What is a 0 anyway

gg

2018-04-11

To run this file: Rscript -e “rmarkdown::render(‘zero.Rmd’)”

0 is an observation, not a ground truth

We will explore what a 0 value means in a probabilistic sense

We take the dataset, which is technical replicates of the yeast dataset and compare one technical replicate with the other

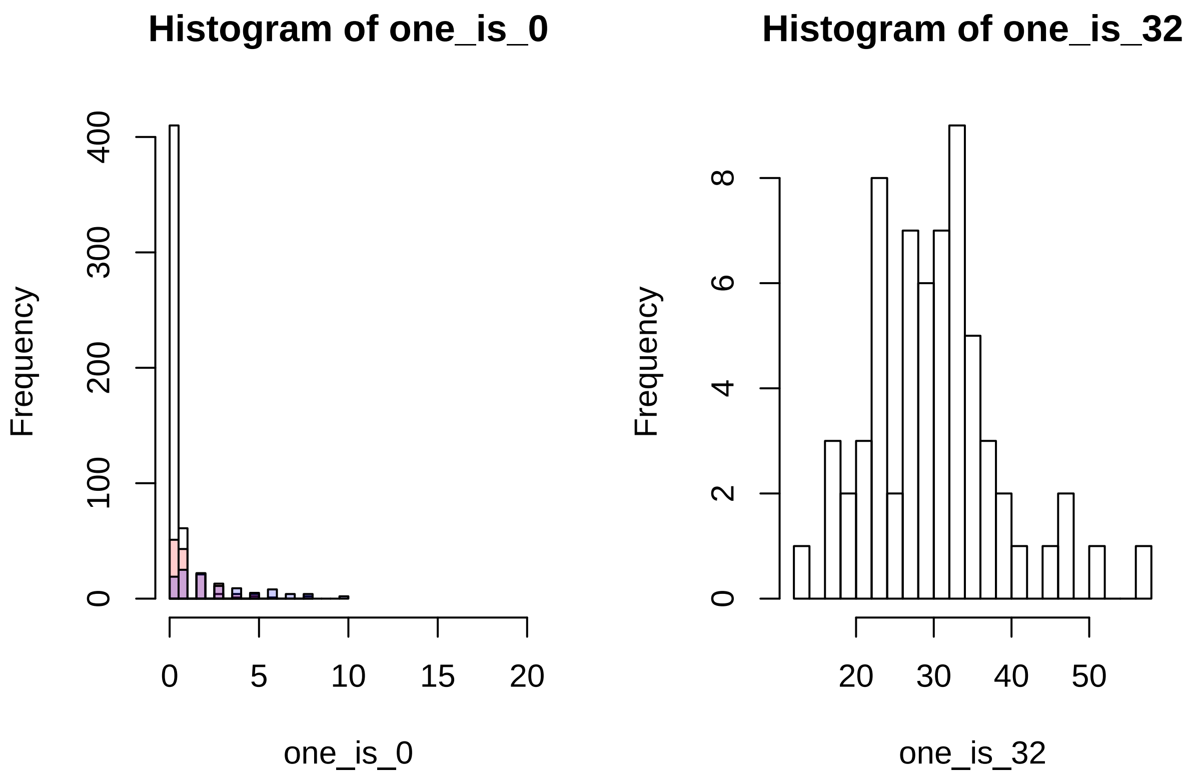
```
d <- as.matrix(read.table("data/countfinal2.tsv", header=T, row.names=1))

# remove 0 sum features
d.n0 <- d[rowSums(d) > 0,]
```

At this point we have removed all features that are 0 in all samples. These features are almost certainly not 0 if we sequence 10X more deeply, but they are of no consequence.

In the context of RNA-seq, 0 is an anomaly since most genes have a low level of stochastic expression. In the context of single cell seq, a 0 can be a 0 because expression is somewhat binary in single cells, but averaged over a population expression is continuous.

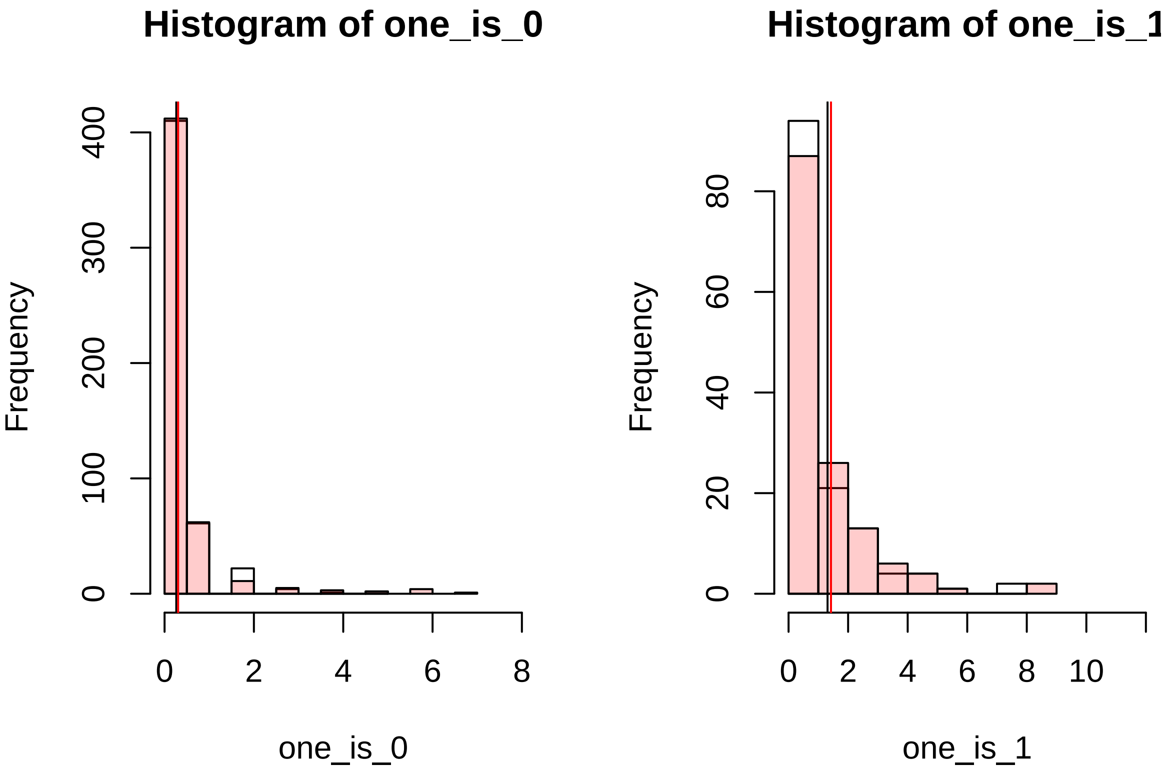
We can now examine replicates and see what a 0 means in the context of a near perfect replicate. What happens if one replicate is 0?

```
one_is_0 <- d.n0[,2][d.n0[,1] == 0]
one_is_1 <- d.n0[,2][d.n0[,1] == 1]
one_is_2 <- d.n0[,2][d.n0[,1] == 2]
one_is_32 <- d.n0[,2][d.n0[,1] == 32]

par(mfrow=c(1,2))
hist(one_is_0, breaks=9, xlim=c(0,20))
hist(one_is_1, breaks=12, add=T, col=rgb(1,0,0,0.2))
hist(one_is_2, breaks=15, add=T, col=rgb(0,0,1,0.2))
hist(one_is_32, breaks=20)
```



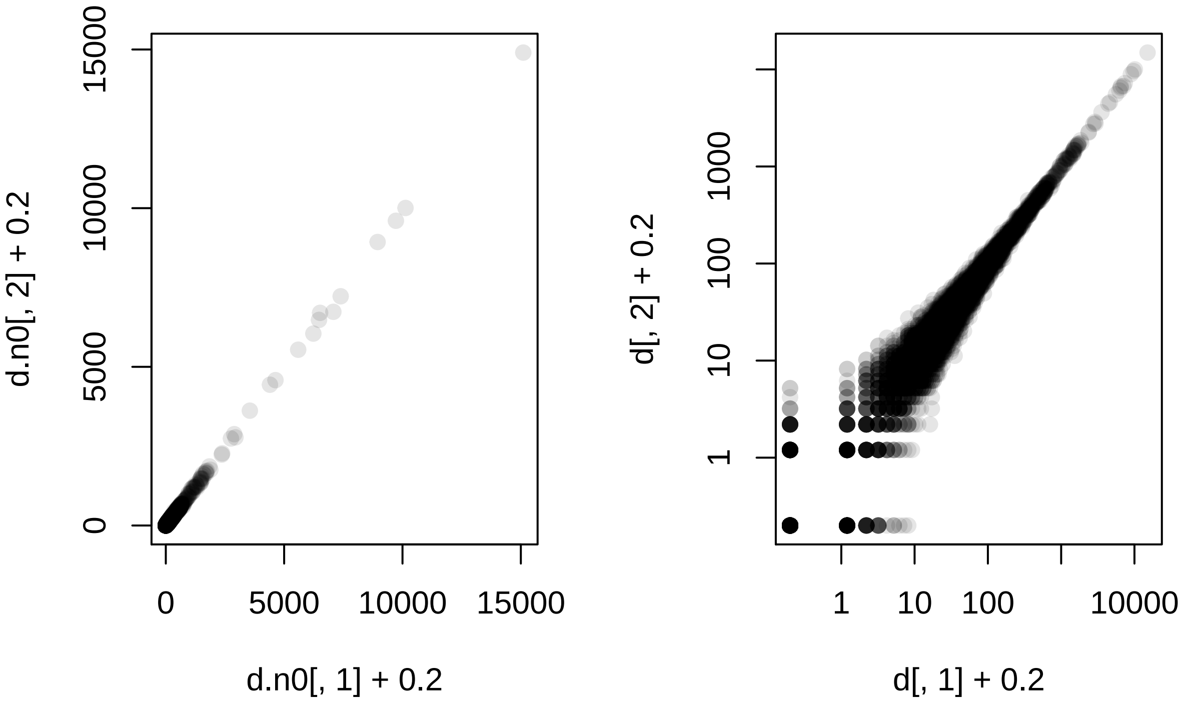
And likewise we can see if this holds for another replicate.

```
# compare replicates
one_is_0b <- d.n0[,3][d.n0[,1] == 0]
one_is_1b <- d.n0[,3][d.n0[,1] == 1]
one_is_2b <- d.n0[,3][d.n0[,1] == 2]
one_is_32b <- d.n0[,3][d.n0[,1] == 32]

par(mfrow=c(1,2))
hist(one_is_0, breaks=9, xlim=c(0,8))
hist(one_is_0b, breaks=15, add=T, col=rgb(1,0,0,0.2))
abline(v=mean(one_is_0))
abline(v=mean(one_is_0b), col="red")

hist(one_is_1, breaks=9, xlim=c(0,12))
hist(one_is_1b, breaks=9, add=T, col=rgb(1,0,0,0.2))
abline(v=mean(one_is_1))
abline(v=mean(one_is_1b), col="red")
```



We can plot all vs all

```
par(mfrow=c(1,2))
plot(d.n0[,1] + 0.2, d.n0[,2] + 0.2, pch=19, col=rgb(0,0,0,0.1))
plot(d[,1] +0.2, d[,2] +0.2, log="xy", pch=19, col=rgb(0,0,0,0.1))
```



In RNA-seq we typically do not have enough samples to have independent power for each feature.

probability of 0 is not 1 In this dataset the E(0) is ~ 0.22 but we cannot estimate it in most datasets, so add a prior

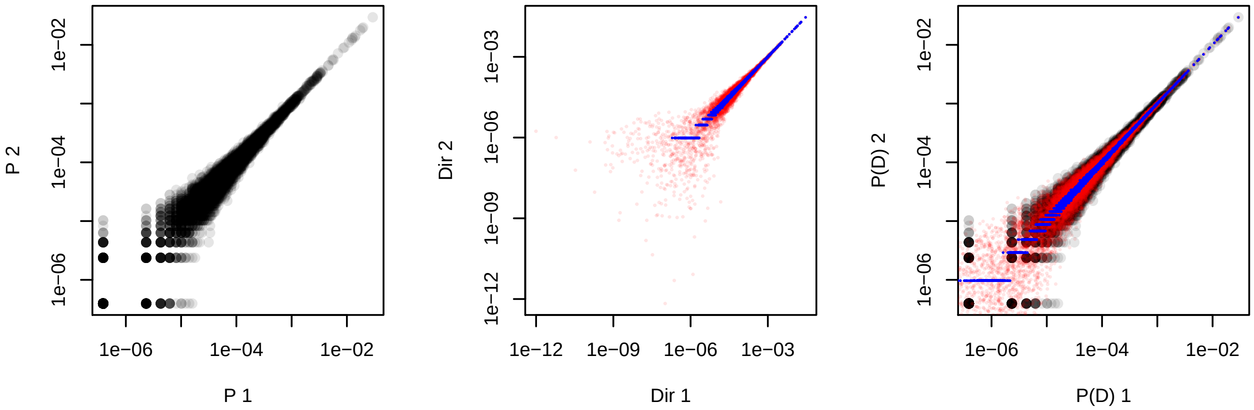
The data are discrete probabilities of observing the feature in the DNA sequencing library scaled by the read count; converted to integers by 'counting'. If we sequenced 10X deeper, then 0 values would be converted to a integers between 0 and ~10 (sampling). If we sequenced 1000X deeper, then we would get an even better estimate of the actual underlying value for our 0s. Most of the 0s at some point will be converted to a count—so what is an appropriate value for 0? Somewhere between 0 (can never occur in the observable universe) and 1 (we always should see it) is a least likely value that, over many experiments will be least likely to perturb the system? That value is 0.5. Not always the best, usually not the optimum, but in general the least wrong.

Lets generate random instances of the data. The maximum likelihood probability is the count divided by the sum of the sample. This is a discrete probability since we are dealing with finite values.

Since we know that a pure replicate is not identical and differs by sampling variation the ML estimate is strongly affected by sampling. We can convert the discrete estimate of the probability to a continuous probability by modelling as a multivariate Poisson process with a fixed sum: this is a Dirichlet distribution.

Even though the E(Dir) is very close to the ML estimate, the individual Dir instances are variable. This variation is a reasonable match for the actual underlying technical variation (tails are a bit broader in general), but is reasonable.

```
par(mfrow=c(1,3))
plot((d.n0[,1] + 0.2) /sum(d.n0[,1] + 0.2),
    (d.n0[,2] + 0.2)/sum(d.n0[,2] + 0.2),
    log="xy", xlab="P 1", ylab="P 2",
    pch=19, col=rgb(0,0,0,0.1))


d.dir <- rdirichlet(16, d.n0[,1]+0.5)
ml <- (d.n0[,1]+0.5) / sum(d.n0[,1]+0.5)
E_dir <- apply(d.dir, 2, mean)
plot(d.dir[1,], d.dir[2,], pch=19, cex=0.2,xlab="Dir 1",
    ylab="Dir 2", col=rgb(1,0,0,0.1), log="xy")
points(E_dir, ml, pch=19,cex=0.1, col="blue")

plot((d.n0[,1] + 0.2) /sum(d.n0[,1] + 0.2),
    (d.n0[,2] + 0.2)/sum(d.n0[,2] + 0.2),
    log="xy", pch=19, col=rgb(0,0,0,0.1),
    xlab="P(D) 1", ylab="P(D) 2")
points(d.dir[1,], d.dir[2,], pch=19, cex=0.2, col=rgb(1,0,0,0.1))
points(d.dir[1,], d.dir[3,], pch=19, cex=0.2, col=rgb(1,0,0,0.1))
points(d.dir[1,], d.dir[4,], pch=19, cex=0.2, col=rgb(1,0,0,0.1))
points(E_dir, ml, pch=19,cex=0.1, col="blue")
```



The expected value of the Dir instances is close to the