Notes on Weights

2025-05-27

When generating the gold standard networks, we must consider how to weight the edges.

First, generate a scale-free graph using the `sample_pa` function from `igraph`.

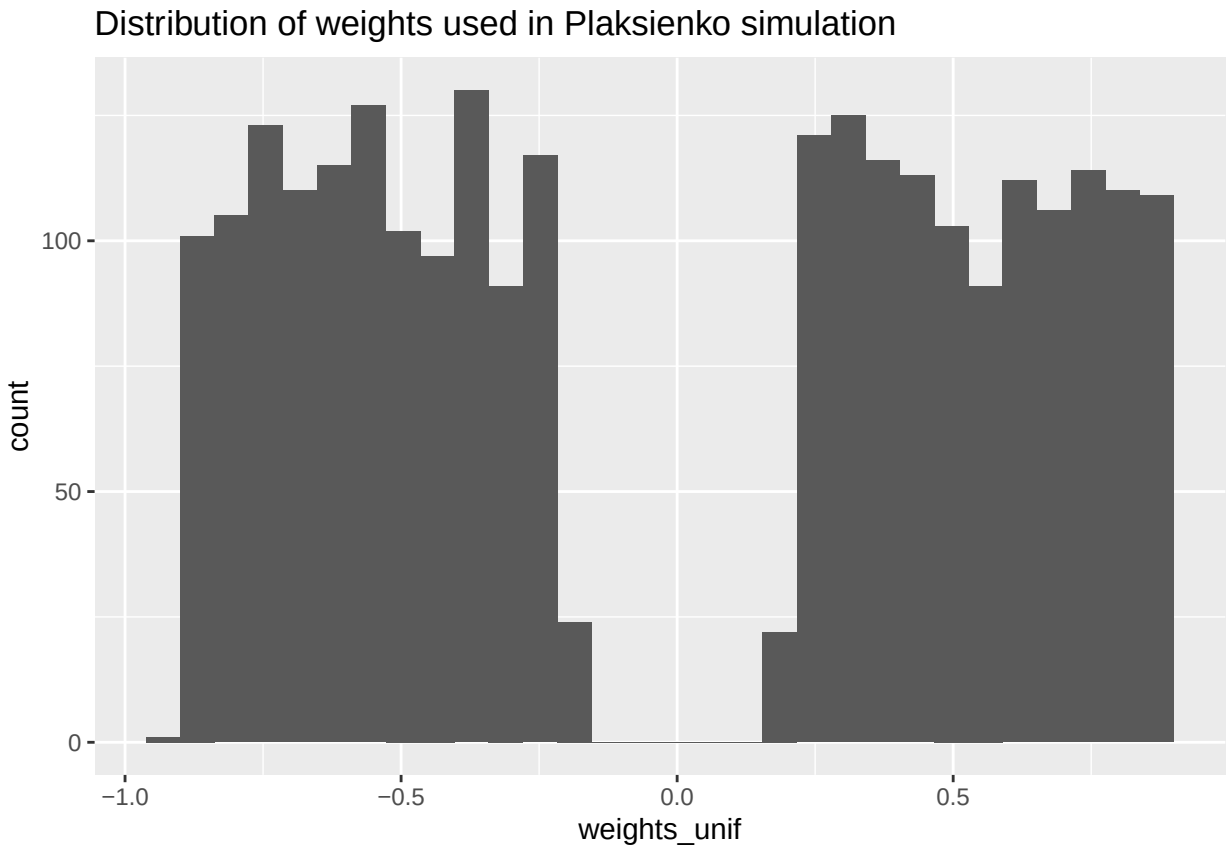
```
p = 500
g1 = sample_pa(p, power = 1, m = 5, directed = FALSE)
```

One simple option is to sample from a uniform distribution, say between 0.2 to 0.9 and assign those as weights to non-zero entries to the simulated adjacency matrix. This strategy was used by Plaksienko et. al in their recent simulation paper.

```
G <- as.matrix(as_adjacency_matrix(g1))
size <- sum(G) / 2
a <- 0.2
b <-  0.9
samp_right <- runif(ceiling(size / 2), min = a, max = b)
samp_left <- runif(ceiling(size / 2), min = -b, max = -a)
weights_unif <- sample(c(samp_left, samp_right), size)

ggplot(data.frame(weights_unif = weights_unif), aes(x = weights_unif)) +
  geom_histogram() +
  labs(title = "Distribution of weights used in Plaksienko simulation")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In an effort to simulate data that better reflects real-world data one may encounter, Shutta et al generated an empirical distribution of edge weights by using real data. They used metabolomic data from CATHGEN to generate the weights and probabilities.

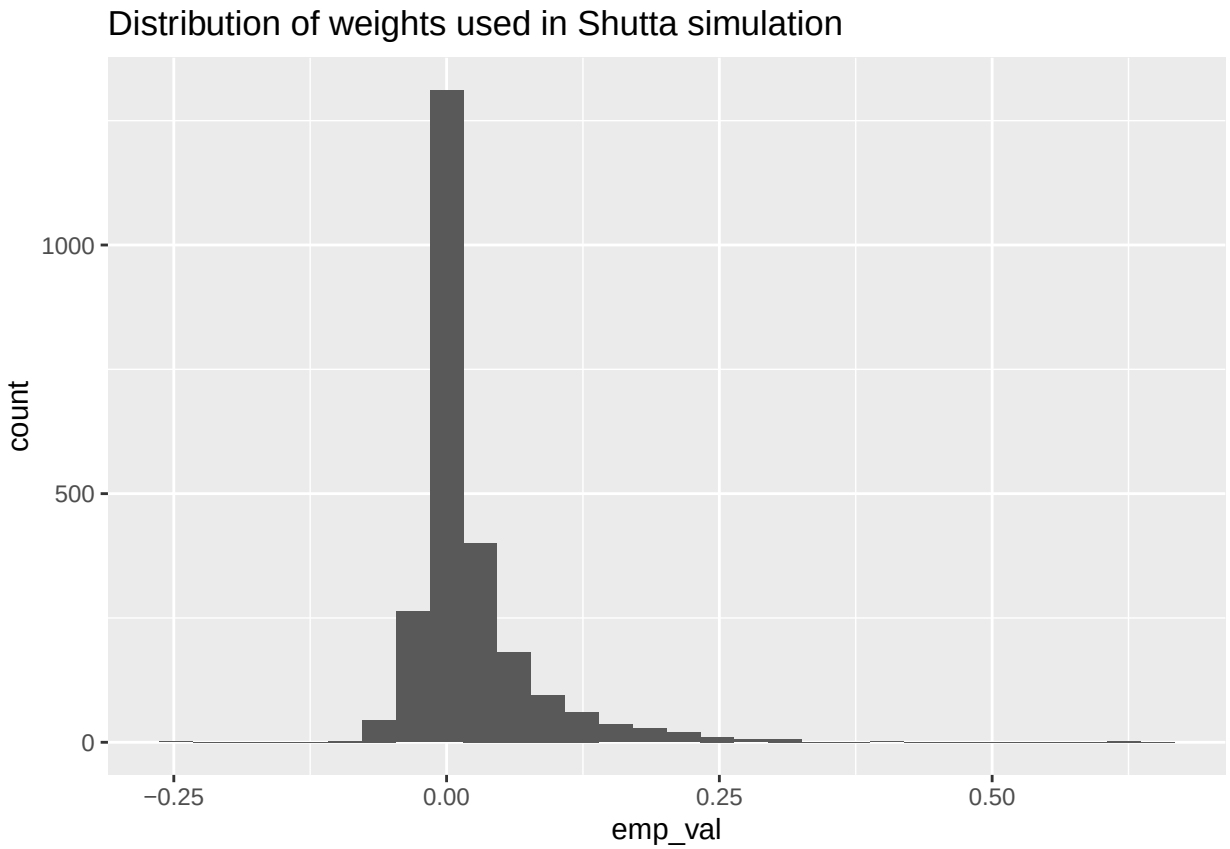
```
# load empirical distribution of weights from metabolomics
realDataHist = read.csv("mxDist.csv")
```

As we can see below, by sampling the weights using the empirical distribution, the generated weights for the same graph above are more closely centered around zero, with fewer large values. This would reflect real world data than the uniform distribution above.

```
weights = sample(realDataHist$mids, replace=T,size=length(E(g1)), prob=realDataHist$density)

ggplot(data.frame(emp_val = weights), aes(x = emp_val)) +
  geom_histogram() +
  labs(title = "Distribution of weights used in Shutta simulation")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# ggplot(realDataHist, aes(x = mids, y = density)) +
#   geom_bar(stat = "identity")
```