Reference Manual: GiniClust2

GiniClust2 is a clustering algorithm for the simultaneous detection of common and rare cell types from single-cell gene expression data. It uses a novel cluster-aware weighted consensus clustering algorithm to combine GiniClust and Fano-based k-means clustering results, by maximizing the strengths of these individual clustering methods in detecting rare and common clusters, respectively.

GiniClust2 is written in the R programming language. To apply GiniClust2 to any scRNA-seq data set, first download the "GiniClust2_download" folder. This folder contains both GiniClust2 code, under the "Rfunction" folder, and the "Main.R" R script, which includes all steps for running GiniClust2.

Follow the steps below to run GiniClust2 (also contained in the "Main.R" file). Here we use a simulated data set as an example.

Set parameters

Fixed parameters

The following parameters are generally fixed for all datasets and do not require tuning. These include parameters for data filtering, defining the high Gini gene space, and defining thresholds for differential expression.

```
minCellNum           = 3
# filtering, remove genes expressed in fewer than minCellNum cells
minGeneNum           = 2000
# filtering, remove cells expressed in fewer than minGeneNum genes
expressed_cutoff     = 1
# filtering, for raw counts
gini.bi              = 0
# fitting, default is 0, for qPCR data, set as 1
log2.expr.cutoffl    = 0
# cutoff for range of gene expression
log2.expr.cutoffh    = 20
# cutoff for range of gene expression
Gini.pvalue_cutoff   = 0.0001
# fitting, Pvalue, control how many Gini genes chosen
Norm.Gini.cutoff     = 1
# fitting, NormGini, control how many Gini genes chosen, 1 means not us
ed
span                 = 0.9
# parameter for LOESS fitting
outlier_remove       = 0.75
# parameter for LOESS fitting
GeneList             = 1
# parameter for clustering, 1 means using pvalue, 0 means using HighNor
```

```
mGini
Gamma                 = 0.9
# parameter for clustering
diff.cutoff           = 1
# MAST analysis, filter genes that don't have high log2_foldchange to r
educe gene num
lr.p_value_cutoff     = 1e-5
# MAST analysis, pvalue cutoff to identify differentially expressed gen
es
CountsForNormalized   = 100000
# if normalizing- by default not used
```

Paths

Create a new folder in the "Proj" folder. This is where all results and figures will be placed. Change the "workdir" path to match this folder's path.

```
workdir               = "/path/to/GiniClust2_download/Proj/Simulation/"
```

Change the `Rfundir` path to match the location of the "GiniClust2/Rfunction" folder.

```
Rfundir               = "/path/to/GiniClust2_download/Rfunction/"
```

Give your data set a unique name or id by renaming `exprimentID`.

```
exprimentID           = "simu"
```

Set working directory to the `workdir` path, and create folders "results" and "figures" to store GiniClust2 output for the current project.

```
setwd(workdir)
dir.create(file.path(workdir, "results"), showWarnings = FALSE)
#folder to save results
dir.create(file.path(workdir, "figures"), showWarnings = FALSE)
#folder to save figures
```

Dataset-specific parameters

Dataset-specific parameters for k-means (`k`) and DBSCAN (`MinPts`, `eps`) clustering can be determined automatically, by setting `gap_statistic`, `automatic_eps`, and `automatic_minpts` to `TRUE`. The maximum k considered under the gap statistic is set to `K.max=10`, unless otherwise specified. To set these cluster parameters manually, values can specified by the `k`, `MinPts`, and `eps` parameters, in addition to setting automatic parameters to `FALSE`.

```
MinPts                = 3
# parameter for DBSCAN
eps                   = 0.5
# parameter for DBSCAN
k                     = 3
# k for k-means step
gap_statistic         = TRUE
```

```
# whether the gap statistic should be used to determine k
K.max                 = 10
# if using the gap statistic, highest k that should be considered
automatic_eps         = TRUE
# whether to determine eps using KNN
automatic_minpts      = TRUE
# whether to determine MinPts based on the size of the data set
```

tSNE visualization parameters `perplexity` and `max_iter` can be tuned for Gini and Fano-based visualizations; otherwise, default values of 30 and 1000, respectively, will be used.

```
perplexity_G          = 30
# parameter for Gini tSNE
perplexity_F          = 30
# parameter for Fano tSNE
max_iter_G            = 1000
# parameter for Gini tSNE
max_iter_F            = 1000
# parameter for Fano tSNE
```

Load packages and functions

```
source(paste(Rfundir,"GiniClust2_packages.R",sep=""))
source(paste(Rfundir,"GiniClust2_functions.R",sep=""))
```

Load, preprocess and filter data

Upload data set and label this `data`. The data should be formatted as a matrix or data frame, with gene names as row names, and column names optional.

```
data<-read.table("data/Data_2000_1000_10_6_4_3.xls", sep="\t", head=TRU
E, row.names=1)
```

Preprocessing and filtering steps:

```
source(paste(Rfundir,"GiniClust2_preprocess.R",sep=""))
source(paste(Rfundir,"GiniClust2_filtering_RawCounts.R",sep=""))
```

Run GiniClust

```
#Gini-based clustering steps
source(paste(Rfundir,"GiniClust2_fitting.R",sep=""))
source(paste(Rfundir,"GiniClust2_Gini_clustering.R",sep=""))

table(P_G) #P_G is the Gini-based clustering result

## P_G
## db_1 db_2 db_3 db_4 db_5
## 3000   10    6    4    3
```

```
source(paste(Rfundir,"GiniClust2_Gini_tSNE.R",sep=""))
#visualization of GiniClust results using tSNE
```

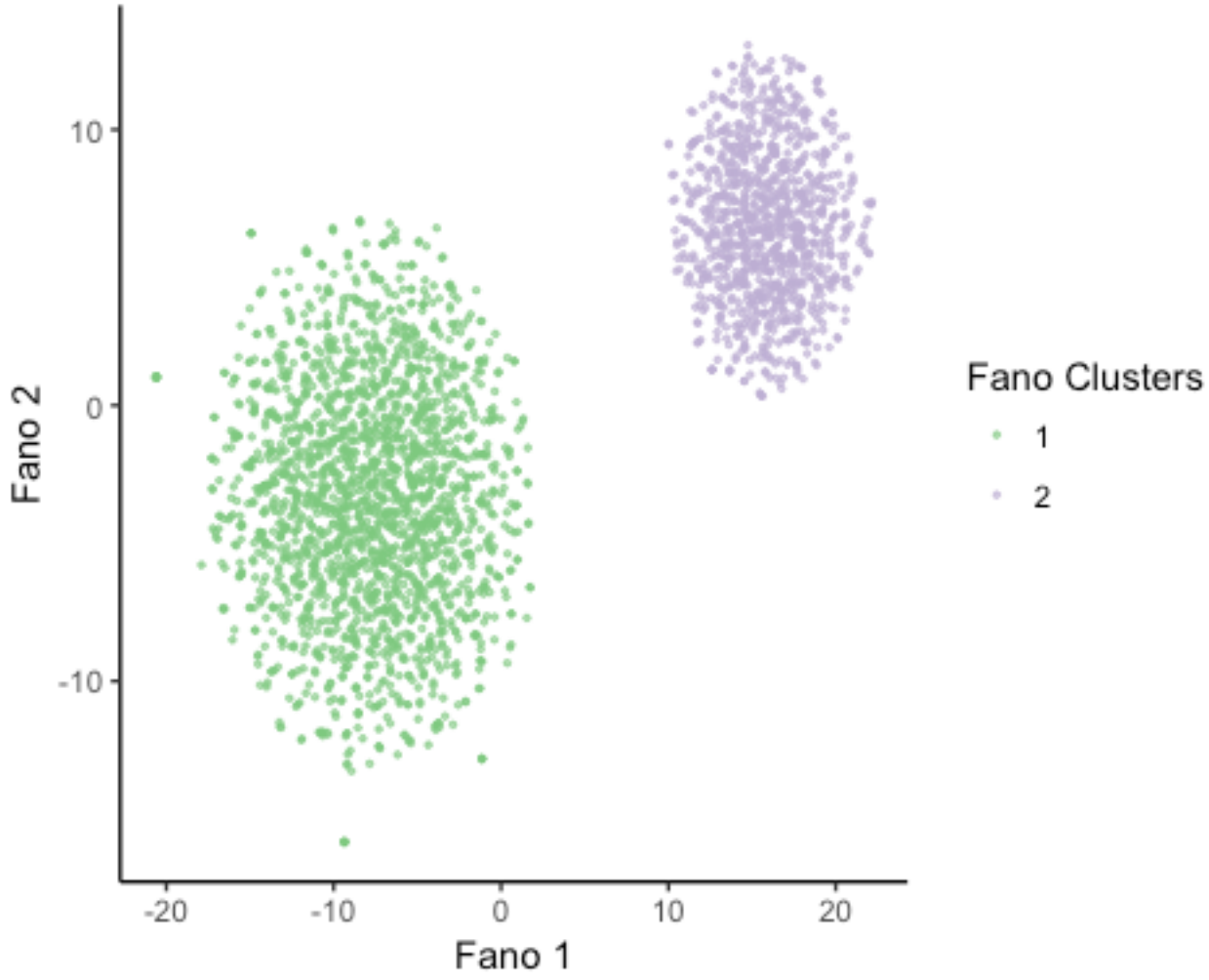
Run Fano-based k-means clustering steps

```
#Fano-based clustering steps
source(paste(Rfundir,"GiniClust2_Fano_clustering.R",sep=""))

table(P_F) #P_F is the Fano-based clustering result

## P_F
##    1    2
## 2023 1000

source(paste(Rfundir,"GiniClust2_Fano_tSNE.R",sep=""))
#visualization of k-means results using tSNE
```



Run cluster-aware weighted consensus clustering

```
#weighted consensus clustering
source(paste(Rfundir,"GiniClust2_consensus_clustering.R",sep=""))
```

```
table(finalCluster) #finalCluster is the weighted consensus clustering
result
```

```
## finalCluster
##    1    2    3    4    5    6
##    3    4 2000    6   10 1000
```

Find differentially expressed genes and visualize results

```
#final analyses
source(paste(Rfundir,"GiniClust2_DE.R",sep=""))
#find differentially expressed genes for each finalCluster
```

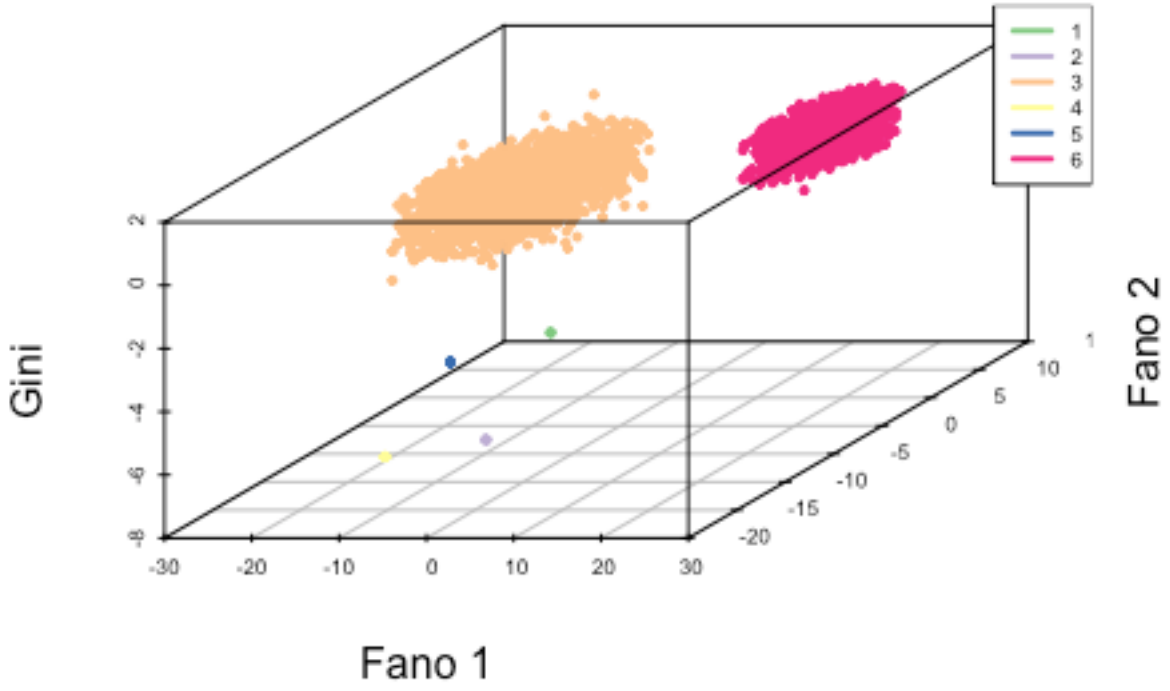
An example of top differentially expressed genes for cluster 3:

```
cluster3<-read.table("results/3_lrTest_Sig.csv",sep=",",header=T,row.na
mes=1)
head(cluster3)
```

```
##             Gene test.type p_value log2.mean.Cluster_Other log2.mean
.3
## 57  Wdr90__chr17    hurdle       0               -2.728996    4.9062
65
## 45 Srd5a2__chr17    hurdle       0               -2.961867    4.7987
76
## 13 Nkx2-1__chr12    hurdle       0               -3.052561    3.9942
63
## 26   Pcnt__chr10    hurdle       0               -1.991283    5.7464
07
## 52   Trcg1__chr9    hurdle       0               -2.208731    4.6446
06
## 25 Papss2__chr19    hurdle       0               -2.114115    4.3624
35
##    log2fold_change   Auc
## 57        7.635261 0.998
## 45        7.760643 0.998
## 13        7.046824 0.998
## 26        7.737690 0.997
## 52        6.853337 0.996
## 25        6.476550 0.996
```

A visualization of the final GiniClust2 results using a 3D tSNE plot:

```
source(paste(Rfundir,"GiniClust2_figures.R",sep=""))
#plot composite tSNE and gene-overlap venn diagrams
```


Gini
Fano 1
Fano 2
1
2
3
4
5
6
2
0
-2
-4
-6
-8
-30
-20
-10
0
10
20
30
-20
-15
-10
-5
0
5
10
1