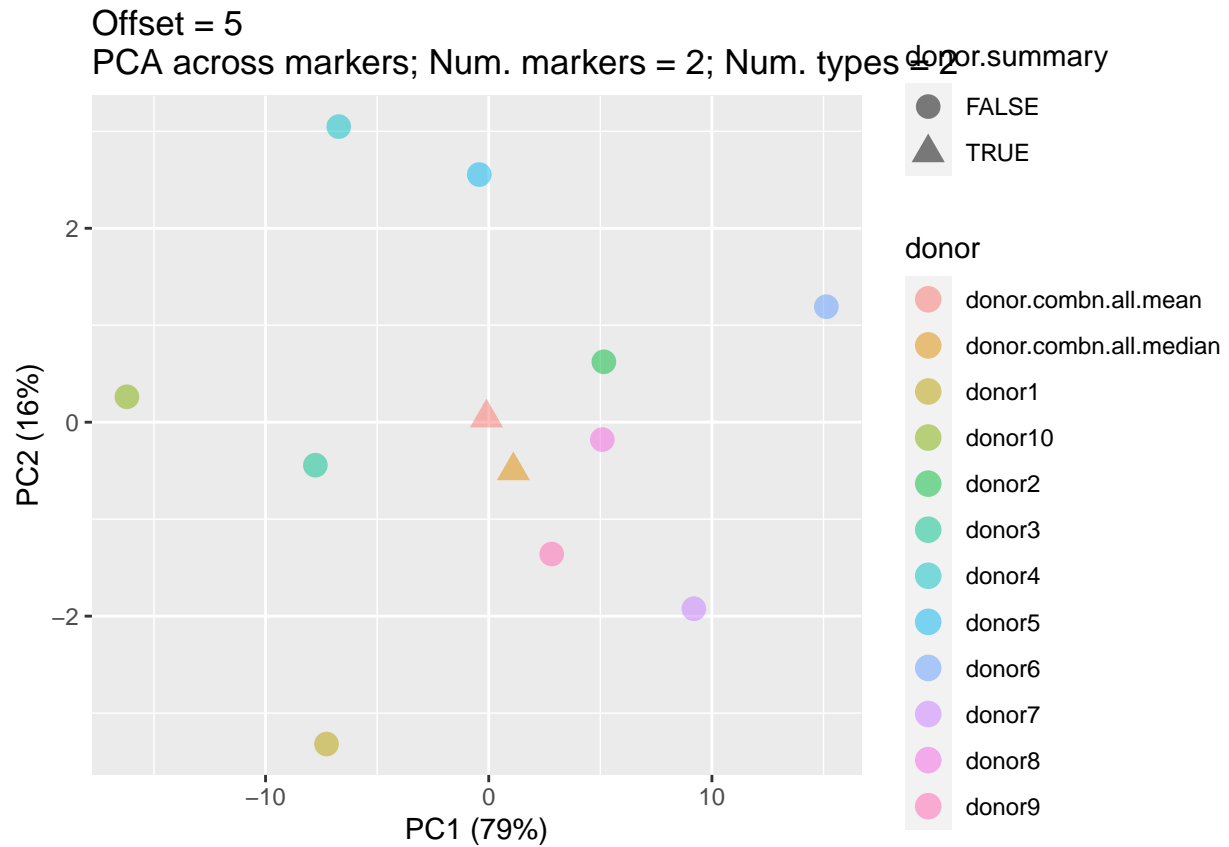
Marker bias plots

PC1 vs. PC2 of donor bias

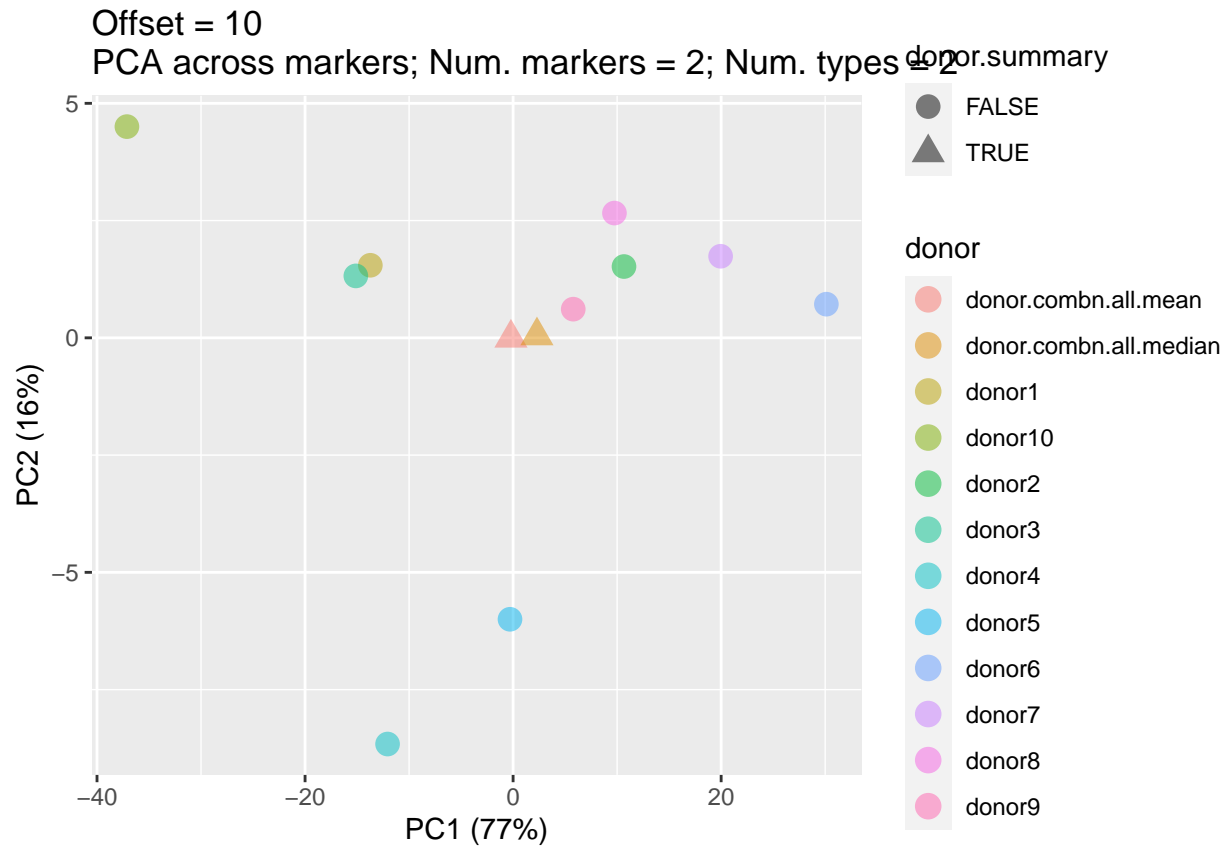
```
lexpt$`offset:2`$lpca.markers$pca.bydonor$scatterplot.pc1.pc2
```



```
lexpt$`offset:5`$lpca.markers$pca.bydonor$scatterplot.pc1.pc2
```

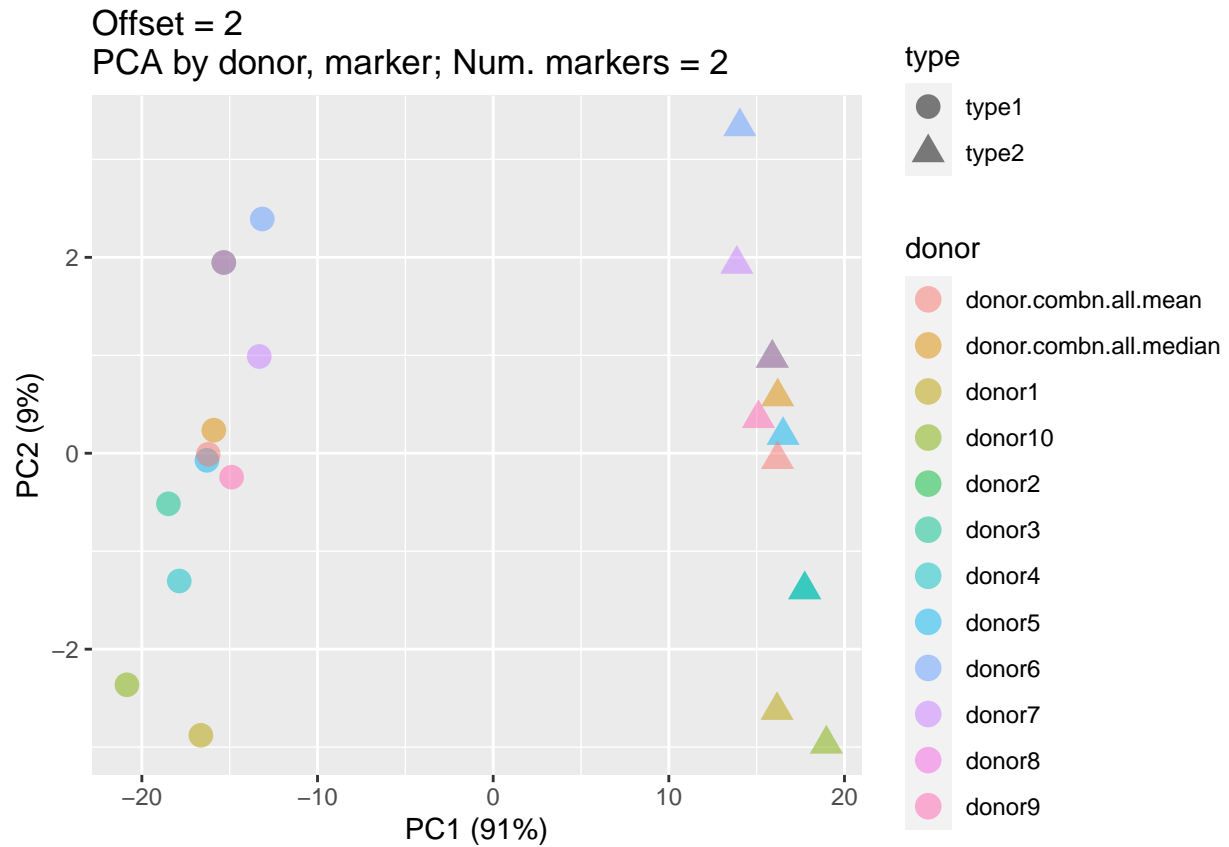


```
lexpt$`offset:10`$lpc.markers$pc.bydonor$scatterplot.pc1.pc2
```

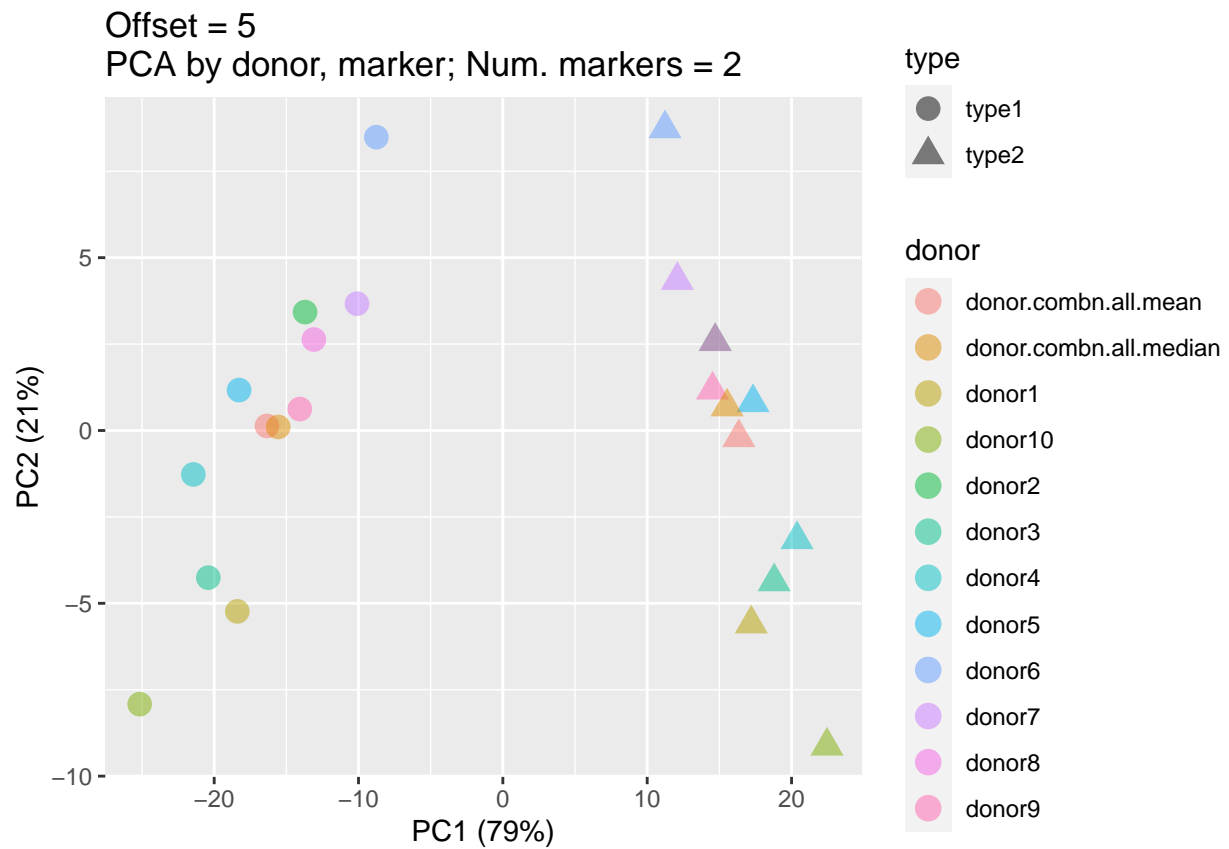


PC1 vs. PC2 of donor, marker variances

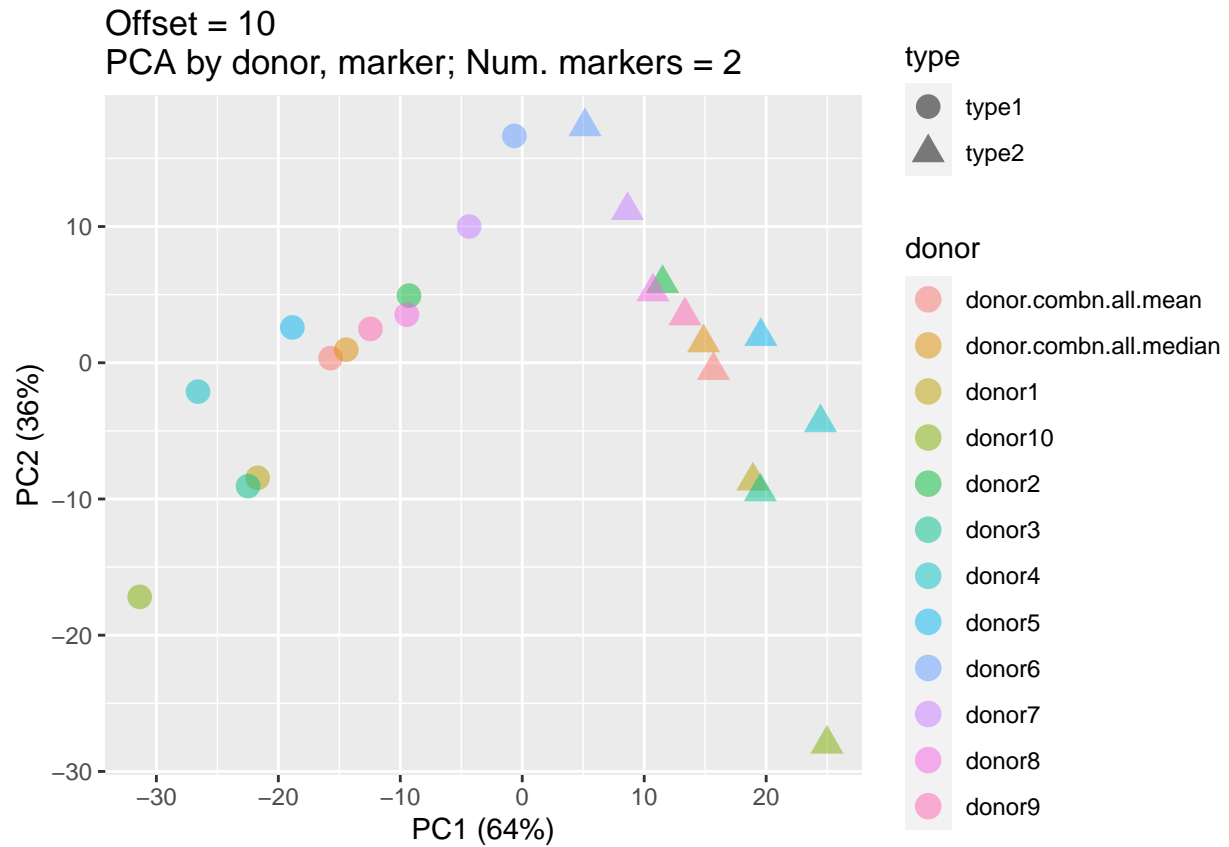
```
lexpt$`offset:2`$lpca.markers$pca.bydonortype$scatterplot.pc1.pc2
```

```
lexpt$`offset:5`$lpca.markers$pca.bydonortype$scatterplot.pc1.pc2
```



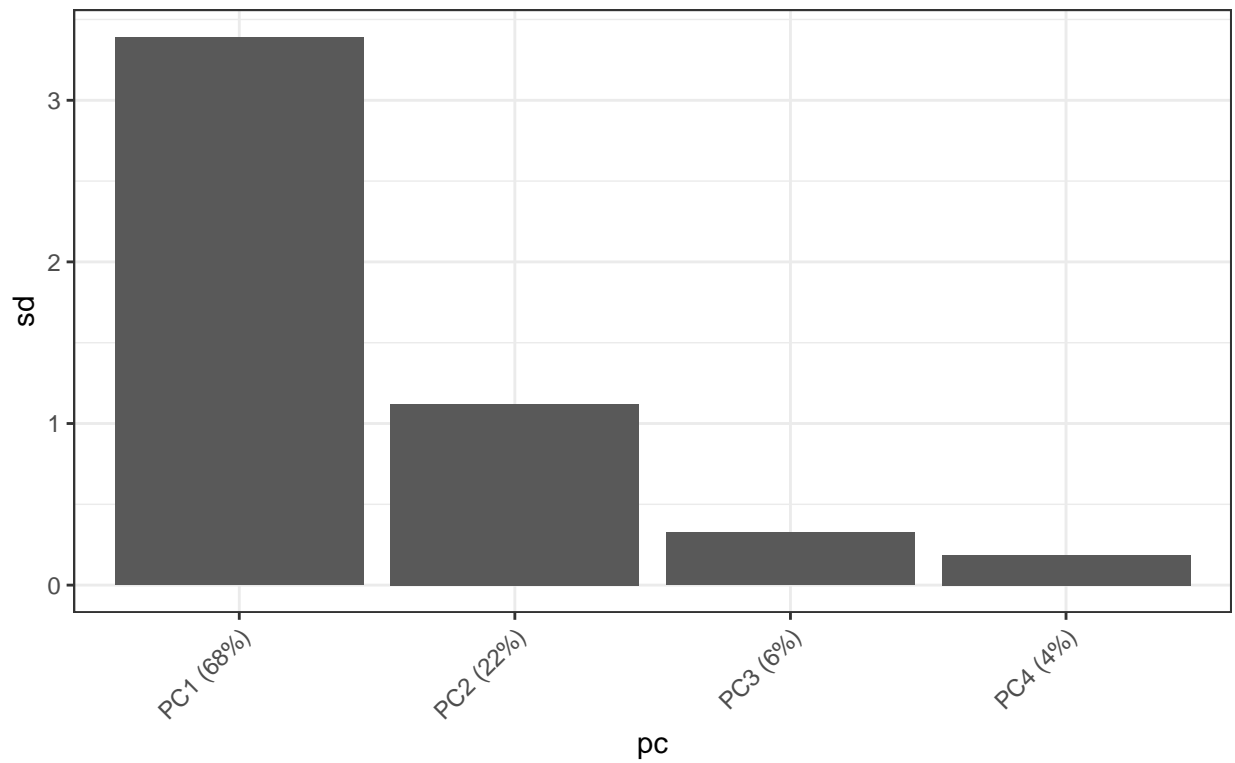
```
lexpt$`offset:10`$lpc.markers$pca.bydonortype$scatterplot.pc1.pc2
```



ScreepLOTS, PCA by donor

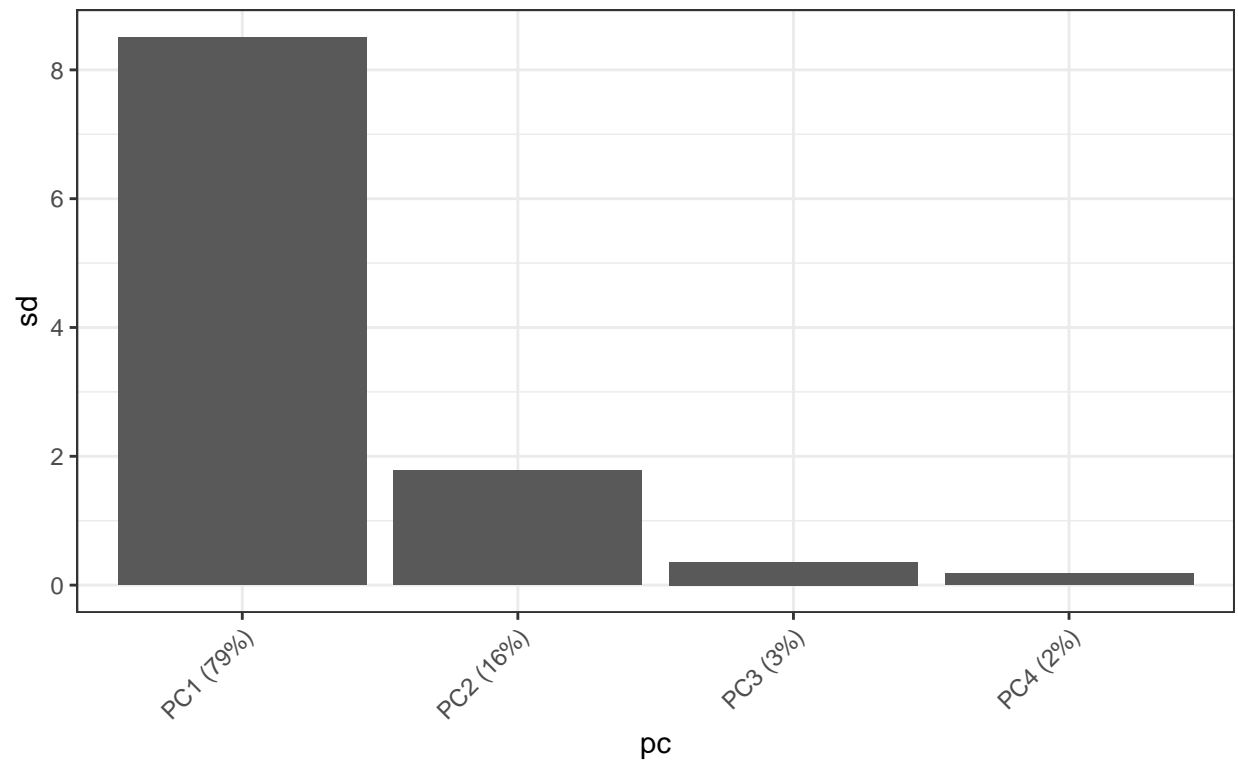
```
lexpt$`offset:2`$lpca.markers$pca.bydonor$screepLOT
```

Offset = 2
Screeplot; Num. markers = 4



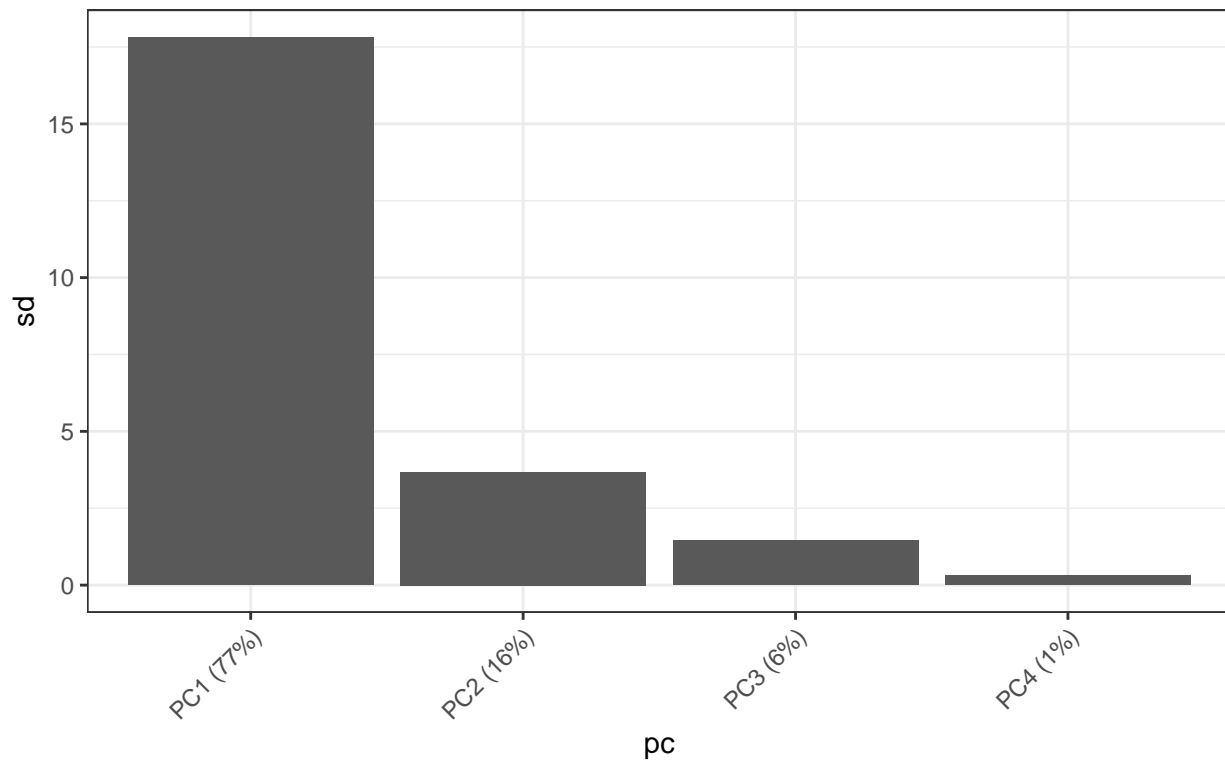
```
lexpt$`offset:5`$lpca.markers$pca.bydonor$screeplot
```

Offset = 5
Screeplot; Num. markers = 4



```
lexpt$`offset:10`$lpca.markers$pca.bydonor$screeplot
```

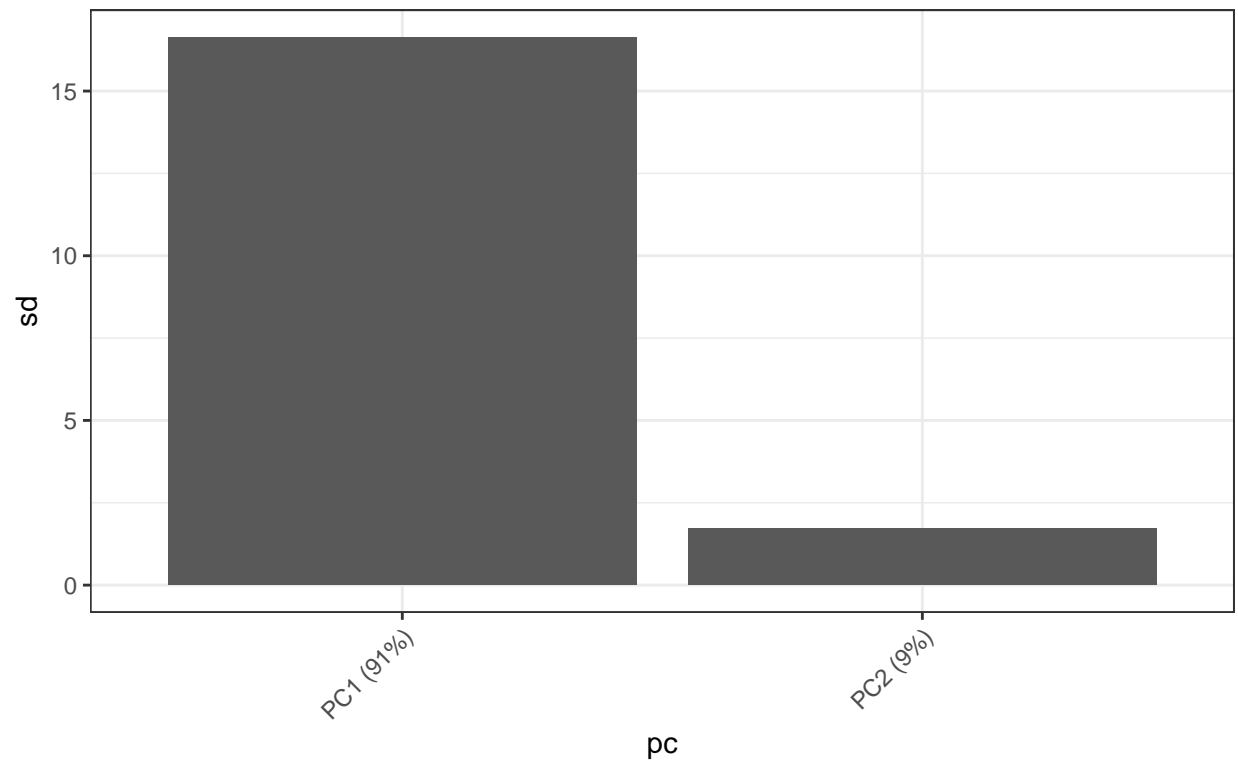
Offset = 10
Screeplot; Num. markers = 4



Screeplots, PCA by donor;type

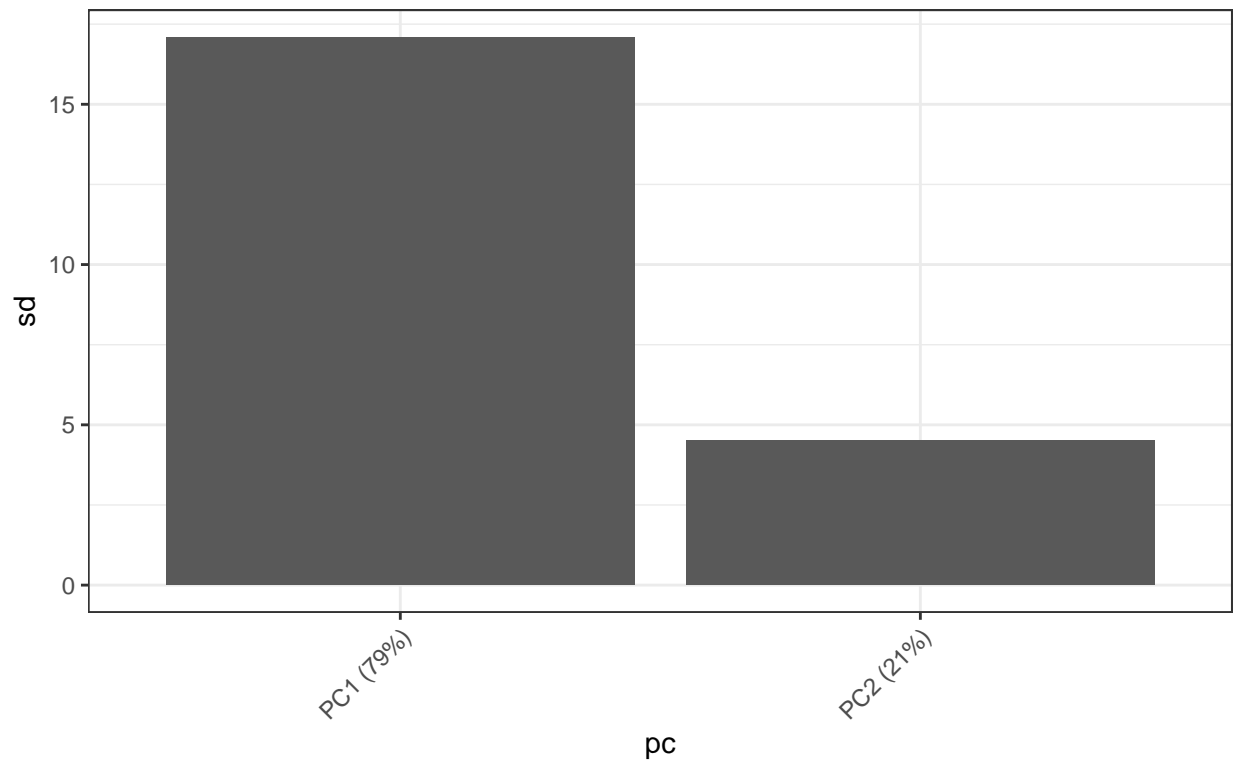
```
lexpt$`offset:2`$lpca.markers$pca.bydonortype$screeplot
```

Offset = 2
Num. markers = 2



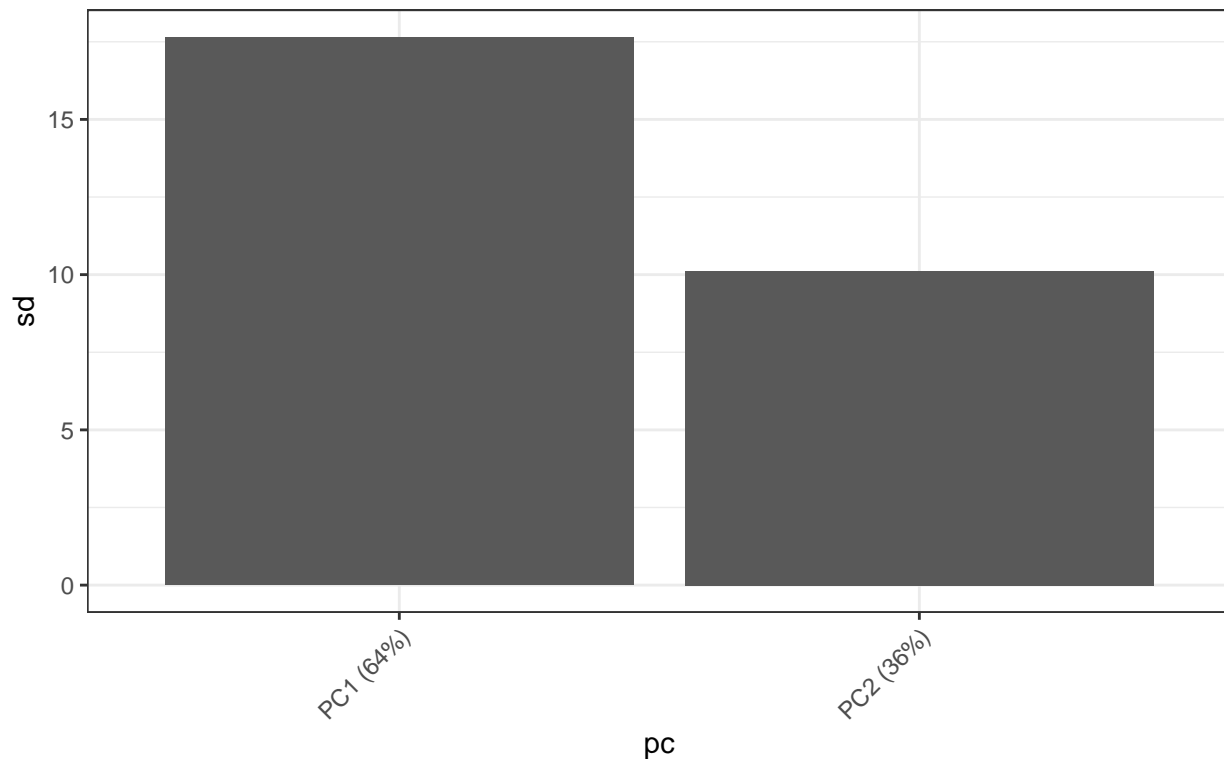
```
lexpt$`offset:5`$lpca.markers$pca.bydonortype$screeplot
```

Offset = 5
Num. markers = 2



```
lexpt$`offset:10`$lca.markers$pca.bydonortype$screepplot
```


Offset = 10
Num. markers = 2



Deconvolution results – all tests

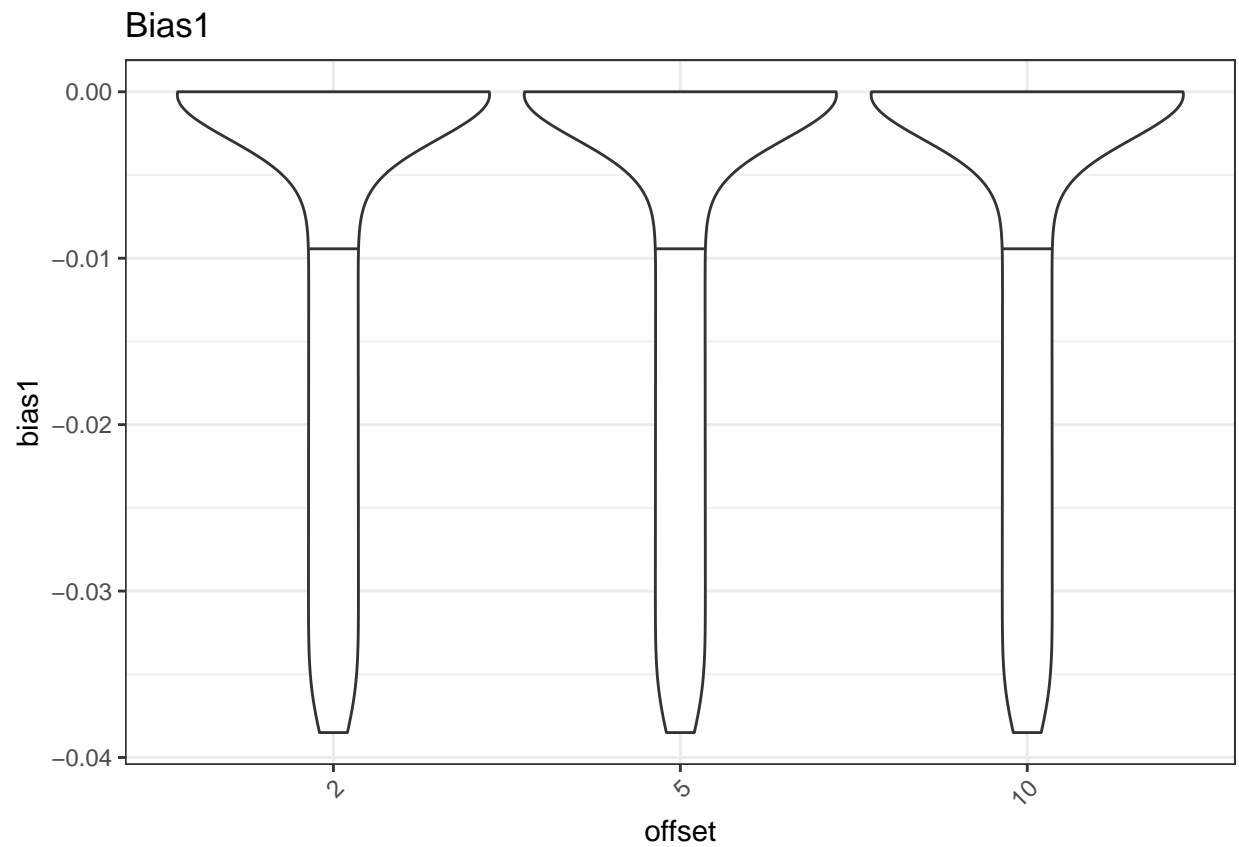
Results table

```
# get all results
dfres <- do.call(rbind, lapply(offsetv, function(offi){
  lres <- lexpt[grepl(paste0("offset:",offi,"$"), names(lexpt))][[1]]$decon.results
  dfresi <- do.call(rbind, lapply(lres, function(resi){resi$dfres}))
  dfresi <- as.data.frame(dfresi)
  dfresi$offset <- offi
  dfresi
})))
```

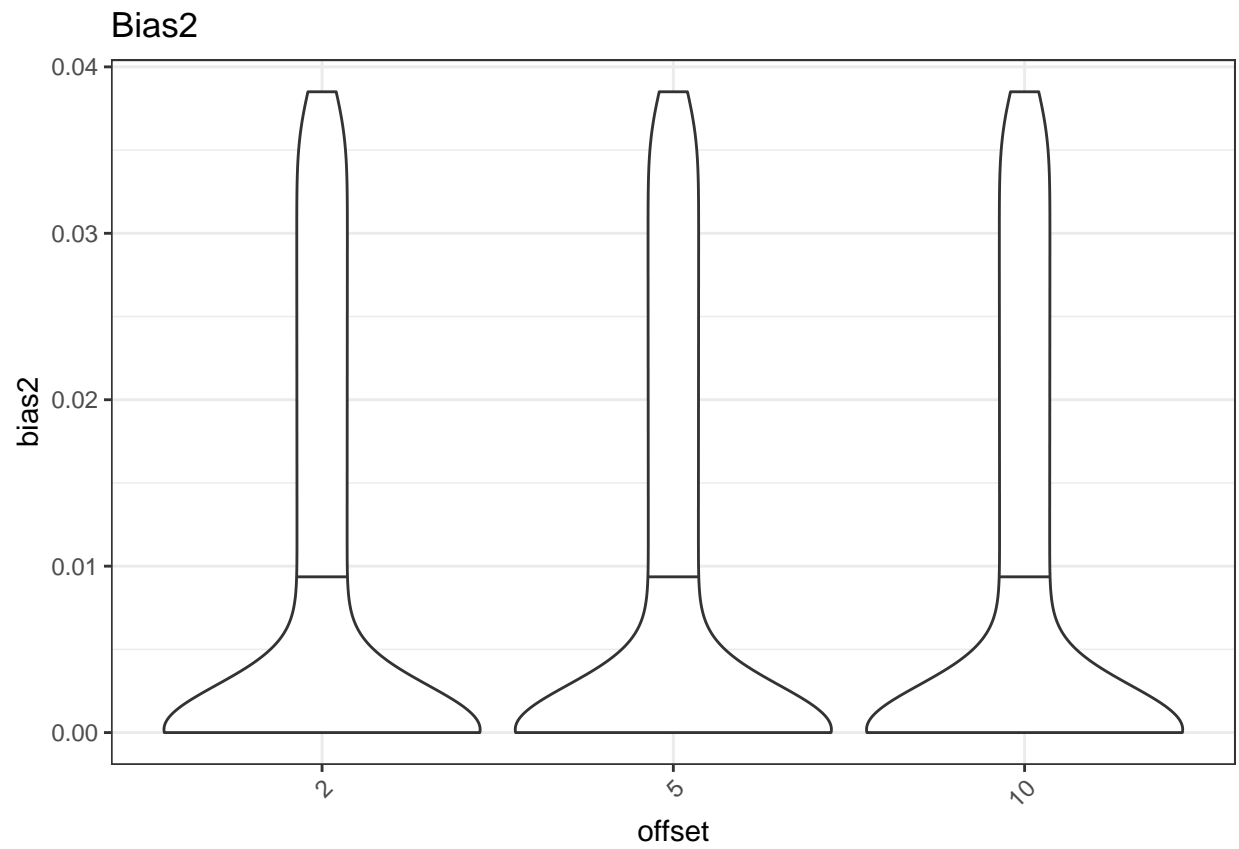
Bias violin plots

```
dfres$offset <- as.factor(dfres$offset)

ggplot(dfres, aes(x = offset, y = bias1)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias1") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

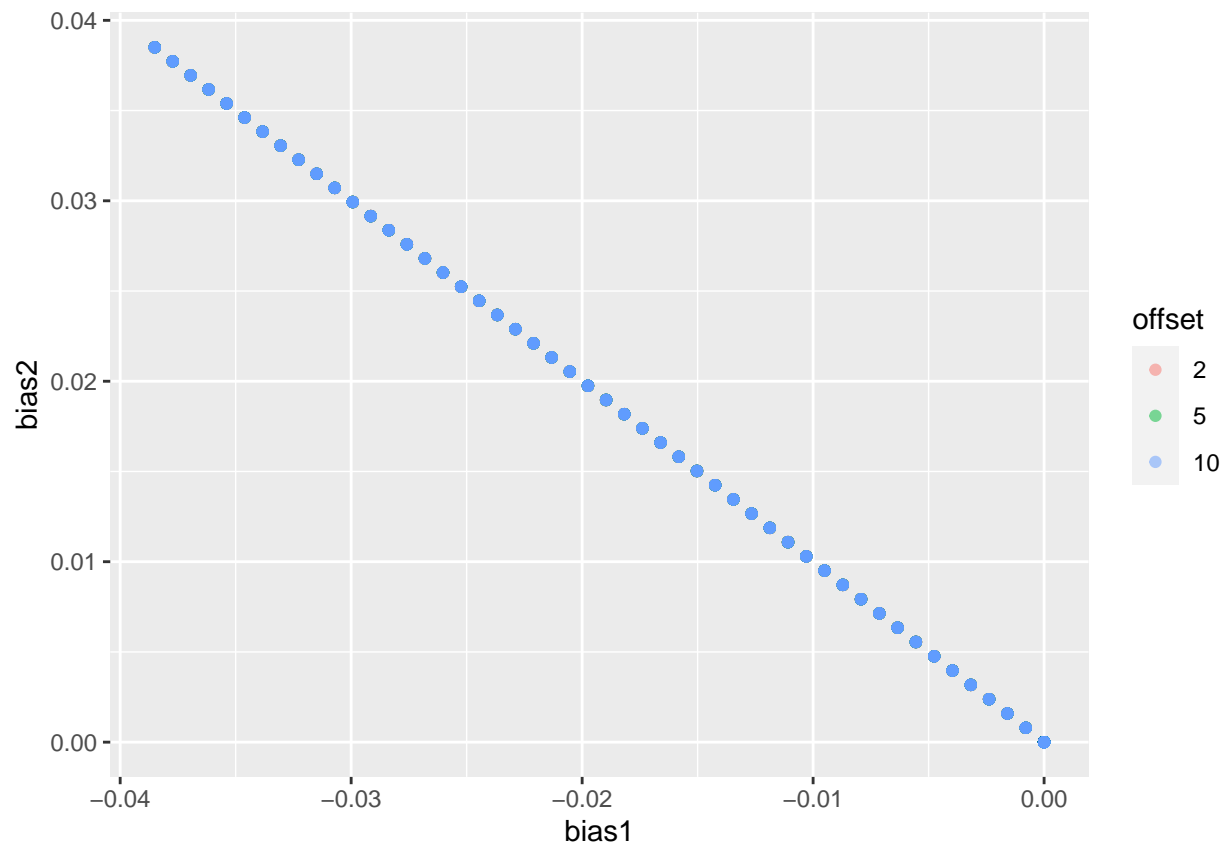


```
ggplot(dfres, aes(x = offset, y = bias2)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias2") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

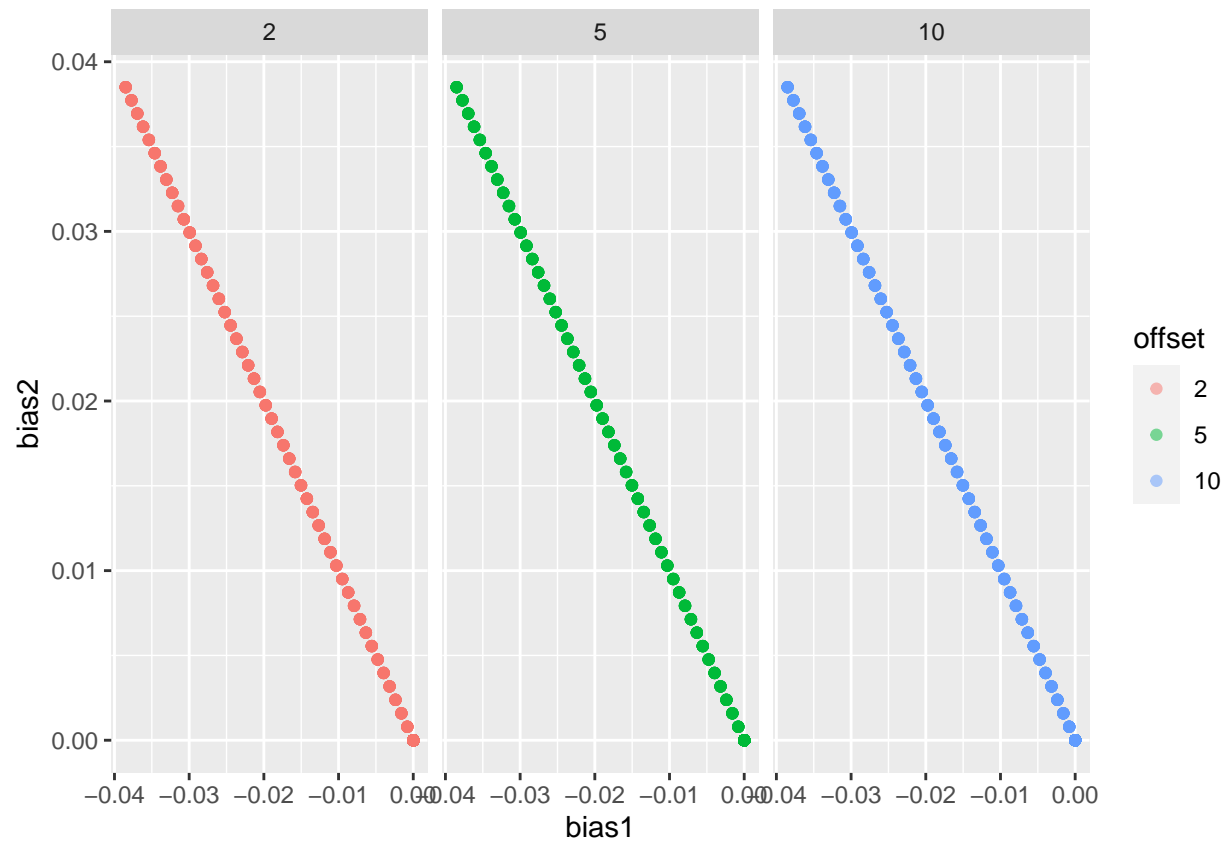


Bias scatterplots

```
ggpt <- ggplot(dfres, aes(x = bias1, y = bias2, color = offset)) + geom_point(alpha = 0.5)  
ggpt
```

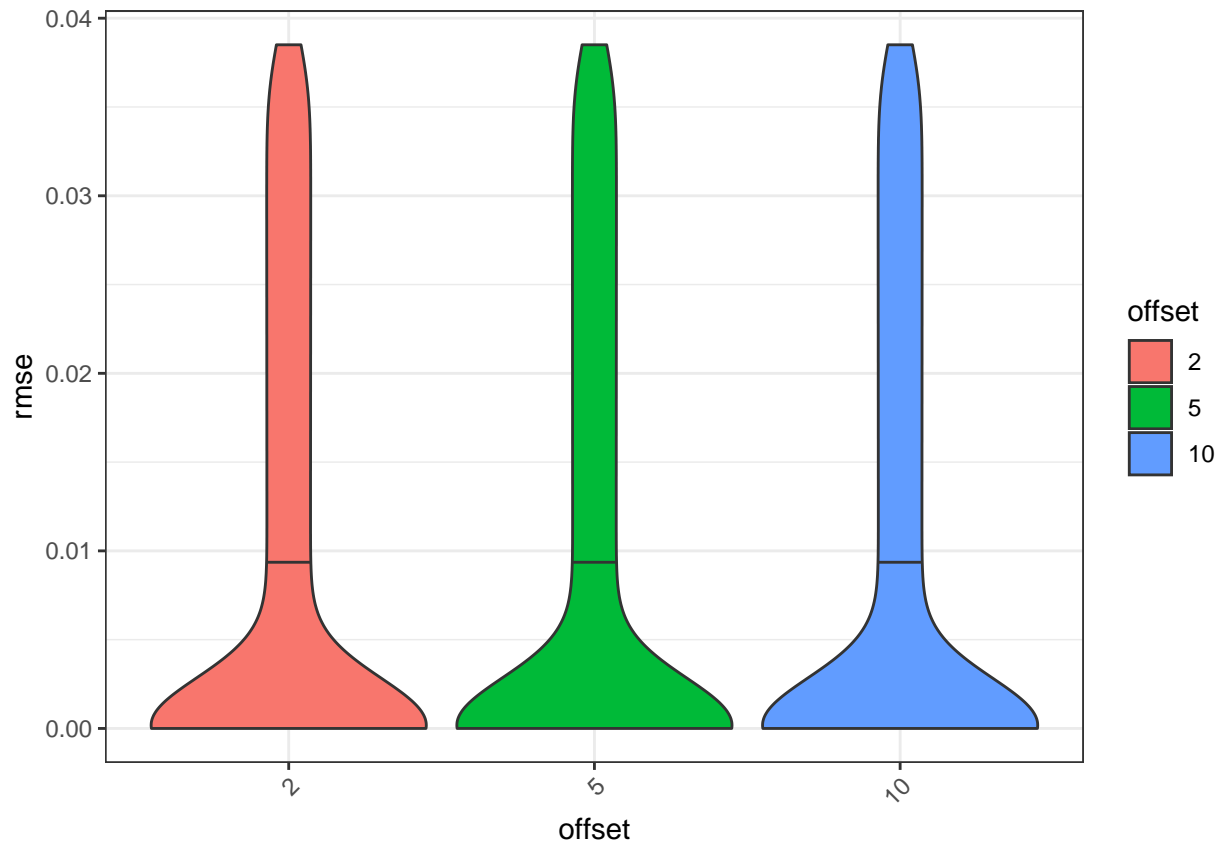


```
ggpt + facet_wrap(~offset)
```



RMSE violin plot

```
ggplot(dfres, aes(x = offset, y = rmse, fill = offset)) +
  geom_violin(draw_quantiles = 0.5) + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Deconvolution results – without S transformations

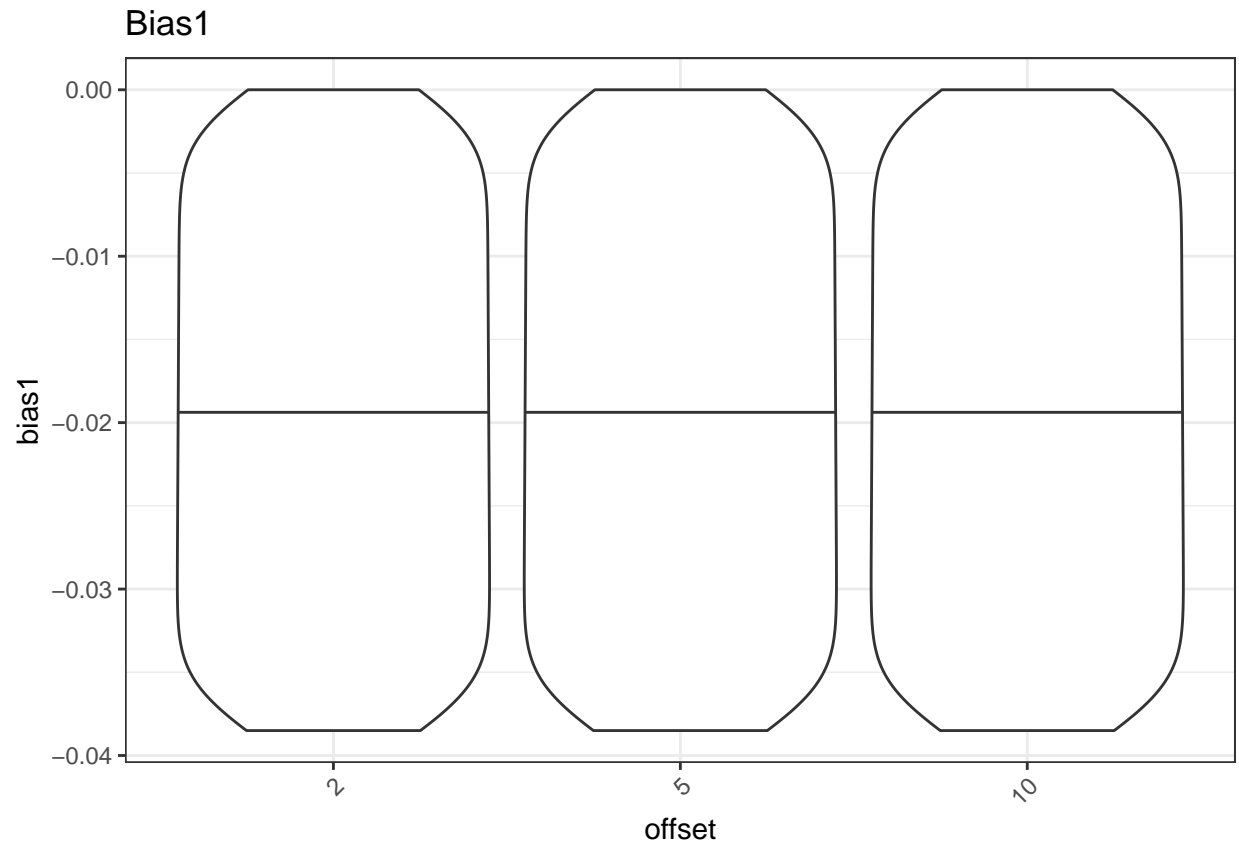
Results table

```
dfres.all <- dfres
dfres <- dfres.all[dfres.all$zs_transform==FALSE,]
```

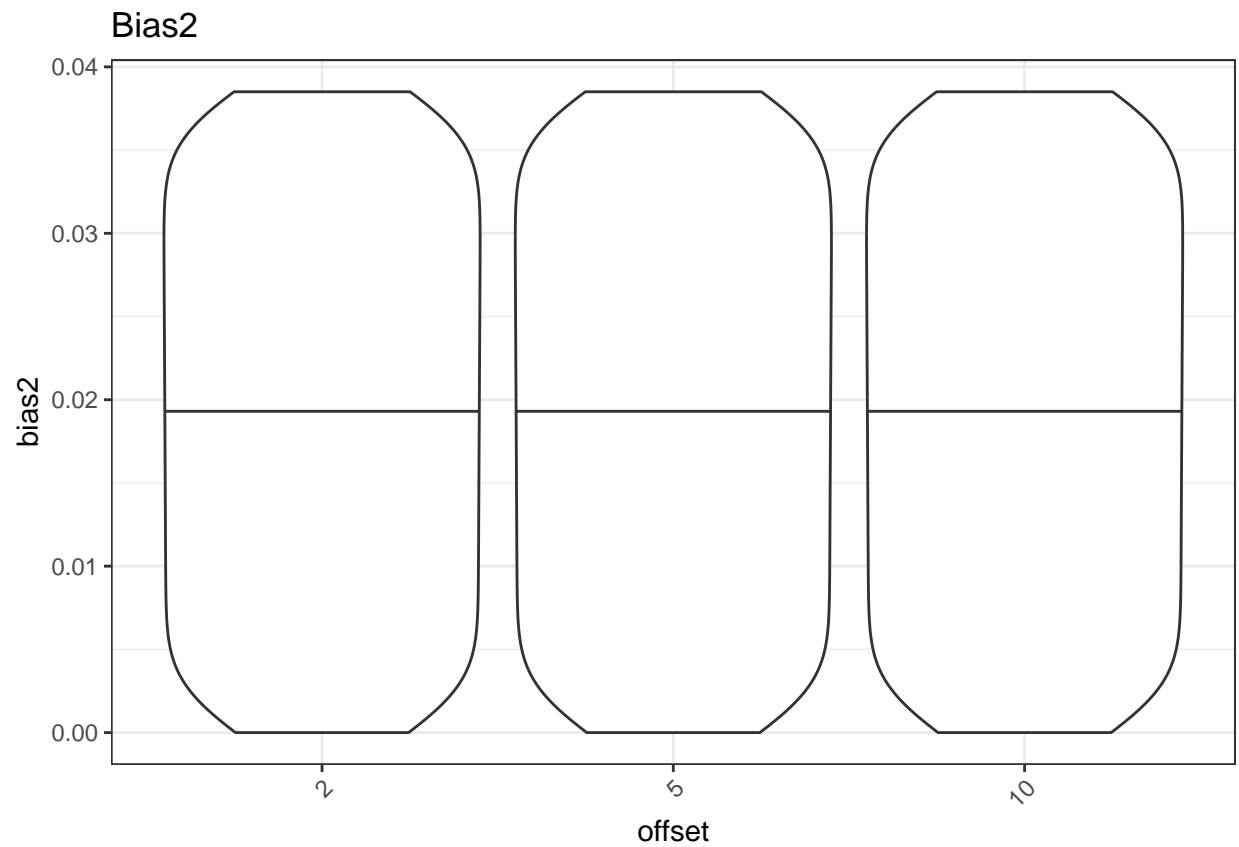
Bias violin plots

```
dfres$offset <- as.factor(dfres$offset)

ggplot(dfres, aes(x = offset, y = bias1)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias1") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

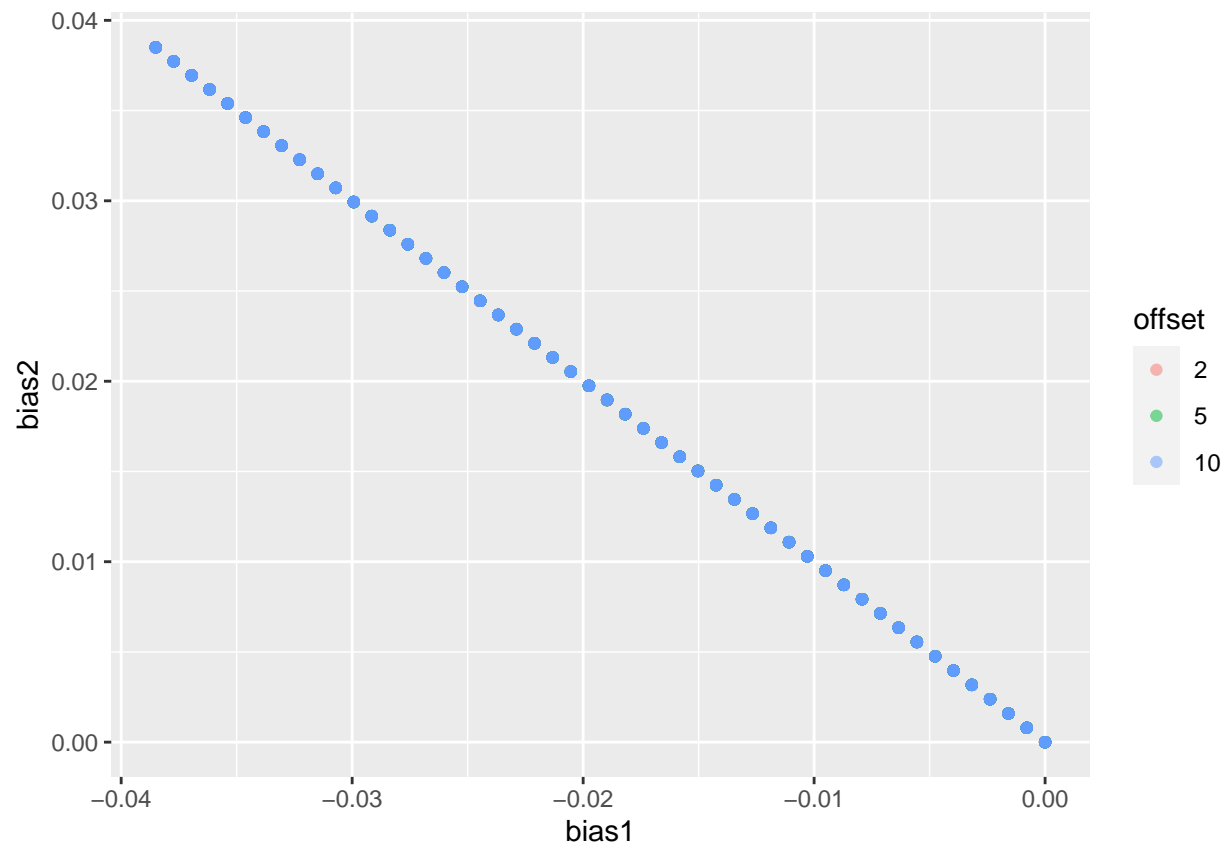


```
ggplot(dfres, aes(x = offset, y = bias2)) + geom_violin(draw_quantiles = 0.5) +  
  theme_bw() + ggtitle("Bias2") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

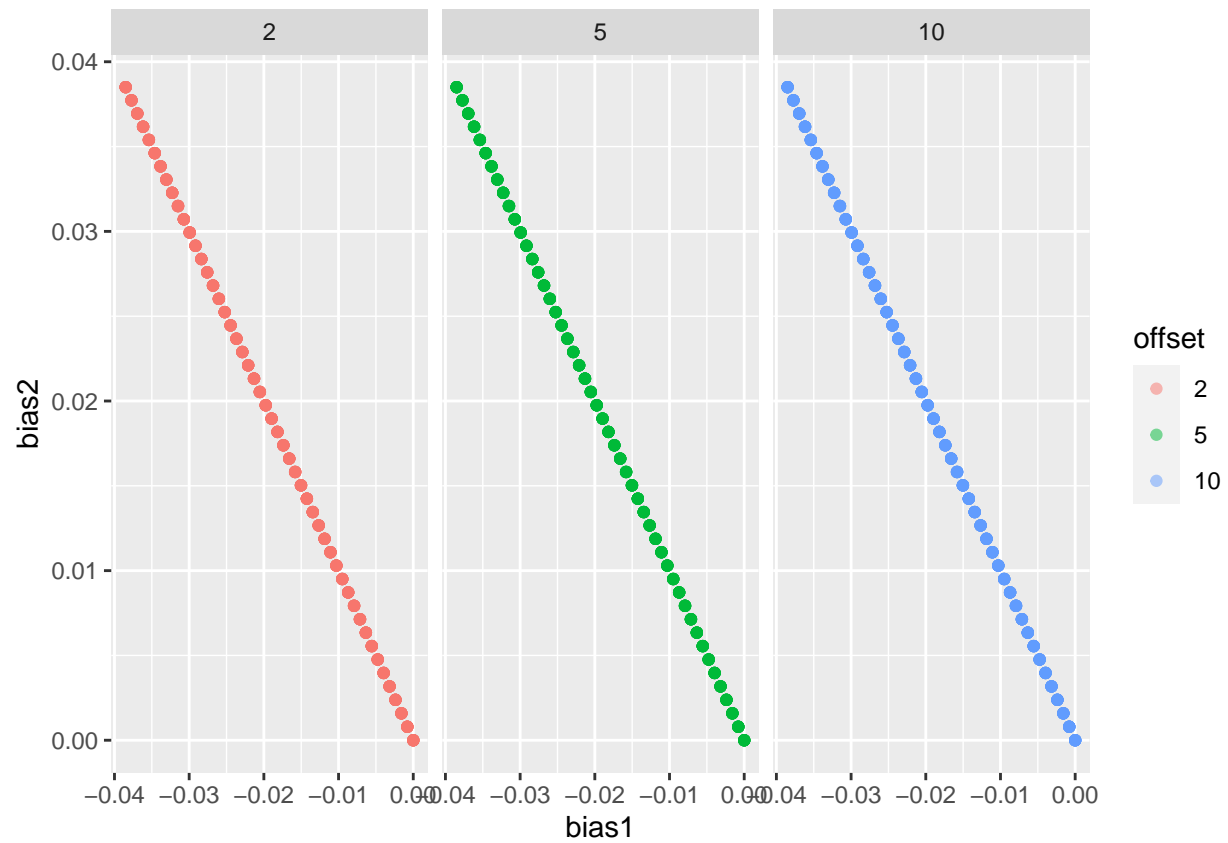


Bias scatterplots

```
ggpt <- ggplot(dfres, aes(x = bias1, y = bias2, color = offset)) + geom_point(alpha = 0.5)  
ggpt
```

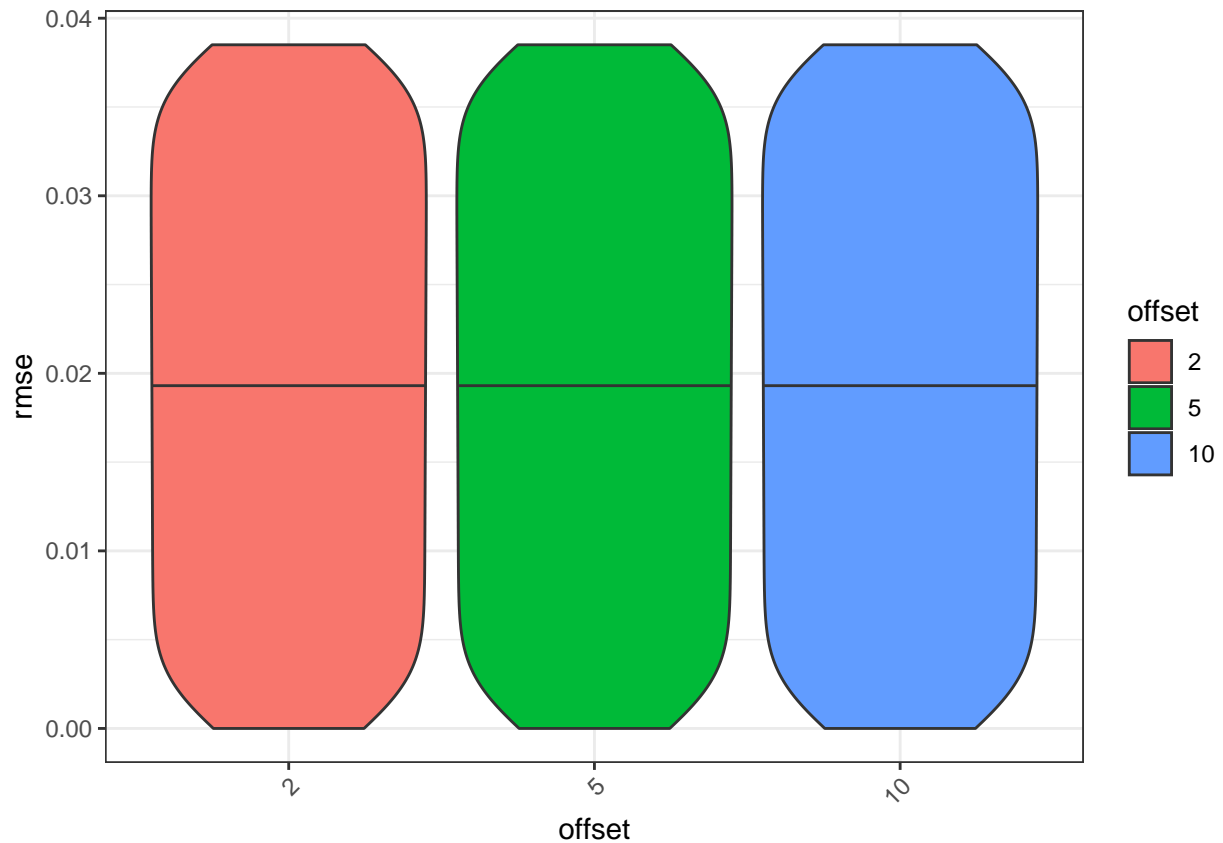



```
ggpt + facet_wrap(~offset)
```



RMSE violin plot

```
ggplot(dfres, aes(x = offset, y = rmse, fill = offset)) +
  geom_violin(draw_quantiles = 0.5) + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



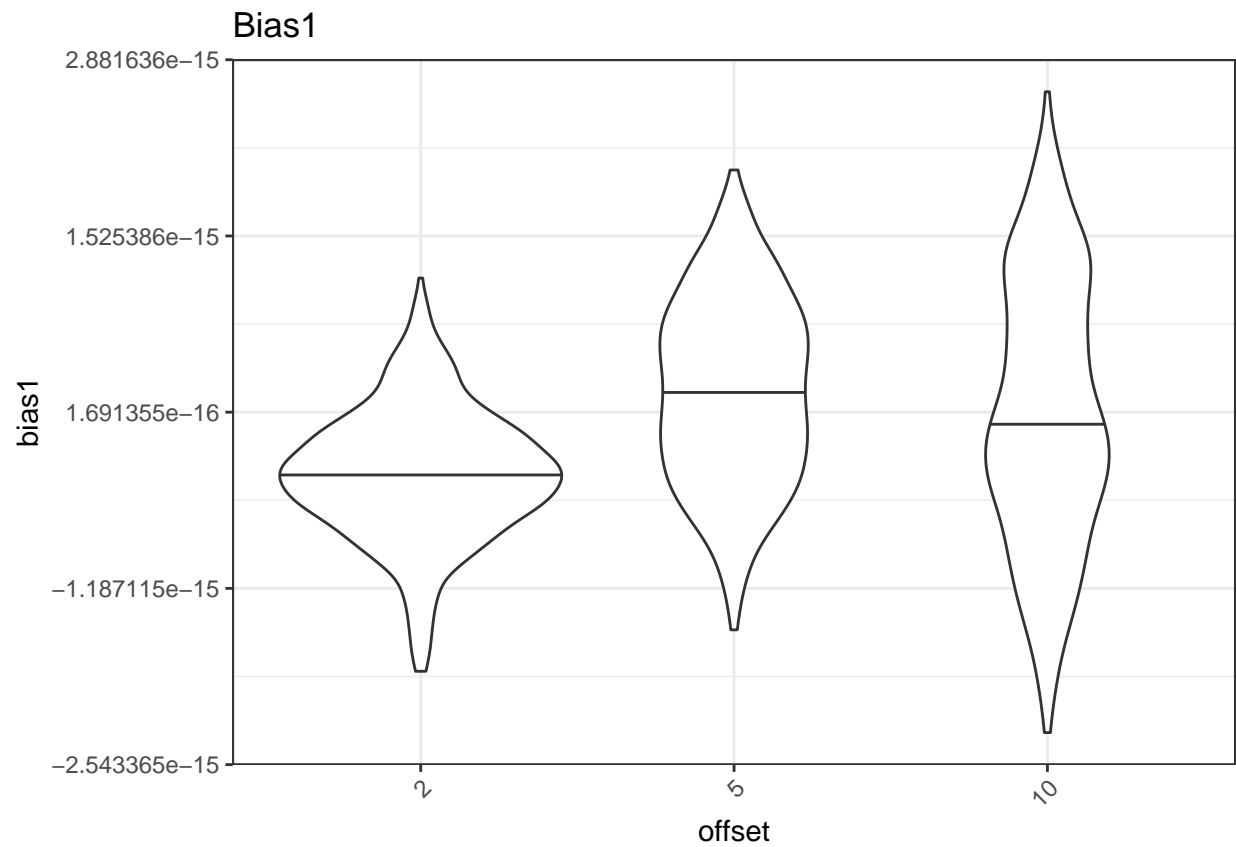
Deconvolution results – with S transformations

```
dfres <- dfres.all[dfres.all$zs_transform==TRUE,]
```

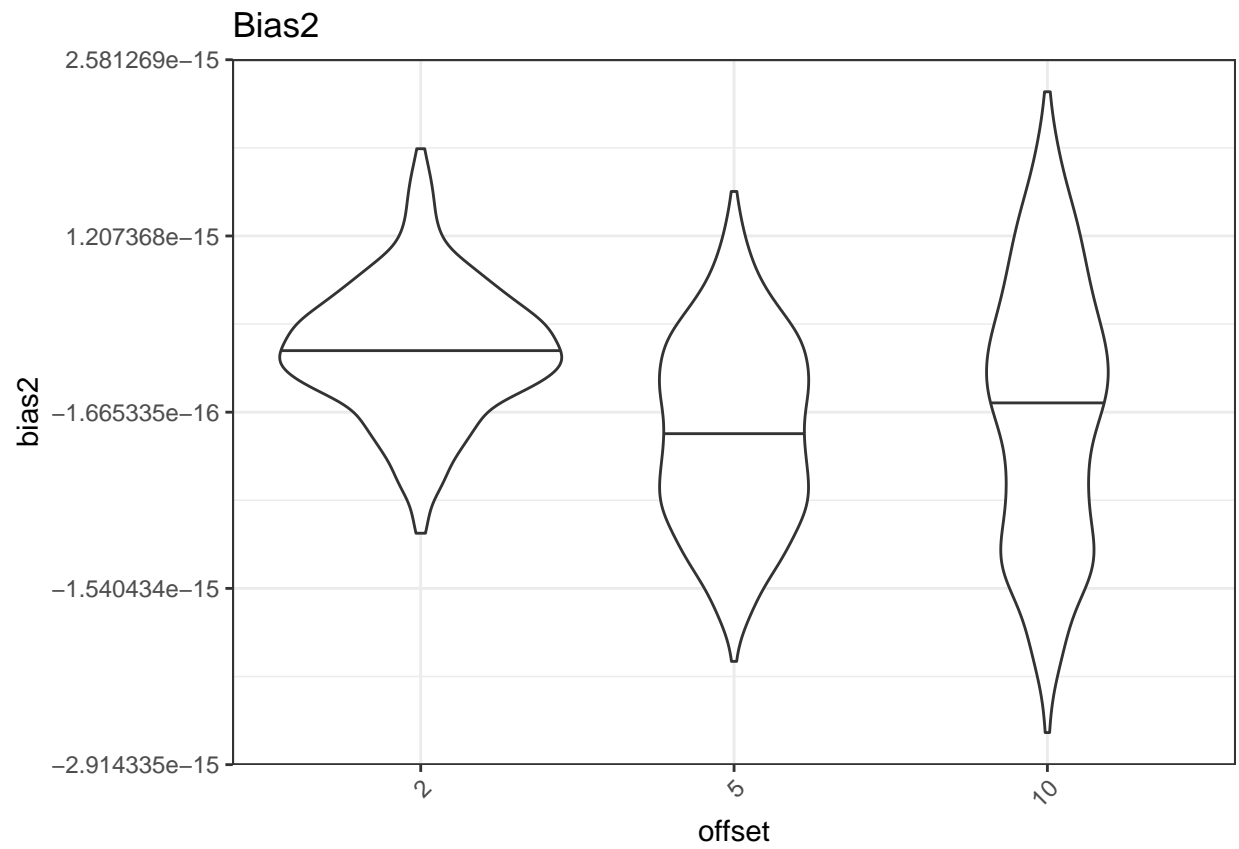
Bias violin plots

```
dfres$offset <- as.factor(dfres$offset)

ggplot(dfres, aes(x = offset, y = bias1)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias1") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

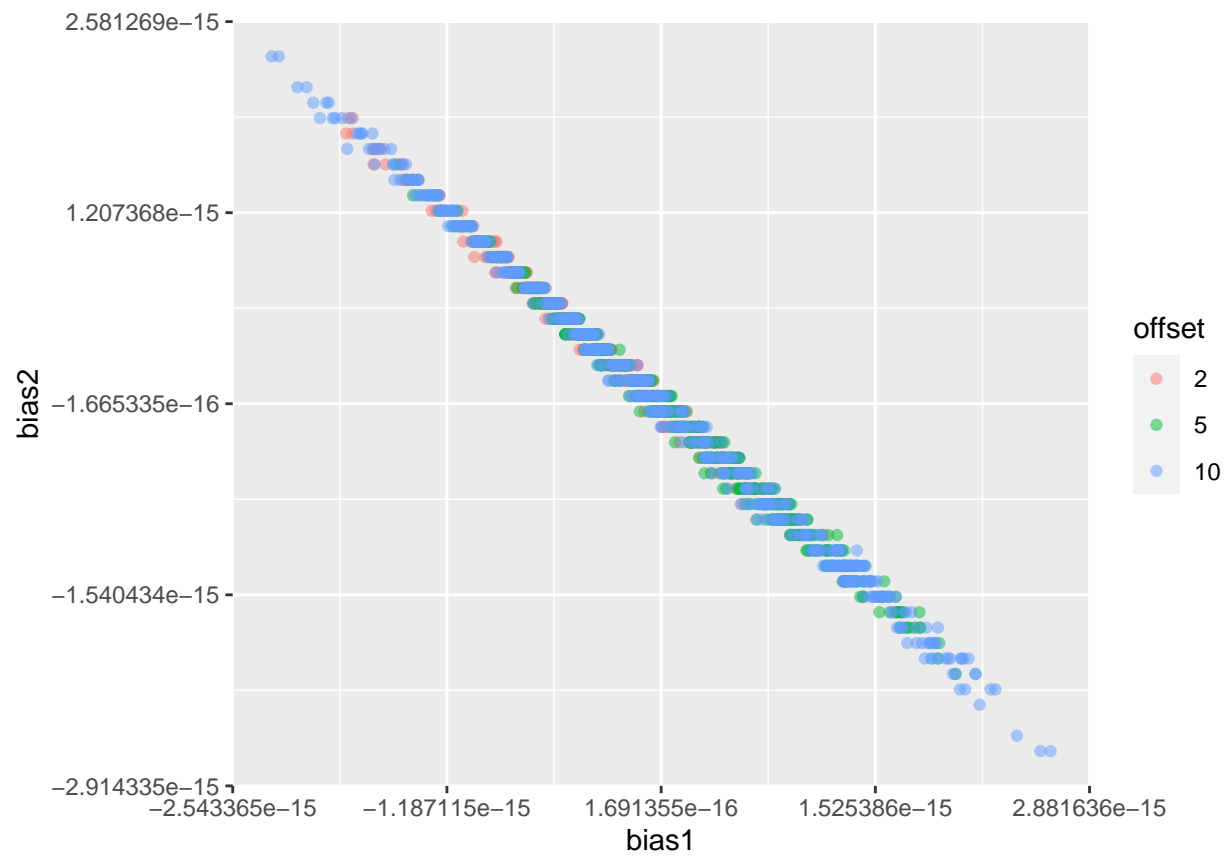


```
ggplot(dfres, aes(x = offset, y = bias2)) + geom_violin(draw_quantiles = 0.5) +
  theme_bw() + ggtitle("Bias2") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

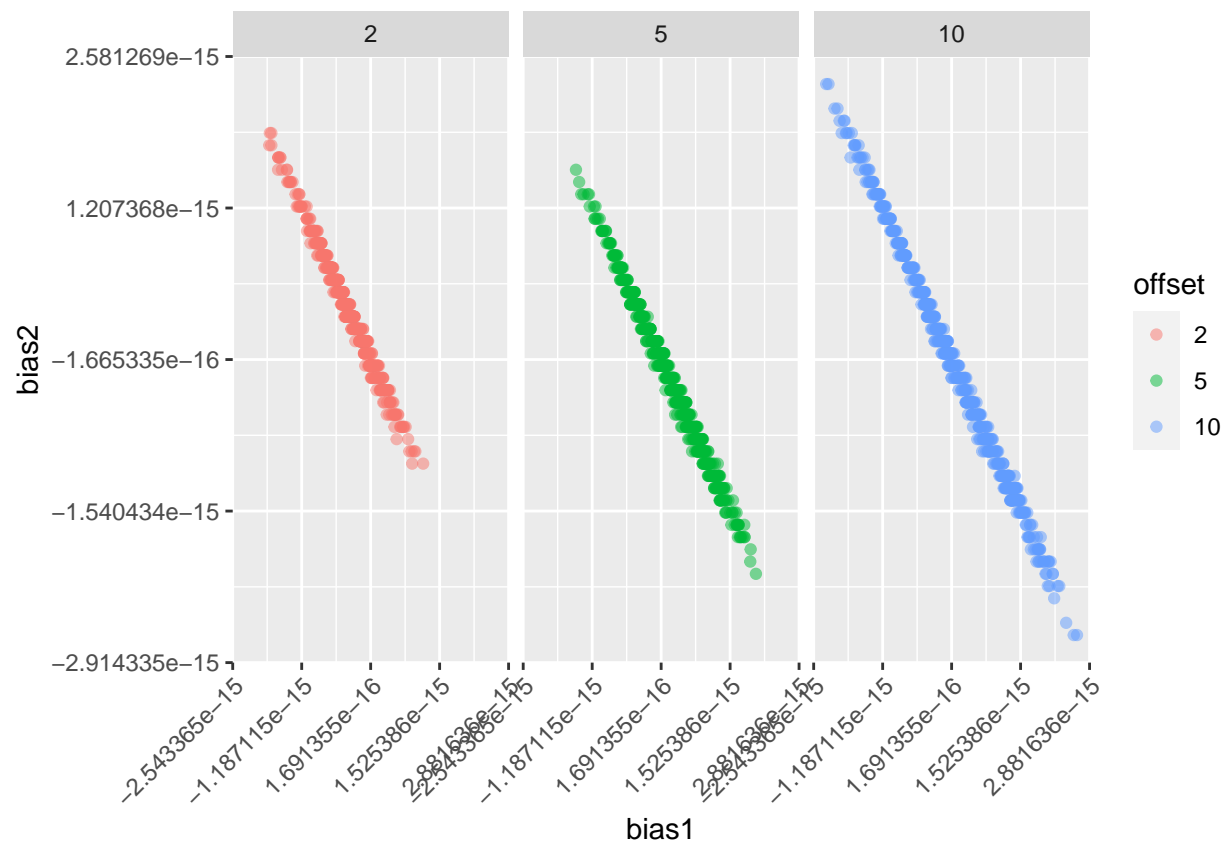


Bias scatterplots

```
ggpt <- ggplot(dfres, aes(x = bias1, y = bias2, color = offset)) + geom_point(alpha = 0.5)
ggpt
```

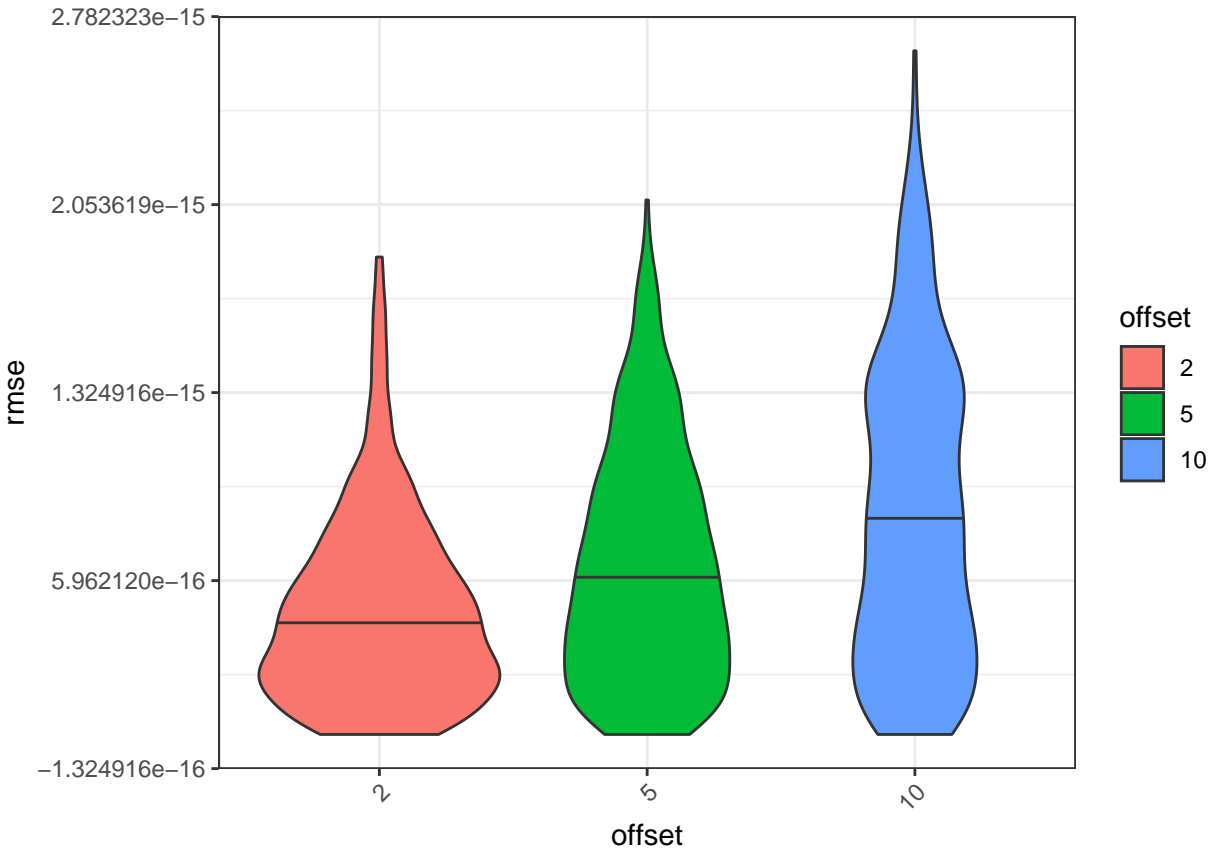


```
ggpt + facet_wrap(~offset) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



RMSE violin plot

```
ggplot(dfres, aes(x = offset, y = rmse, fill = offset)) +
  geom_violin(draw_quantiles = 0.5) + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Use variance-based weighting

We can evaluate the impact of using a variance-based weighting scheme to correct for donor-specific biases. First, set the parameters for the experiment. We initially produce a pseudobulked Y object with high donor-specific variances for 10 donors.

```
# get decon results
# note: we need to use the same pseudobulked Y data in each case.
# make the ypb from sim data with high donor variances
set.seed(0)
ndonor <- 10
gindexv <- c(1, 2)
offseti <- 1
df <- rand_donor_marker_table(ndonor = ndonor, gindexv = gindexv,
                              sd.offset.pos = offseti,
                              sd.offset.neg = offseti)
Z <- matrix(df[,ndonor+1], nrow = 2)
P <- c(0.25, 0.75)
Ypb <- t(t(P) %*% t(Z))
```

Next, we simulate a new Z signature matrix from a new randomization with the same amount of donor-specific variation. Using this and the previously generated pseudobulked Y object, we generate our unadjusted predictions.


```

# get the unadjusted predictions from new simulated donor data
set.seed(1)
ndonor <- 10
gindexv <- c(1, 2)
offseti <- 50
df.new <- rand_donor_marker_table(ndonor = ndonor, gindexv = gindexv,
                                  sd.offset.pos = offseti,
                                  sd.offset.neg = offseti)
Zunadj <- matrix(df.new[,ndonor+1], nrow = 2)
punadj <- predtype(Z = Zunadj, Y = Ypb, strict_method = "nnls",
                  proportions = TRUE, verbose = TRUE)

```

Computing proportions from outputs.

Finally we repeat predictions using the same Z signature matrix, but adding an adjustment on the between-donor variances for each gene in each type. Here, we simply divide the marker signals by the variances. We use the resulting adjusted Z signature matrix to produced our adjusted proportion predictions.

```

# get the adjusted predictions from new simulated donor data
# make adj data -- divide variances
df.adj <- df
varv <- rowVars(as.matrix(df.adj[,seq(ndonor)]))
df.adj[,ndonor+1] <- df.adj[,ndonor+1]/varv
Zadj <- matrix(df.adj[,ndonor+1], nrow = 2)
padj <- predtype(Z = Zadj, Y = Ypb, strict_method = "nnls",
                proportions = TRUE, verbose = TRUE)

```

Computing proportions from outputs.

Comparing the adjusted and unadjusted data, we see the variance adjustment produced accurate predictions whereas the unadjusted predictions are considerably inaccurate.

```

dft <- data.frame(unadj = punadj, adj = padj, true = P)
knitr::kable(dft, align = "c")

```

unadj	adj	true
0	0.2230412	0.25
1	0.7769588	0.75