

RJclust

RJ CLUST

The purpose of this package is to implement the scaled RJ clust algorithm. This algorithm has not been completely finished, but is functional for TCGA datasets and datasets with no missing values. The README file has more information.

The purpose of this vignette is to walk through an example with a small dataset ## Step 1 -

First, let's look at the data. You can see that the data is originally set up as a 10600x10000 matrix.

```
data = generateSimulationData()
dim(data$X)
```

```
## [1] 1600 1000
```

Step 2 - run RJ algorithm

Let's run the RJ algorithm and look at the results. We will use the RJscale version of RJclust here, so we must indicate how many bins we want to use to divide the data for clustering. Note that the currently suggested number of cuts is the square root of the number of columns in our matrix (in this case it's a simple round number, otherwise we would have to take the floor of the answer)

```
num_cut_example = sqrt(ncol(data$X))
num_cut_example = floor(num_cut_example)
print(num_cut_example)
```

```
## [1] 31
```

```
clust = RJclust(data$X, num_cut_example)
```

We can see there are 7 clusters.

```
clust$G
```

```
## [1] 6
```

```
table(clust$classification)
```

```
##
##  1  2  3  4  5  6
## 273 225 502 145 156 299
```

Conclusion

Overall, these results are not meaningful since the data is run on a subsetted dataset. The purpose of this vignette is to show the usefulness of this algorithm. In the future, more functionality will be provided.