

scGPS introduction

Quan Nguyen

2018-04-20

Contents

1. Installation instruction	1
2. A simple workflow of the scGPS:	2
2.1 Setup scGPS objects	2
2.2 Run predictions	2
2.3 Summarise results	8
3. A complete workflow of the scGPS:	17
3.1 Identify clusters in a using CORE	17
3.2 Visualise all cluster results in all iterations	18
3.3 Plot the optimal clustering result	21
3.4 Compare clustering results with other dimensional reduction methods (e.g., CIDR)	21
3.5 Find gene markers and annotate clusters	23
4. Relationship between clusters within one sample or between two samples	25
4.1 Start the scGPS prediction to find relationship between clusters	25
Start the scGPS prediction to find relationship between clusters	25
4.2 Display summary results for the prediction	29
##4.3 Plot the relationship between clusters	30
Relationship between clusters within one sample or between two samples	30

1. Installation instruction

```
# Prior to installing scGPS you need to install the SummarizedExperiment
# bioconductor package as the following
# source('https://bioconductor.org/biocLite.R') biocLite('SummarizedExperiment')

# To install scGPS from github (Depending on the configuration of the local
# computer or HPC, possible custom C++ compilation may be required - see
# installation trouble-shootings below)
devtools::install_github("IMB-Computational-Genomics-Lab/scGPS")

# for C++ compilation trouble-shooting, manual download and installation can be
# done from github

git clone https://github.com/IMB-Computational-Genomics-Lab/scGPS

# then check in scGPS/src if any of the precompiled (e.g. those with *.so and
# *.o) files exist and delete them before recompiling

# create a Makevars file in the scGPS/src with one line: PKG_LIBS =
```

```
# $(LAPACK_LIBS) $(BLAS_LIBS) $(FLIBS)

# then with the scGPS as the R working directory, manually recompile scGPS in R
# using devtools to load and install functions
devtools::document()
# update the NAMESPACE using the update_NAMESPACE.sh
sh update_NAMESPACE.sh
#for window system, to update the NAMESPACE: copy and paste the content of the file NAMESPACE_toAdd.cpp
```

2. A simple workflow of the scGPS:

The purpose of this workflow is to solve the following task: given a mixed population with known subpopulations, estimate transition scores between these subpopulation

2.1 Setup scGPS objects

```
devtools::load_all()

# load mixed population 1 (loaded from sample1 dataset, named it as day2)
day2 <- sample1

mixedpop1 <- NewscGPS_SME(ExpressionMatrix = day2$dat2_counts, GeneMetadata = day2$dat2geneInfo,
  CellMetadata = day2$dat2_clusters)

# load mixed population 2 (loaded from sample2 dataset, named it as day5)
day5 <- sample2
mixedpop2 <- NewscGPS_SME(ExpressionMatrix = day5$dat5_counts, GeneMetadata = day5$dat5geneInfo,
  CellMetadata = day5$dat5_clusters)

# load gene list (this can be any lists of user selected genes)
genes <- GeneList
genes <- genes$Merged_unique

# select a subpopulation
c_selectID <- 1
```

2.2 Run predictions

```
# run the test bootstrap

LSOLDA_dat <- bootstrap_scGPS(nboots = 2, mixedpop1 = mixedpop1,
  mixedpop2 = mixedpop2, genes = genes, c_selectID, listData = list())
#> Warning: package 'caret' was built under R version 3.4.4
#> Loading required package: lattice
#> Loading required package: ggplot2
#> Warning: model fit failed for Fold01.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 70 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 46 appears to be constant within groups
```

```

#> Warning: model fit failed for Fold03.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 5 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 54 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 1 appears to be constant within groups
#> Warning: model fit failed for Fold06.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 80 appears to be constant within groups
#> Warning: model fit failed for Fold07.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 23 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 80 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 9 54 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 46 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 1 appears to be constant within groups
#> Warning: model fit failed for Fold07.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 23 appear to be constant within groups
#> Warning: model fit failed for Fold09.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 70 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 1 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 70 appears to be constant within groups
#> Warning: model fit failed for Fold03.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 80 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 23 46 appear to be constant within groups
#> Warning: model fit failed for Fold06.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 7 appears to be constant within groups
#> Warning: model fit failed for Fold07.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 54 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 5 appears to be constant within groups
#> Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
#> trainInfo, : There were missing values in resampled performance measures.
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -2.563e-15 3.002e-01
#> [2,]  1  4.431e-02 2.736e-01
#> [3,]  1  8.184e-02 2.493e-01
#> [4,]  2  1.180e-01 2.271e-01
#> [5,]  2  1.558e-01 2.069e-01
#> [6,]  2  1.887e-01 1.886e-01
#> [7,]  2  2.178e-01 1.718e-01
#> [8,]  2  2.437e-01 1.565e-01
#> [9,]  2  2.669e-01 1.426e-01
#> [10,] 2  2.878e-01 1.300e-01
#> [11,] 4  3.091e-01 1.184e-01

```

```

#> [12,] 5 3.304e-01 1.079e-01
#> [13,] 6 3.511e-01 9.831e-02
#> [14,] 6 3.708e-01 8.958e-02
#> [15,] 8 3.898e-01 8.162e-02
#> [16,] 9 4.084e-01 7.437e-02
#> [17,] 9 4.259e-01 6.776e-02
#> [18,] 11 4.420e-01 6.174e-02
#> [19,] 11 4.586e-01 5.626e-02
#> [20,] 13 4.757e-01 5.126e-02
#> [21,] 14 4.922e-01 4.671e-02
#> [22,] 15 5.092e-01 4.256e-02
#> [23,] 15 5.245e-01 3.878e-02
#> [24,] 18 5.399e-01 3.533e-02
#> [25,] 19 5.577e-01 3.219e-02
#> [26,] 24 5.754e-01 2.933e-02
#> [27,] 26 5.936e-01 2.673e-02
#> [28,] 30 6.110e-01 2.435e-02
#> [29,] 31 6.279e-01 2.219e-02
#> [30,] 34 6.436e-01 2.022e-02
#> [31,] 34 6.586e-01 1.842e-02
#> [32,] 37 6.723e-01 1.679e-02
#> [33,] 38 6.856e-01 1.529e-02
#> [34,] 38 6.977e-01 1.394e-02
#> [35,] 40 7.095e-01 1.270e-02
#> [36,] 40 7.205e-01 1.157e-02
#> [37,] 46 7.311e-01 1.054e-02
#> [38,] 48 7.421e-01 9.605e-03
#> [39,] 50 7.527e-01 8.752e-03
#> [40,] 53 7.631e-01 7.975e-03
#> [41,] 56 7.730e-01 7.266e-03
#> [42,] 58 7.833e-01 6.621e-03
#> [43,] 59 7.931e-01 6.032e-03
#> [44,] 61 8.024e-01 5.497e-03
#> [45,] 61 8.112e-01 5.008e-03
#> [46,] 60 8.197e-01 4.563e-03
#> [47,] 60 8.277e-01 4.158e-03
#> [48,] 61 8.351e-01 3.789e-03
#> [49,] 60 8.424e-01 3.452e-03
#> [50,] 61 8.494e-01 3.145e-03
#> [51,] 61 8.560e-01 2.866e-03
#> [52,] 62 8.623e-01 2.611e-03
#> [53,] 63 8.689e-01 2.379e-03
#> [54,] 65 8.755e-01 2.168e-03
#> [55,] 66 8.821e-01 1.975e-03
#> [56,] 66 8.884e-01 1.800e-03
#> [57,] 66 8.947e-01 1.640e-03
#> [58,] 67 9.007e-01 1.494e-03
#> [59,] 68 9.074e-01 1.362e-03
#> [60,] 69 9.147e-01 1.241e-03
#> [61,] 69 9.214e-01 1.130e-03
#> [62,] 71 9.285e-01 1.030e-03
#> [63,] 69 9.352e-01 9.384e-04
#> [64,] 69 9.412e-01 8.551e-04

```

```

#> [65,] 70 9.468e-01 7.791e-04
#> [66,] 69 9.517e-01 7.099e-04
#> [67,] 70 9.562e-01 6.468e-04
#> [68,] 70 9.603e-01 5.894e-04
#> [69,] 70 9.640e-01 5.370e-04
#> [70,] 72 9.673e-01 4.893e-04
#> [71,] 72 9.704e-01 4.458e-04
#> [72,] 71 9.731e-01 4.062e-04
#> [73,] 70 9.756e-01 3.701e-04
#> [74,] 71 9.778e-01 3.373e-04
#> [75,] 71 9.799e-01 3.073e-04
#> [76,] 71 9.817e-01 2.800e-04
#> [77,] 71 9.834e-01 2.551e-04
#> [78,] 71 9.849e-01 2.325e-04
#> [79,] 72 9.863e-01 2.118e-04
#> [80,] 72 9.875e-01 1.930e-04
#> [81,] 72 9.886e-01 1.758e-04
#> [82,] 72 9.897e-01 1.602e-04
#> [83,] 72 9.906e-01 1.460e-04
#> [84,] 72 9.914e-01 1.330e-04
#> [85,] 72 9.922e-01 1.212e-04
#> [86,] 71 9.929e-01 1.104e-04
#> [87,] 71 9.935e-01 1.006e-04
#> [88,] 71 9.941e-01 9.169e-05
#> [89,] 71 9.946e-01 8.354e-05
#> [90,] 71 9.951e-01 7.612e-05
#> [91,] 71 9.955e-01 6.936e-05
#> [92,] 71 9.959e-01 6.320e-05
#> [93,] 71 9.963e-01 5.758e-05
#> [94,] 71 9.966e-01 5.247e-05
#> [95,] 71 9.969e-01 4.781e-05
#> [96,] 71 9.972e-01 4.356e-05
#> [97,] 71 9.974e-01 3.969e-05
#> [98,] 71 9.976e-01 3.616e-05
#> [99,] 71 9.979e-01 3.295e-05
#> [100,] 71 9.980e-01 3.002e-05
#> [1] "done bootstrap 1"
#> Warning: model fit failed for Fold01.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 64 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 79 84 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 1 49 80 appear to be constant within groups
#> Warning: model fit failed for Fold04.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 46 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 31 58 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold07.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 25 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 17 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep1: parameter=none Error in lda.default(x, grouping, ...) :

```

```

#> variable 57 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 93 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 31 93 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold04.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 1 5 58 appear to be constant within groups
#> Warning: model fit failed for Fold05.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 64 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold07.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 25 46 57 79 appear to be constant within groups
#> Warning: model fit failed for Fold08.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 17 49 80 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold10.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 84 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 17 93 appear to be constant within groups
#> Warning: model fit failed for Fold02.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 49 58 80 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 25 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 1 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold06.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 31 57 79 84 appear to be constant within groups
#> Warning: model fit failed for Fold07.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 46 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold10.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 64 appears to be constant within groups
#> Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
#> trainInfo, : There were missing values in resampled performance measures.
#> Warning in lda.default(x, grouping, ...): variables are collinear
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -2.563e-15 2.500e-01
#> [2,]  1  3.078e-02 2.278e-01
#> [3,]  1  5.719e-02 2.075e-01
#> [4,]  2  8.902e-02 1.891e-01
#> [5,]  2  1.170e-01 1.723e-01
#> [6,]  3  1.442e-01 1.570e-01
#> [7,]  6  1.738e-01 1.430e-01
#> [8,]  7  2.022e-01 1.303e-01

```



```

#> [9,] 7 2.278e-01 1.188e-01
#> [10,] 8 2.511e-01 1.082e-01
#> [11,] 10 2.761e-01 9.859e-02
#> [12,] 10 2.989e-01 8.983e-02
#> [13,] 10 3.193e-01 8.185e-02
#> [14,] 11 3.381e-01 7.458e-02
#> [15,] 14 3.586e-01 6.795e-02
#> [16,] 14 3.780e-01 6.192e-02
#> [17,] 14 3.954e-01 5.642e-02
#> [18,] 14 4.112e-01 5.140e-02
#> [19,] 18 4.276e-01 4.684e-02
#> [20,] 19 4.465e-01 4.268e-02
#> [21,] 22 4.677e-01 3.889e-02
#> [22,] 23 4.875e-01 3.543e-02
#> [23,] 25 5.064e-01 3.228e-02
#> [24,] 25 5.237e-01 2.942e-02
#> [25,] 27 5.396e-01 2.680e-02
#> [26,] 34 5.559e-01 2.442e-02
#> [27,] 37 5.737e-01 2.225e-02
#> [28,] 39 5.901e-01 2.028e-02
#> [29,] 40 6.053e-01 1.847e-02
#> [30,] 40 6.193e-01 1.683e-02
#> [31,] 42 6.320e-01 1.534e-02
#> [32,] 46 6.449e-01 1.397e-02
#> [33,] 48 6.572e-01 1.273e-02
#> [34,] 51 6.695e-01 1.160e-02
#> [35,] 53 6.811e-01 1.057e-02
#> [36,] 54 6.921e-01 9.632e-03
#> [37,] 56 7.029e-01 8.777e-03
#> [38,] 60 7.141e-01 7.997e-03
#> [39,] 64 7.259e-01 7.286e-03
#> [40,] 64 7.376e-01 6.639e-03
#> [41,] 65 7.488e-01 6.049e-03
#> [42,] 64 7.594e-01 5.512e-03
#> [43,] 64 7.692e-01 5.022e-03
#> [44,] 65 7.785e-01 4.576e-03
#> [45,] 66 7.874e-01 4.170e-03
#> [46,] 70 7.959e-01 3.799e-03
#> [47,] 70 8.043e-01 3.462e-03
#> [48,] 70 8.124e-01 3.154e-03
#> [49,] 71 8.199e-01 2.874e-03
#> [50,] 71 8.274e-01 2.619e-03
#> [51,] 70 8.346e-01 2.386e-03
#> [52,] 71 8.418e-01 2.174e-03
#> [53,] 71 8.495e-01 1.981e-03
#> [54,] 71 8.581e-01 1.805e-03
#> [55,] 72 8.670e-01 1.645e-03
#> [56,] 71 8.759e-01 1.498e-03
#> [57,] 73 8.844e-01 1.365e-03
#> [58,] 72 8.926e-01 1.244e-03
#> [59,] 70 9.004e-01 1.134e-03
#> [60,] 71 9.079e-01 1.033e-03
#> [61,] 71 9.152e-01 9.411e-04

```

```

#> [62,] 70 9.225e-01 8.575e-04
#> [63,] 69 9.292e-01 7.813e-04
#> [64,] 70 9.355e-01 7.119e-04
#> [65,] 70 9.414e-01 6.487e-04
#> [66,] 71 9.467e-01 5.910e-04
#> [67,] 71 9.516e-01 5.385e-04
#> [68,] 70 9.561e-01 4.907e-04
#> [69,] 70 9.601e-01 4.471e-04
#> [70,] 70 9.639e-01 4.074e-04
#> [71,] 69 9.671e-01 3.712e-04
#> [72,] 69 9.701e-01 3.382e-04
#> [73,] 70 9.728e-01 3.082e-04
#> [74,] 70 9.753e-01 2.808e-04
#> [75,] 69 9.775e-01 2.558e-04
#> [76,] 69 9.796e-01 2.331e-04
#> [77,] 69 9.814e-01 2.124e-04
#> [78,] 69 9.831e-01 1.935e-04
#> [79,] 69 9.846e-01 1.763e-04
#> [80,] 69 9.860e-01 1.607e-04
#> [81,] 68 9.872e-01 1.464e-04
#> [82,] 68 9.884e-01 1.334e-04
#> [83,] 68 9.894e-01 1.215e-04
#> [84,] 68 9.903e-01 1.107e-04
#> [85,] 68 9.912e-01 1.009e-04
#> [86,] 68 9.920e-01 9.195e-05
#> [87,] 68 9.927e-01 8.378e-05
#> [88,] 69 9.933e-01 7.633e-05
#> [89,] 69 9.939e-01 6.955e-05
#> [90,] 70 9.944e-01 6.337e-05
#> [91,] 70 9.949e-01 5.774e-05
#> [92,] 70 9.954e-01 5.261e-05
#> [93,] 70 9.958e-01 4.794e-05
#> [94,] 70 9.961e-01 4.368e-05
#> [95,] 69 9.965e-01 3.980e-05
#> [96,] 69 9.968e-01 3.627e-05
#> [97,] 71 9.971e-01 3.304e-05
#> [98,] 71 9.973e-01 3.011e-05
#> [99,] 71 9.975e-01 2.743e-05
#> [100,] 71 9.978e-01 2.500e-05
#> [1] "done bootstrap 2"

```

2.3 Summarise results

```

LSOLDA_dat <- bootstrap_scGPS(nboots = 2, mixedpop1 = mixedpop1,
  mixedpop2 = mixedpop2, genes = genes, c_selectID, listData = list())
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold04.Rep1: parameter=none Error in lda.default(x, grouping, ...) :

```



```

#> variables 48 81 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold06.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 80 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold09.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 56 63 73 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold02.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 48 81 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 56 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold07.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 63 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 80 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold10.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 73 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold02.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 63 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold04.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 48 81 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold07.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 80 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 56 73 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear

#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
#> trainInfo, : There were missing values in resampled performance measures.
#> Warning in lda.default(x, grouping, ...): variables are collinear
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -2.563e-15 2.782e-01
#> [2,]  1  3.801e-02 2.534e-01

```

```

#> [3,] 1 7.008e-02 2.309e-01
#> [4,] 2 9.924e-02 2.104e-01
#> [5,] 2 1.316e-01 1.917e-01
#> [6,] 2 1.595e-01 1.747e-01
#> [7,] 2 1.840e-01 1.592e-01
#> [8,] 3 2.073e-01 1.450e-01
#> [9,] 3 2.286e-01 1.321e-01
#> [10,] 3 2.476e-01 1.204e-01
#> [11,] 4 2.657e-01 1.097e-01
#> [12,] 6 2.861e-01 9.996e-02
#> [13,] 6 3.056e-01 9.108e-02
#> [14,] 7 3.242e-01 8.299e-02
#> [15,] 7 3.419e-01 7.562e-02
#> [16,] 7 3.577e-01 6.890e-02
#> [17,] 7 3.719e-01 6.278e-02
#> [18,] 8 3.868e-01 5.720e-02
#> [19,] 11 4.016e-01 5.212e-02
#> [20,] 16 4.179e-01 4.749e-02
#> [21,] 19 4.358e-01 4.327e-02
#> [22,] 21 4.540e-01 3.943e-02
#> [23,] 24 4.711e-01 3.592e-02
#> [24,] 26 4.869e-01 3.273e-02
#> [25,] 28 5.019e-01 2.983e-02
#> [26,] 30 5.161e-01 2.718e-02
#> [27,] 32 5.299e-01 2.476e-02
#> [28,] 33 5.428e-01 2.256e-02
#> [29,] 37 5.548e-01 2.056e-02
#> [30,] 41 5.673e-01 1.873e-02
#> [31,] 44 5.809e-01 1.707e-02
#> [32,] 46 5.939e-01 1.555e-02
#> [33,] 46 6.064e-01 1.417e-02
#> [34,] 53 6.179e-01 1.291e-02
#> [35,] 57 6.300e-01 1.176e-02
#> [36,] 60 6.431e-01 1.072e-02
#> [37,] 61 6.564e-01 9.766e-03
#> [38,] 64 6.694e-01 8.899e-03
#> [39,] 65 6.821e-01 8.108e-03
#> [40,] 66 6.941e-01 7.388e-03
#> [41,] 68 7.057e-01 6.732e-03
#> [42,] 69 7.170e-01 6.134e-03
#> [43,] 71 7.279e-01 5.589e-03
#> [44,] 72 7.384e-01 5.092e-03
#> [45,] 73 7.484e-01 4.640e-03
#> [46,] 73 7.577e-01 4.228e-03
#> [47,] 73 7.665e-01 3.852e-03
#> [48,] 73 7.751e-01 3.510e-03
#> [49,] 73 7.833e-01 3.198e-03
#> [50,] 74 7.914e-01 2.914e-03
#> [51,] 74 7.994e-01 2.655e-03
#> [52,] 76 8.078e-01 2.419e-03
#> [53,] 76 8.167e-01 2.204e-03
#> [54,] 76 8.250e-01 2.008e-03
#> [55,] 76 8.334e-01 1.830e-03

```

```

#> [56,] 78 8.424e-01 1.667e-03
#> [57,] 78 8.514e-01 1.519e-03
#> [58,] 79 8.609e-01 1.384e-03
#> [59,] 80 8.714e-01 1.261e-03
#> [60,] 79 8.820e-01 1.149e-03
#> [61,] 78 8.923e-01 1.047e-03
#> [62,] 79 9.018e-01 9.542e-04
#> [63,] 79 9.106e-01 8.694e-04
#> [64,] 79 9.185e-01 7.922e-04
#> [65,] 78 9.258e-01 7.218e-04
#> [66,] 79 9.326e-01 6.577e-04
#> [67,] 80 9.388e-01 5.993e-04
#> [68,] 80 9.445e-01 5.460e-04
#> [69,] 79 9.496e-01 4.975e-04
#> [70,] 79 9.542e-01 4.533e-04
#> [71,] 78 9.584e-01 4.130e-04
#> [72,] 78 9.623e-01 3.764e-04
#> [73,] 77 9.657e-01 3.429e-04
#> [74,] 77 9.688e-01 3.125e-04
#> [75,] 77 9.716e-01 2.847e-04
#> [76,] 77 9.742e-01 2.594e-04
#> [77,] 77 9.765e-01 2.364e-04
#> [78,] 77 9.786e-01 2.154e-04
#> [79,] 77 9.805e-01 1.962e-04
#> [80,] 77 9.823e-01 1.788e-04
#> [81,] 77 9.839e-01 1.629e-04
#> [82,] 77 9.853e-01 1.484e-04
#> [83,] 77 9.866e-01 1.353e-04
#> [84,] 77 9.878e-01 1.232e-04
#> [85,] 76 9.889e-01 1.123e-04
#> [86,] 76 9.899e-01 1.023e-04
#> [87,] 76 9.907e-01 9.323e-05
#> [88,] 77 9.916e-01 8.494e-05
#> [89,] 77 9.923e-01 7.740e-05
#> [90,] 78 9.930e-01 7.052e-05
#> [91,] 78 9.936e-01 6.426e-05
#> [92,] 79 9.942e-01 5.855e-05
#> [93,] 79 9.947e-01 5.335e-05
#> [94,] 78 9.951e-01 4.861e-05
#> [95,] 78 9.956e-01 4.429e-05
#> [96,] 78 9.960e-01 4.036e-05
#> [97,] 78 9.963e-01 3.677e-05
#> [98,] 78 9.966e-01 3.350e-05
#> [99,] 78 9.969e-01 3.053e-05
#> [100,] 78 9.972e-01 2.782e-05
#> [1] "done bootstrap 1"
#> Warning: model fit failed for Fold01.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 76 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 77 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 28 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep1: parameter=none Error in lda.default(x, grouping, ...) :

```

```

#> variable 44 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 9 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold09.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 55 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 82 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variables 5 28 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 76 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 9 appears to be constant within groups
#> Warning: model fit failed for Fold06.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 77 appears to be constant within groups
#> Warning: model fit failed for Fold07.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 82 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 55 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 44 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 77 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 82 appears to be constant within groups
#> Warning: model fit failed for Fold06.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 55 appears to be constant within groups
#> Warning: model fit failed for Fold07.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 9 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 5 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 76 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#> variable 44 appears to be constant within groups
#> Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
#> trainInfo, : There were missing values in resampled performance measures.
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -2.563e-15 2.430e-01
#> [2,]  2  3.518e-02 2.214e-01
#> [3,]  2  6.941e-02 2.017e-01
#> [4,]  3  9.954e-02 1.838e-01
#> [5,]  4  1.338e-01 1.675e-01
#> [6,]  5  1.671e-01 1.526e-01
#> [7,]  6  1.991e-01 1.390e-01
#> [8,]  6  2.297e-01 1.267e-01
#> [9,]  6  2.567e-01 1.154e-01
#> [10,] 6  2.806e-01 1.052e-01
#> [11,] 7  3.037e-01 9.584e-02

```

```

#> [12,] 7 3.243e-01 8.733e-02
#> [13,] 7 3.426e-01 7.957e-02
#> [14,] 7 3.589e-01 7.250e-02
#> [15,] 8 3.745e-01 6.606e-02
#> [16,] 10 3.889e-01 6.019e-02
#> [17,] 13 4.052e-01 5.484e-02
#> [18,] 14 4.218e-01 4.997e-02
#> [19,] 17 4.406e-01 4.553e-02
#> [20,] 19 4.594e-01 4.149e-02
#> [21,] 19 4.761e-01 3.780e-02
#> [22,] 22 4.929e-01 3.444e-02
#> [23,] 25 5.104e-01 3.138e-02
#> [24,] 27 5.300e-01 2.860e-02
#> [25,] 28 5.483e-01 2.605e-02
#> [26,] 32 5.651e-01 2.374e-02
#> [27,] 34 5.809e-01 2.163e-02
#> [28,] 37 5.958e-01 1.971e-02
#> [29,] 40 6.105e-01 1.796e-02
#> [30,] 41 6.243e-01 1.636e-02
#> [31,] 40 6.372e-01 1.491e-02
#> [32,] 40 6.484e-01 1.358e-02
#> [33,] 42 6.586e-01 1.238e-02
#> [34,] 45 6.683e-01 1.128e-02
#> [35,] 46 6.781e-01 1.028e-02
#> [36,] 47 6.873e-01 9.364e-03
#> [37,] 49 6.961e-01 8.532e-03
#> [38,] 49 7.054e-01 7.774e-03
#> [39,] 51 7.142e-01 7.083e-03
#> [40,] 54 7.229e-01 6.454e-03
#> [41,] 55 7.314e-01 5.881e-03
#> [42,] 56 7.395e-01 5.358e-03
#> [43,] 59 7.472e-01 4.882e-03
#> [44,] 61 7.557e-01 4.448e-03
#> [45,] 64 7.641e-01 4.053e-03
#> [46,] 66 7.730e-01 3.693e-03
#> [47,] 68 7.819e-01 3.365e-03
#> [48,] 70 7.908e-01 3.066e-03
#> [49,] 73 7.998e-01 2.794e-03
#> [50,] 74 8.088e-01 2.546e-03
#> [51,] 75 8.179e-01 2.319e-03
#> [52,] 75 8.270e-01 2.113e-03
#> [53,] 73 8.360e-01 1.926e-03
#> [54,] 72 8.449e-01 1.755e-03
#> [55,] 73 8.536e-01 1.599e-03
#> [56,] 72 8.619e-01 1.457e-03
#> [57,] 72 8.699e-01 1.327e-03
#> [58,] 72 8.776e-01 1.209e-03
#> [59,] 73 8.855e-01 1.102e-03
#> [60,] 73 8.932e-01 1.004e-03
#> [61,] 74 9.007e-01 9.148e-04
#> [62,] 74 9.078e-01 8.336e-04
#> [63,] 72 9.150e-01 7.595e-04
#> [64,] 72 9.214e-01 6.920e-04

```

```

#> [65,] 72 9.277e-01 6.306e-04
#> [66,] 72 9.336e-01 5.745e-04
#> [67,] 73 9.392e-01 5.235e-04
#> [68,] 72 9.443e-01 4.770e-04
#> [69,] 72 9.489e-01 4.346e-04
#> [70,] 73 9.532e-01 3.960e-04
#> [71,] 71 9.573e-01 3.608e-04
#> [72,] 71 9.611e-01 3.288e-04
#> [73,] 71 9.646e-01 2.996e-04
#> [74,] 70 9.677e-01 2.730e-04
#> [75,] 70 9.706e-01 2.487e-04
#> [76,] 70 9.732e-01 2.266e-04
#> [77,] 71 9.756e-01 2.065e-04
#> [78,] 71 9.778e-01 1.881e-04
#> [79,] 69 9.797e-01 1.714e-04
#> [80,] 69 9.815e-01 1.562e-04
#> [81,] 69 9.831e-01 1.423e-04
#> [82,] 70 9.846e-01 1.297e-04
#> [83,] 70 9.860e-01 1.182e-04
#> [84,] 70 9.872e-01 1.077e-04
#> [85,] 70 9.883e-01 9.809e-05
#> [86,] 70 9.894e-01 8.938e-05
#> [87,] 70 9.903e-01 8.144e-05
#> [88,] 70 9.911e-01 7.420e-05
#> [89,] 70 9.919e-01 6.761e-05
#> [90,] 70 9.926e-01 6.161e-05
#> [91,] 70 9.933e-01 5.613e-05
#> [92,] 70 9.939e-01 5.115e-05
#> [93,] 70 9.944e-01 4.660e-05
#> [94,] 70 9.949e-01 4.246e-05
#> [95,] 70 9.953e-01 3.869e-05
#> [96,] 71 9.957e-01 3.525e-05
#> [97,] 71 9.961e-01 3.212e-05
#> [98,] 71 9.964e-01 2.927e-05
#> [99,] 70 9.967e-01 2.667e-05
#> [100,] 71 9.970e-01 2.430e-05
#> [1] "done bootstrap 2"

# display the list of result information in the LASOLDA_dat object
names(LSOLDA_dat)
#> [1] "Accuracy"      "LassoGenes"    "Deviance"      "LassoFit"
#> [5] "LDAFit"        "predictor_S1"  "LassoPredict"  "LDAPredict"
LSOLDA_dat$LassoPredict
#> [[1]]
#> [[1]][[1]]
#> [1] "LASSO for subpop1 in target mixedpop2"
#>
#> [[1]][[2]]
#> [1] 9.090909
#>
#> [[1]][[3]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>

```

```

#> [[1]][[4]]
#> [1] 53.57143
#>
#> [[1]][[5]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 3.007519
#>
#> [[1]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 12.5
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LASSO for subpop1 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 74.86631
#>
#> [[2]][[3]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 97.14286
#>
#> [[2]][[5]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 54.13534
#>
#> [[2]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 70
LSOLDA_dat$LDAPredict
#> [[1]]
#> [[1]][[1]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[1]][[2]]
#> [1] 0.5347594
#>
#> [[1]][[3]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 34.28571

```



```

#>
#> [[1]][[5]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 5.263158
#>
#> [[1]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 15
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 80.2139
#>
#> [[2]][[3]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 60
#>
#> [[2]][[5]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 51.12782
#>
#> [[2]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 55

# summary results LDA
summary_prediction_lda(LSOLDA_dat = LSOLDA_dat, nPredSubpop = 4)
#>           V1           V2           names
#> 1 0.53475935828877 80.2139037433155 LDA for subpop 1 in target mixedpop2
#> 2 34.2857142857143      60 LDA for subpop 2 in target mixedpop2
#> 3 5.26315789473684 51.1278195488722 LDA for subpop 3 in target mixedpop2
#> 4           15           55 LDA for subpop 4 in target mixedpop2

# summary results Lasso
summary_prediction_lasso(LSOLDA_dat = LSOLDA_dat, nPredSubpop = 4)
#>           V1           V2           names
#> 1 9.09090909090909 74.8663101604278 LASSO for subpop1 in target mixedpop2
#> 2 53.5714285714286 97.1428571428571 LASSO for subpop2 in target mixedpop2
#> 3 3.00751879699248 54.1353383458647 LASSO for subpop3 in target mixedpop2

```

```

#> 4                12.5                70 LASSO for subpop4 in target mixedpop2

# summary deviance
summary_deviance(object = LSOLDA_dat)
#> $allDeviance
#> [1] "0.3577" "0.3243"
#>
#> $DeviMax
#>      Dfd  Deviance      DEgenes
#> 1      0 -2.563e-15 genes_cluster1
#> 2      1  0.07008 genes_cluster1
#> 3      2   0.184 genes_cluster1
#> 4      3   0.2476 genes_cluster1
#> 5      4   0.2657 genes_cluster1
#> 6      6   0.3056 genes_cluster1
#> 7      7   0.3577 genes_cluster1
#> 8 remaining      1      DEgenes
#>
#> $LassoGenesMax
#>                1                name
#> (Intercept)      0.21049408      (Intercept)
#> FN1_ENSG00000115414 -0.01081277 FN1_ENSG00000115414
#> T_ENSG00000164458   0.15093144   T_ENSG00000164458
#> SOX17_ENSG00000164736 -0.05404855 SOX17_ENSG00000164736
#> MESP1_ENSG00000166823  0.03922346 MESP1_ENSG00000166823
#> FOXF1_ENSG00000103241  0.18054447 FOXF1_ENSG00000103241
#> SNAI1_ENSG00000124216  0.16535395 SNAI1_ENSG00000124216
#> FOXA3_ENSG00000170608 -0.16336103 FOXA3_ENSG00000170608

```

3. A complete workflow of the scGPS:

The purpose of this workflow is to solve the following task: given an unknown mixed population, find clusters and estimate relationship between clusters

3.1 Identify clusters in a using CORE

(skip this step if clusters are known)

```

#Let's find clustering information in an expresion data
day5 <- sample2
cellnames <- colnames(day5$dat5_counts)
cluster <- day5$dat5_clusters
cellnames <- data.frame("Cluster"=cluster, "cellBarcodes" = cellnames)
mixedpop2 <- NewscGPS_SME(ExpressionMatrix = day5$dat5_counts, GeneMetadata = day5$dat5geneInfo, CellMeta

CORE_cluster <- CORE_scGPS(mixedpop2, remove_outlier = c(0), PCA=FALSE)
#> [1] "Identifying top variable genes"
#> [1] "Calculating distance matrix"
#> [1] "Performing hierarchical clustering"
#> [1] "Finding clustering information"
#> [1] "No more outliers detected after 1 filtering round"

```

```

#> [1] "writing clustering result for run 1"
#> [1] "writing clustering result for run 2"
#> [1] "writing clustering result for run 3"
#> [1] "writing clustering result for run 4"
#> [1] "writing clustering result for run 5"
#> [1] "writing clustering result for run 6"
#> [1] "writing clustering result for run 7"
#> [1] "writing clustering result for run 8"
#> [1] "writing clustering result for run 9"
#> [1] "writing clustering result for run 10"
#> [1] "writing clustering result for run 11"
#> [1] "writing clustering result for run 12"
#> [1] "writing clustering result for run 13"
#> [1] "writing clustering result for run 14"
#> [1] "writing clustering result for run 15"
#> [1] "writing clustering result for run 16"
#> [1] "writing clustering result for run 17"
#> [1] "writing clustering result for run 18"
#> [1] "writing clustering result for run 19"
#> [1] "writing clustering result for run 20"
#> [1] "writing clustering result for run 21"
#> [1] "writing clustering result for run 22"
#> [1] "writing clustering result for run 23"
#> [1] "writing clustering result for run 24"
#> [1] "writing clustering result for run 25"
#> [1] "writing clustering result for run 26"
#> [1] "writing clustering result for run 27"
#> [1] "writing clustering result for run 28"
#> [1] "writing clustering result for run 29"
#> [1] "writing clustering result for run 30"
#> [1] "writing clustering result for run 31"
#> [1] "writing clustering result for run 32"
#> [1] "writing clustering result for run 33"
#> [1] "writing clustering result for run 34"
#> [1] "writing clustering result for run 35"
#> [1] "writing clustering result for run 36"
#> [1] "writing clustering result for run 37"
#> [1] "writing clustering result for run 38"
#> [1] "writing clustering result for run 39"
#> [1] "writing clustering result for run 40"
#> [1] "Done clustering, moving to stability calculation..."
#> [1] "Done calculating stability..."
#> [1] "Start finding optimal clustering..."
#> [1] "Done finding optimal clustering..."

```

3.2 Visualise all cluster results in all iterations

```

#plot with default colors
plot_CORE(CORE_cluster$tree, CORE_cluster$Cluster)

#let's find the CORE clusters
CORE_cluster <- CORE_scGPS(mixedpop2, remove_outlier = c(0), PCA=FALSE)

```

```

#> [1] "Identifying top variable genes"
#> [1] "Calculating distance matrix"
#> [1] "Performing hierarchical clustering"
#> [1] "Finding clustering information"
#> [1] "No more outliers detected after 1 filtering round"
#> [1] "writing clustering result for run 1"
#> [1] "writing clustering result for run 2"
#> [1] "writing clustering result for run 3"
#> [1] "writing clustering result for run 4"
#> [1] "writing clustering result for run 5"
#> [1] "writing clustering result for run 6"
#> [1] "writing clustering result for run 7"
#> [1] "writing clustering result for run 8"
#> [1] "writing clustering result for run 9"
#> [1] "writing clustering result for run 10"
#> [1] "writing clustering result for run 11"
#> [1] "writing clustering result for run 12"
#> [1] "writing clustering result for run 13"
#> [1] "writing clustering result for run 14"
#> [1] "writing clustering result for run 15"
#> [1] "writing clustering result for run 16"
#> [1] "writing clustering result for run 17"
#> [1] "writing clustering result for run 18"
#> [1] "writing clustering result for run 19"
#> [1] "writing clustering result for run 20"
#> [1] "writing clustering result for run 21"
#> [1] "writing clustering result for run 22"
#> [1] "writing clustering result for run 23"
#> [1] "writing clustering result for run 24"
#> [1] "writing clustering result for run 25"
#> [1] "writing clustering result for run 26"
#> [1] "writing clustering result for run 27"
#> [1] "writing clustering result for run 28"
#> [1] "writing clustering result for run 29"
#> [1] "writing clustering result for run 30"
#> [1] "writing clustering result for run 31"
#> [1] "writing clustering result for run 32"
#> [1] "writing clustering result for run 33"
#> [1] "writing clustering result for run 34"
#> [1] "writing clustering result for run 35"
#> [1] "writing clustering result for run 36"
#> [1] "writing clustering result for run 37"
#> [1] "writing clustering result for run 38"
#> [1] "writing clustering result for run 39"
#> [1] "writing clustering result for run 40"
#> [1] "Done clustering, moving to stability calculation..."
#> [1] "Done calculating stability..."
#> [1] "Start finding optimal clustering..."
#> [1] "Done finding optimal clustering..."

#let's plot all clusters
plot_CORE(CORE_cluster$tree, CORE_cluster$Cluster)

```

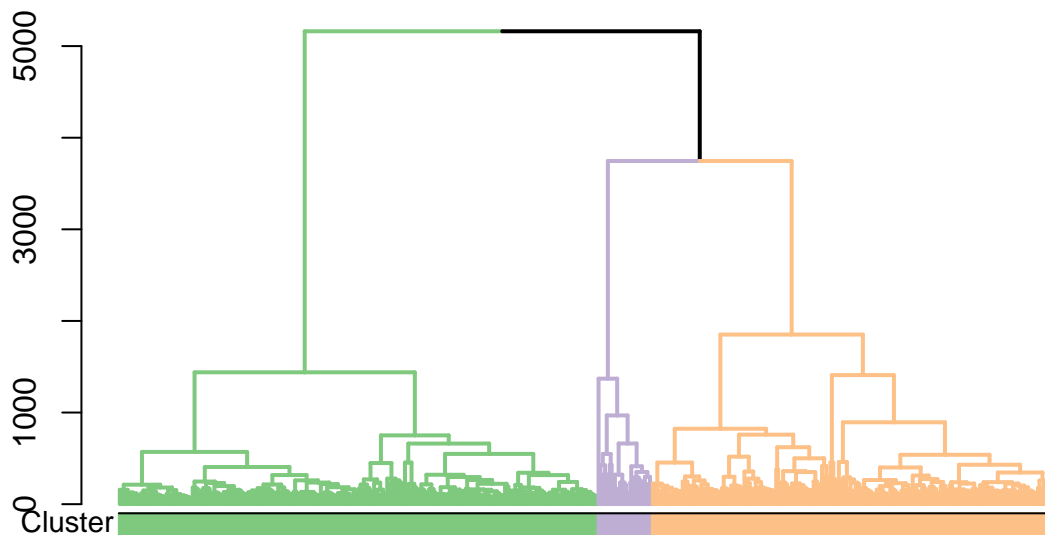
```
plot_CORE(CORE_cluster$tree, CORE_cluster$Cluster, color_branch = c("#208eb7", "#6ce9d3", "#1c5e39", "#f44336"))
```

```
#you can customise the cluster color bars (provide color_branch values)
plot_CORE(CORE_cluster$tree, CORE_cluster$Cluster, color_branch = c("#208eb7", "#6ce9d3", "#1c5e39", "#f4cccc"))
```

3.3 Plot the optimal clustering result

```
#extract optimal index identified by CORE_scGPS
optimal_index = which(CORE_cluster$optimalClust$KeyStats$Height == CORE_cluster$optimalClust$OptimalRes)

#plot the optimal result
plot_optimal_CORE(original_tree= CORE_cluster$tree, optimal_cluster = unlist(CORE_cluster$Cluster[optimal_index]))
#> [1] "Ordering and assigning labels..."
#> [1] 2
#> [1] 128 270 NA
#> [1] 3
#> [1] 128 270 393
#> [1] "Plotting the colored dendrogram now...."
```

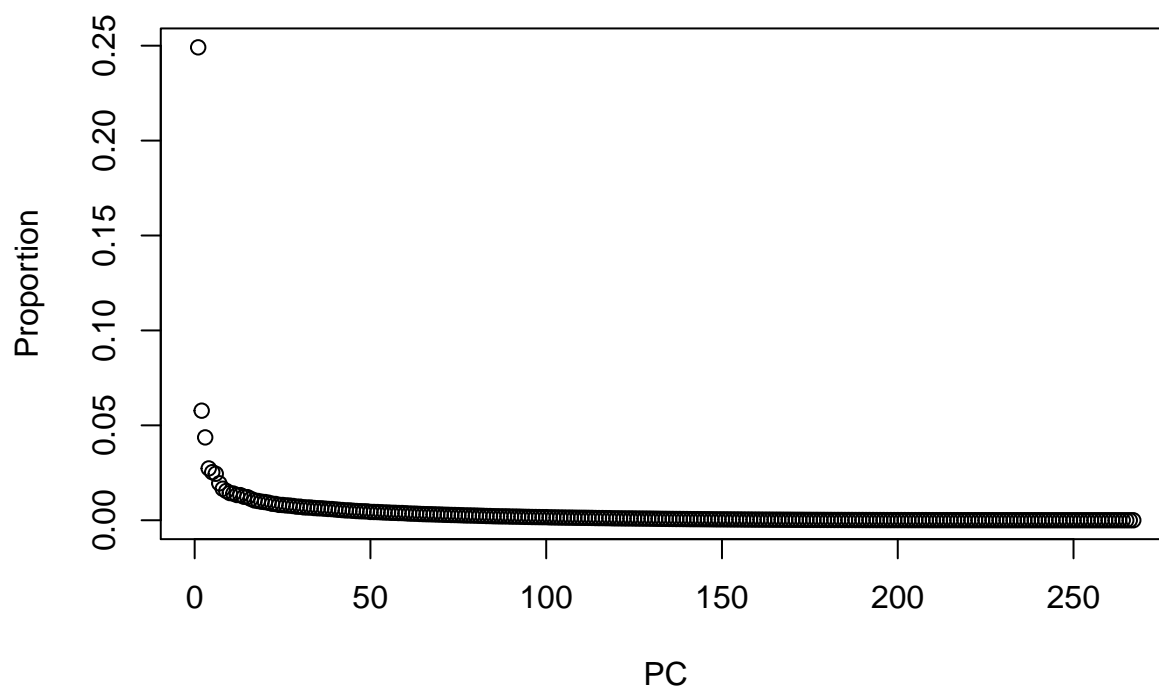


```
#> [1] "Plotting the bar underneath now...."
```

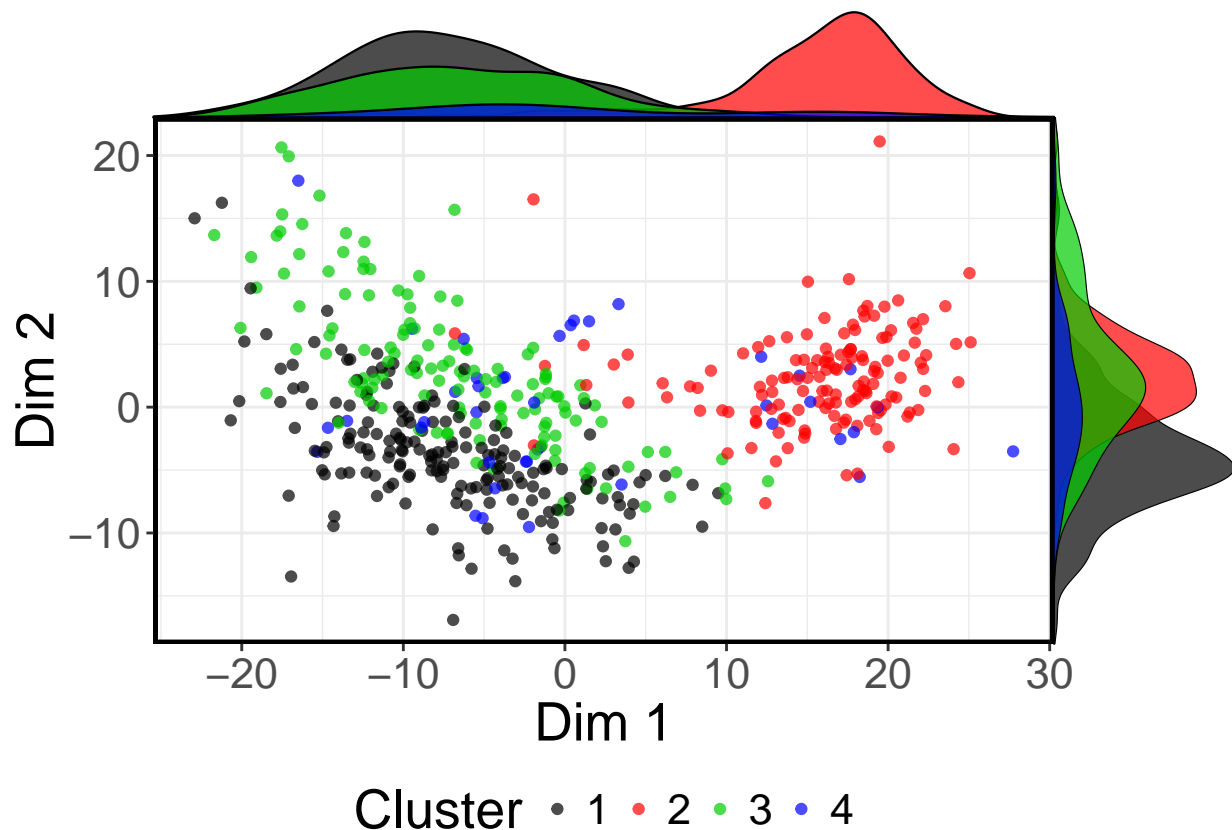
3.4 Compare clustering results with other dimensional reduction methods (e.g., CDR)

```
library(cidr)
t <- CIDR_scGPS(expression.matrix=assay(mixedpop2))
#> [1] "building cidr object..."
#> [1] "determine dropout candidates..."
#> [1] "determine the imputation weighting threshold..."
#> [1] "computes the _CIDR_ dissimilarity matrix..."
#> [1] "PCA plot with proportion of variance explained..."
```

Proportion of Variation



```
#> [1] "find the number of PC..."
#> [1] "perform clustering..."
p2 <-plotReduced_scGPS(t, color_fac = factor(colData(mixedpop2)[,1]),palletes =1:length(unique(colData(
#>
#> Attaching package: 'cowplot'
#> The following object is masked from 'package:ggplot2':
#>
#>     ggsave
p2
```

3.5 Find gene markers and annotate clusters

```
#load gene list (this can be any lists of user-selected genes)
genes <-GeneList
genes <-genes$Merged_unique

#the gene list can also be objectively identified by differential expression analysis
#cluster information is required for findMarkers_scGPS. Here, we use CORE results.

Optimal_index <- which( CORE_cluster$OptimalClust$KeyStats$Height == CORE_cluster$OptimalClust$OptimalR
colData(mixedpop2)[,1] <- unlist(CORE_cluster$Cluster[[Optimal_index]])

suppressMessages(library(locfit))
suppressMessages(library(DESeq))

DEgenes <- findMarkers_scGPS(expression_matrix=assay(mixedpop2), cluster = colData(mixedpop2)[,1],
                             selected_cluster=unique(colData(mixedpop2)[,1]))

#> [1] "Start estimate dispersions for cluster 1..."
#> [1] "Done estimate dispersions. Start nbinom test for cluster 1..."
#> [1] "Done nbinom test for cluster 1 ..."
#> [1] "Adjust foldchange by subtracting basemean to 1..."
#> [1] "Start estimate dispersions for cluster 2..."
#> [1] "Done estimate dispersions. Start nbinom test for cluster 2..."
#> [1] "Done nbinom test for cluster 2 ..."
#> [1] "Adjust foldchange by subtracting basemean to 1..."
```

```

#> [1] "Start estimate dispersions for cluster 3..."
#> [1] "Done estimate dispersions. Start nbinom test for cluster 3..."
#> [1] "Done nbinom test for cluster 3 ..."
#> [1] "Adjust foldchange by subtracting basemean to 1..."

#the output contains dataframes for each cluster.
#the data frame contains all genes, sorted by p-values
names(DEgenes)
#> [1] "DE_Subpop1vsRemaining" "DE_Subpop2vsRemaining" "DE_Subpop3vsRemaining"

#you can annotate the identified clusters
DEgeneList_3vsOthers <- DEgenes$DE_Subpop3vsRemaining$id

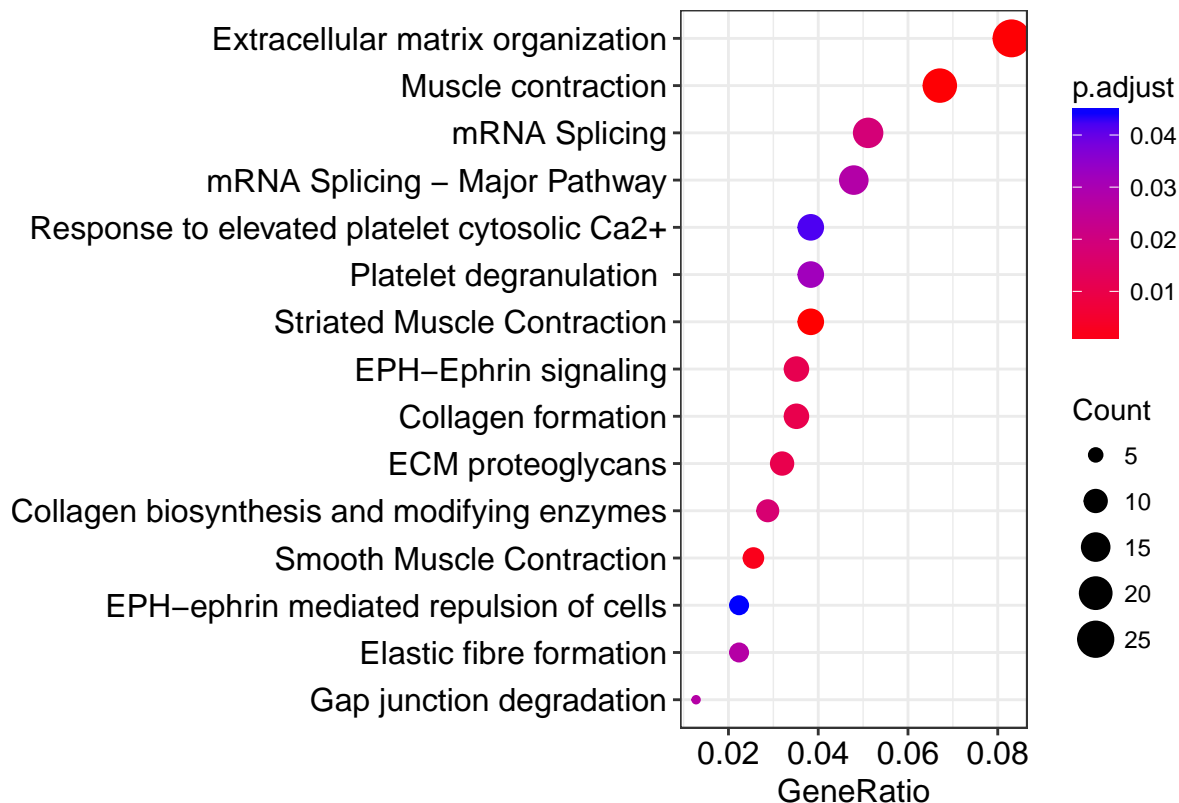
#users need to check the format of the gene input to make sure they are consistent to
#the gene names in the expression matrix
DEgeneList_3vsOthers <-gsub("_.*", "", DEgeneList_3vsOthers )

#the following command saves the file "PathwayEnrichment.xlsx" to the working dir
#use 500 top DE genes
suppressMessages(library(DOSE))
suppressMessages(library(ReactomePA))
suppressMessages(library(clusterProfiler))

enrichment_test <- annotate_scGPS(DEgeneList_3vsOthers[1:500], pvalueCutoff=0.05, gene_symbol=TRUE, outp
#> [1] "Original gene number in geneList"
#> [1] 500
#> [1] "Number of genes successfully converted"
#> [1] 490

#the enrichment outputs can be displayed by running
dotplot(enrichment_test, showCategory=15)

```



4. Relationship between clusters within one sample or between two samples

The purpose of this workflow is to solve the following task: given one or two unknown mixed population(s) and clusters in each mixed population, estimate and visualise relationship between clusters

4.1 Start the scGPS prediction to find relationship between clusters

Start the scGPS prediction to find relationship between clusters

```
#select a subpopulation, and input gene list
c_selectID <- 1
genes = DEgenes$DE_Subpop1vsRemaining$id[1:500]
#format gene names
genes <- gsub("_.*", "", genes)

#run the test bootstrap with nboots = 2 runs
sink("temp")
LSOLDA_dat <- bootstrap_scGPS(nboots = 2, mixedpop1 = mixedpop2, mixedpop2 = mixedpop2, genes=genes, c_s
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev  Lambda
#> [1,]  0 -1.922e-15 0.361200
```

```

#> [2,] 1 3.351e-02 0.344800
#> [3,] 1 6.445e-02 0.329100
#> [4,] 1 9.334e-02 0.314100
#> [5,] 1 1.206e-01 0.299900
#> [6,] 1 1.465e-01 0.286200
#> [7,] 1 1.712e-01 0.273200
#> [8,] 1 1.949e-01 0.260800
#> [9,] 1 2.178e-01 0.248900
#> [10,] 1 2.399e-01 0.237600
#> [11,] 1 2.613e-01 0.226800
#> [12,] 1 2.820e-01 0.216500
#> [13,] 1 3.021e-01 0.206700
#> [14,] 1 3.215e-01 0.197300
#> [15,] 1 3.404e-01 0.188300
#> [16,] 1 3.587e-01 0.179800
#> [17,] 1 3.764e-01 0.171600
#> [18,] 1 3.936e-01 0.163800
#> [19,] 1 4.102e-01 0.156300
#> [20,] 1 4.262e-01 0.149200
#> [21,] 2 4.421e-01 0.142500
#> [22,] 2 4.585e-01 0.136000
#> [23,] 3 4.747e-01 0.129800
#> [24,] 3 4.903e-01 0.123900
#> [25,] 4 5.054e-01 0.118300
#> [26,] 4 5.199e-01 0.112900
#> [27,] 4 5.338e-01 0.107800
#> [28,] 5 5.473e-01 0.102900
#> [29,] 5 5.604e-01 0.098190
#> [30,] 5 5.730e-01 0.093720
#> [31,] 5 5.851e-01 0.089460
#> [32,] 5 5.967e-01 0.085400
#> [33,] 5 6.078e-01 0.081520
#> [34,] 6 6.185e-01 0.077810
#> [35,] 7 6.289e-01 0.074280
#> [36,] 7 6.392e-01 0.070900
#> [37,] 7 6.491e-01 0.067680
#> [38,] 10 6.590e-01 0.064600
#> [39,] 11 6.688e-01 0.061660
#> [40,] 10 6.785e-01 0.058860
#> [41,] 10 6.878e-01 0.056190
#> [42,] 12 6.968e-01 0.053630
#> [43,] 13 7.056e-01 0.051200
#> [44,] 13 7.141e-01 0.048870
#> [45,] 13 7.223e-01 0.046650
#> [46,] 14 7.302e-01 0.044530
#> [47,] 14 7.379e-01 0.042500
#> [48,] 16 7.456e-01 0.040570
#> [49,] 17 7.532e-01 0.038730
#> [50,] 19 7.606e-01 0.036970
#> [51,] 21 7.681e-01 0.035290
#> [52,] 20 7.755e-01 0.033680
#> [53,] 19 7.825e-01 0.032150
#> [54,] 19 7.893e-01 0.030690

```

```

#> [55,] 20 7.959e-01 0.029300
#> [56,] 20 8.023e-01 0.027960
#> [57,] 21 8.087e-01 0.026690
#> [58,] 22 8.148e-01 0.025480
#> [59,] 24 8.210e-01 0.024320
#> [60,] 25 8.270e-01 0.023220
#> [61,] 25 8.329e-01 0.022160
#> [62,] 25 8.385e-01 0.021150
#> [63,] 25 8.439e-01 0.020190
#> [64,] 27 8.492e-01 0.019270
#> [65,] 29 8.544e-01 0.018400
#> [66,] 30 8.596e-01 0.017560
#> [67,] 29 8.646e-01 0.016760
#> [68,] 28 8.694e-01 0.016000
#> [69,] 28 8.741e-01 0.015270
#> [70,] 28 8.786e-01 0.014580
#> [71,] 29 8.829e-01 0.013920
#> [72,] 30 8.871e-01 0.013290
#> [73,] 31 8.913e-01 0.012680
#> [74,] 32 8.956e-01 0.012110
#> [75,] 32 8.997e-01 0.011550
#> [76,] 34 9.037e-01 0.011030
#> [77,] 35 9.077e-01 0.010530
#> [78,] 36 9.116e-01 0.010050
#> [79,] 36 9.154e-01 0.009593
#> [80,] 37 9.190e-01 0.009157
#> [81,] 37 9.224e-01 0.008741
#> [82,] 37 9.257e-01 0.008344
#> [83,] 37 9.289e-01 0.007964
#> [84,] 38 9.321e-01 0.007602
#> [85,] 38 9.351e-01 0.007257
#> [86,] 38 9.380e-01 0.006927
#> [87,] 38 9.408e-01 0.006612
#> [88,] 40 9.435e-01 0.006312
#> [89,] 40 9.460e-01 0.006025
#> [90,] 43 9.484e-01 0.005751
#> [91,] 43 9.508e-01 0.005489
#> [92,] 44 9.530e-01 0.005240
#> [93,] 45 9.551e-01 0.005002
#> [94,] 46 9.572e-01 0.004774
#> [95,] 46 9.591e-01 0.004557
#> [96,] 46 9.610e-01 0.004350
#> [97,] 46 9.627e-01 0.004153
#> [98,] 49 9.644e-01 0.003964
#> [99,] 48 9.660e-01 0.003784
#> [100,] 50 9.676e-01 0.003612
#> [1] "done bootstrap 1"
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -1.922e-15 0.371200
#> [2,]  1  3.538e-02 0.354400

```

```

#> [3,] 1 6.794e-02 0.338300
#> [4,] 1 9.815e-02 0.322900
#> [5,] 1 1.264e-01 0.308200
#> [6,] 1 1.530e-01 0.294200
#> [7,] 1 1.781e-01 0.280800
#> [8,] 1 2.020e-01 0.268100
#> [9,] 1 2.247e-01 0.255900
#> [10,] 1 2.465e-01 0.244200
#> [11,] 1 2.673e-01 0.233100
#> [12,] 1 2.874e-01 0.222500
#> [13,] 1 3.066e-01 0.212400
#> [14,] 1 3.252e-01 0.202800
#> [15,] 1 3.431e-01 0.193600
#> [16,] 1 3.603e-01 0.184800
#> [17,] 1 3.770e-01 0.176400
#> [18,] 2 3.934e-01 0.168300
#> [19,] 2 4.116e-01 0.160700
#> [20,] 2 4.290e-01 0.153400
#> [21,] 2 4.456e-01 0.146400
#> [22,] 2 4.615e-01 0.139800
#> [23,] 3 4.771e-01 0.133400
#> [24,] 3 4.923e-01 0.127400
#> [25,] 3 5.069e-01 0.121600
#> [26,] 3 5.208e-01 0.116000
#> [27,] 4 5.345e-01 0.110800
#> [28,] 4 5.476e-01 0.105700
#> [29,] 5 5.607e-01 0.100900
#> [30,] 5 5.734e-01 0.096340
#> [31,] 6 5.857e-01 0.091960
#> [32,] 6 5.974e-01 0.087780
#> [33,] 6 6.086e-01 0.083790
#> [34,] 6 6.195e-01 0.079980
#> [35,] 6 6.298e-01 0.076340
#> [36,] 6 6.398e-01 0.072870
#> [37,] 6 6.493e-01 0.069560
#> [38,] 7 6.586e-01 0.066400
#> [39,] 8 6.679e-01 0.063380
#> [40,] 9 6.773e-01 0.060500
#> [41,] 12 6.871e-01 0.057750
#> [42,] 12 6.970e-01 0.055130
#> [43,] 10 7.064e-01 0.052620
#> [44,] 10 7.152e-01 0.050230
#> [45,] 10 7.238e-01 0.047950
#> [46,] 10 7.320e-01 0.045770
#> [47,] 9 7.399e-01 0.043690
#> [48,] 9 7.475e-01 0.041700
#> [49,] 10 7.550e-01 0.039810
#> [50,] 12 7.623e-01 0.038000
#> [51,] 14 7.696e-01 0.036270
#> [52,] 16 7.768e-01 0.034620
#> [53,] 17 7.838e-01 0.033050
#> [54,] 17 7.906e-01 0.031550
#> [55,] 18 7.971e-01 0.030110

```

```

#> [56,] 19 8.034e-01 0.028740
#> [57,] 20 8.096e-01 0.027440
#> [58,] 21 8.155e-01 0.026190
#> [59,] 21 8.213e-01 0.025000
#> [60,] 21 8.269e-01 0.023860
#> [61,] 22 8.323e-01 0.022780
#> [62,] 23 8.375e-01 0.021740
#> [63,] 26 8.429e-01 0.020750
#> [64,] 28 8.484e-01 0.019810
#> [65,] 28 8.538e-01 0.018910
#> [66,] 29 8.592e-01 0.018050
#> [67,] 30 8.645e-01 0.017230
#> [68,] 31 8.698e-01 0.016450
#> [69,] 32 8.749e-01 0.015700
#> [70,] 31 8.798e-01 0.014990
#> [71,] 32 8.846e-01 0.014310
#> [72,] 35 8.892e-01 0.013660
#> [73,] 35 8.938e-01 0.013030
#> [74,] 35 8.982e-01 0.012440
#> [75,] 35 9.024e-01 0.011880
#> [76,] 37 9.064e-01 0.011340
#> [77,] 36 9.104e-01 0.010820
#> [78,] 35 9.141e-01 0.010330
#> [79,] 34 9.177e-01 0.009860
#> [80,] 37 9.211e-01 0.009412
#> [81,] 39 9.244e-01 0.008984
#> [82,] 39 9.276e-01 0.008576
#> [83,] 39 9.306e-01 0.008186
#> [84,] 40 9.335e-01 0.007814
#> [85,] 39 9.363e-01 0.007459
#> [86,] 39 9.390e-01 0.007120
#> [87,] 40 9.415e-01 0.006796
#> [88,] 43 9.440e-01 0.006487
#> [89,] 43 9.464e-01 0.006193
#> [90,] 46 9.487e-01 0.005911
#> [91,] 46 9.510e-01 0.005642
#> [92,] 47 9.531e-01 0.005386
#> [93,] 48 9.552e-01 0.005141
#> [94,] 49 9.572e-01 0.004907
#> [95,] 49 9.591e-01 0.004684
#> [96,] 48 9.609e-01 0.004472
#> [97,] 49 9.627e-01 0.004268
#> [98,] 49 9.643e-01 0.004074
#> [99,] 49 9.659e-01 0.003889
#> [100,] 49 9.674e-01 0.003712
#> [1] "done bootstrap 2"
sink()

```

4.2 Display summary results for the prediction

```

#get the number of rows for the summary matrix
row_cluster <-length(unique(colData(mixedpop2)[,1]))

```



```

#summary results LDA
summary_prediction_lda(LSOLDA_dat=LSOLDA_dat, nPredSubpop = row_cluster )
#>           V1           V2           names
#> 1      86.328125      89.84375 LDA for subpop 1 in target mixedpop2
#> 2 11.6279069767442 13.953488372093 LDA for subpop 2 in target mixedpop2
#> 3 13.7931034482759           0 LDA for subpop 3 in target mixedpop2

#summary results Lasso
summary_prediction_lasso(LSOLDA_dat=LSOLDA_dat, nPredSubpop = row_cluster)
#>           V1           V2           names
#> 1      98.046875      99.21875 LASSO for subpop1 in target mixedpop2
#> 2 4.65116279069767 6.51162790697674 LASSO for subpop2 in target mixedpop2
#> 3           NA           NA LASSO for subpop3 in target mixedpop2

#summary deviance
summary_deviance(LSOLDA_dat)
#> $allDeviance
#> [1] "0.5473" "0.5607"
#>
#> $DeviMax
#>      Dfd      Deviance      DEgenes
#> 1      0 -1.922e-15 genes_cluster1
#> 2      1      0.377 genes_cluster1
#> 3      2      0.4615 genes_cluster1
#> 4      3      0.5208 genes_cluster1
#> 5      4      0.5476 genes_cluster1
#> 6      5      0.5607 genes_cluster1
#> 7 remaining      1      DEgenes
#>
#> $LassoGenesMax
#>           1           name
#> (Intercept)      -2.5442117050      (Intercept)
#> PPIC_ENSG00000168938      0.1012042564      PPIC_ENSG00000168938
#> ACTB_ENSG00000075624      0.0005233659      ACTB_ENSG00000075624
#> MALAT1_ENSG00000251562      0.0081725012      MALAT1_ENSG00000251562
#> NPC2_ENSG00000119655      0.0034367876      NPC2_ENSG00000119655
#> TPM1_ENSG00000140416      0.0020196479      TPM1_ENSG00000140416

```

##4.3 Plot the relationship between clusters

Relationship between clusters within one sample or between two samples

```

#run prediction for 3 clusters

c_selectID <- 1
genes = DEgenes$DE_Subpop1vsRemaining$id[1:200] #top 200 gene markers distinguishing cluster 1
genes <- gsub("_.*", "", genes)

```

```

LSOLDA_dat1 <- bootstrap_scGPS(nboots = 1,mixedpop1 = mixedpop2, mixedpop2 = mixedpop2, genes=genes, c_
#>
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -1.922e-15 3.479e-01
#> [2,]  1  6.010e-02 3.170e-01
#> [3,]  1  1.136e-01 2.888e-01
#> [4,]  1  1.631e-01 2.632e-01
#> [5,]  1  2.091e-01 2.398e-01
#> [6,]  1  2.520e-01 2.185e-01
#> [7,]  1  2.922e-01 1.991e-01
#> [8,]  2  3.308e-01 1.814e-01
#> [9,]  2  3.686e-01 1.653e-01
#> [10,] 2  4.034e-01 1.506e-01
#> [11,] 3  4.368e-01 1.372e-01
#> [12,] 4  4.677e-01 1.250e-01
#> [13,] 4  4.965e-01 1.139e-01
#> [14,] 6  5.235e-01 1.038e-01
#> [15,] 7  5.490e-01 9.458e-02
#> [16,] 8  5.726e-01 8.618e-02
#> [17,] 9  5.968e-01 7.852e-02
#> [18,] 11 6.197e-01 7.155e-02
#> [19,] 12 6.414e-01 6.519e-02
#> [20,] 12 6.615e-01 5.940e-02
#> [21,] 15 6.808e-01 5.412e-02
#> [22,] 14 6.989e-01 4.931e-02
#> [23,] 15 7.153e-01 4.493e-02
#> [24,] 16 7.309e-01 4.094e-02
#> [25,] 17 7.456e-01 3.730e-02
#> [26,] 18 7.595e-01 3.399e-02
#> [27,] 19 7.727e-01 3.097e-02
#> [28,] 20 7.853e-01 2.822e-02
#> [29,] 20 7.972e-01 2.571e-02
#> [30,] 21 8.085e-01 2.343e-02
#> [31,] 21 8.191e-01 2.135e-02
#> [32,] 21 8.290e-01 1.945e-02
#> [33,] 21 8.383e-01 1.772e-02
#> [34,] 24 8.474e-01 1.615e-02
#> [35,] 24 8.566e-01 1.471e-02
#> [36,] 25 8.651e-01 1.341e-02
#> [37,] 26 8.739e-01 1.222e-02
#> [38,] 25 8.821e-01 1.113e-02
#> [39,] 28 8.899e-01 1.014e-02
#> [40,] 28 8.978e-01 9.241e-03
#> [41,] 28 9.053e-01 8.420e-03
#> [42,] 28 9.123e-01 7.672e-03
#> [43,] 28 9.188e-01 6.990e-03
#> [44,] 32 9.251e-01 6.369e-03
#> [45,] 33 9.312e-01 5.803e-03
#> [46,] 34 9.368e-01 5.288e-03
#> [47,] 35 9.421e-01 4.818e-03
#> [48,] 35 9.469e-01 4.390e-03

```

```

#> [49,] 37 9.514e-01 4.000e-03
#> [50,] 38 9.556e-01 3.645e-03
#> [51,] 41 9.595e-01 3.321e-03
#> [52,] 42 9.631e-01 3.026e-03
#> [53,] 42 9.663e-01 2.757e-03
#> [54,] 42 9.693e-01 2.512e-03
#> [55,] 43 9.720e-01 2.289e-03
#> [56,] 42 9.745e-01 2.086e-03
#> [57,] 43 9.768e-01 1.900e-03
#> [58,] 44 9.788e-01 1.732e-03
#> [59,] 43 9.807e-01 1.578e-03
#> [60,] 43 9.824e-01 1.438e-03
#> [61,] 45 9.840e-01 1.310e-03
#> [62,] 45 9.854e-01 1.193e-03
#> [63,] 47 9.867e-01 1.087e-03
#> [64,] 48 9.879e-01 9.908e-04
#> [65,] 48 9.890e-01 9.028e-04
#> [66,] 49 9.899e-01 8.226e-04
#> [67,] 49 9.908e-01 7.495e-04
#> [68,] 50 9.917e-01 6.829e-04
#> [69,] 52 9.924e-01 6.223e-04
#> [70,] 52 9.931e-01 5.670e-04
#> [71,] 52 9.937e-01 5.166e-04
#> [72,] 52 9.943e-01 4.707e-04
#> [73,] 52 9.948e-01 4.289e-04
#> [74,] 53 9.952e-01 3.908e-04
#> [75,] 53 9.957e-01 3.561e-04
#> [76,] 53 9.960e-01 3.245e-04
#> [77,] 53 9.964e-01 2.956e-04
#> [78,] 53 9.967e-01 2.694e-04
#> [79,] 53 9.970e-01 2.454e-04
#> [80,] 53 9.973e-01 2.236e-04
#> [81,] 53 9.975e-01 2.038e-04
#> [82,] 54 9.977e-01 1.857e-04
#> [83,] 53 9.979e-01 1.692e-04
#> [84,] 53 9.981e-01 1.541e-04
#> [85,] 53 9.983e-01 1.404e-04
#> [86,] 53 9.984e-01 1.280e-04
#> [87,] 53 9.986e-01 1.166e-04
#> [88,] 53 9.987e-01 1.062e-04
#> [89,] 53 9.988e-01 9.681e-05
#> [90,] 53 9.989e-01 8.821e-05
#> [91,] 53 9.990e-01 8.037e-05
#> [1] "done bootstrap 1"

c_selectID <- 2
genes = DEgenes$DE_Subpop2vsRemaining$id[1:200]
genes <- gsub("_.*", "", genes)
LSOLDA_dat2 <- bootstrap_scGPS(nboots = 1, mixedpop1 = mixedpop2, mixedpop2 = mixedpop2, genes=genes, c_
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>

```

```

#>      Df      %Dev      Lambda
#> [1,]  0 -2.403e-15 2.500e-01
#> [2,]  2  3.503e-02 2.278e-01
#> [3,]  3  7.178e-02 2.075e-01
#> [4,]  4  1.070e-01 1.891e-01
#> [5,]  5  1.403e-01 1.723e-01
#> [6,]  6  1.704e-01 1.570e-01
#> [7,]  7  1.976e-01 1.430e-01
#> [8,]  8  2.217e-01 1.303e-01
#> [9,] 10  2.461e-01 1.188e-01
#> [10,] 11  2.682e-01 1.082e-01
#> [11,] 14  2.913e-01 9.859e-02
#> [12,] 15  3.147e-01 8.983e-02
#> [13,] 18  3.408e-01 8.185e-02
#> [14,] 21  3.662e-01 7.458e-02
#> [15,] 21  3.900e-01 6.795e-02
#> [16,] 24  4.128e-01 6.192e-02
#> [17,] 26  4.357e-01 5.642e-02
#> [18,] 28  4.571e-01 5.140e-02
#> [19,] 28  4.777e-01 4.684e-02
#> [20,] 29  4.969e-01 4.268e-02
#> [21,] 31  5.146e-01 3.889e-02
#> [22,] 30  5.309e-01 3.543e-02
#> [23,] 30  5.451e-01 3.228e-02
#> [24,] 34  5.596e-01 2.942e-02
#> [25,] 37  5.776e-01 2.680e-02
#> [26,] 44  5.976e-01 2.442e-02
#> [27,] 47  6.171e-01 2.225e-02
#> [28,] 47  6.369e-01 2.028e-02
#> [29,] 51  6.575e-01 1.847e-02
#> [30,] 53  6.772e-01 1.683e-02
#> [31,] 56  6.962e-01 1.534e-02
#> [32,] 58  7.152e-01 1.397e-02
#> [33,] 60  7.337e-01 1.273e-02
#> [34,] 64  7.524e-01 1.160e-02
#> [35,] 66  7.705e-01 1.057e-02
#> [36,] 66  7.875e-01 9.632e-03
#> [37,] 69  8.032e-01 8.777e-03
#> [38,] 72  8.183e-01 7.997e-03
#> [39,] 74  8.325e-01 7.287e-03
#> [40,] 74  8.459e-01 6.639e-03
#> [41,] 75  8.582e-01 6.049e-03
#> [42,] 79  8.698e-01 5.512e-03
#> [43,] 78  8.807e-01 5.022e-03
#> [44,] 79  8.906e-01 4.576e-03
#> [45,] 81  8.999e-01 4.170e-03
#> [46,] 80  9.085e-01 3.799e-03
#> [47,] 80  9.165e-01 3.462e-03
#> [48,] 84  9.238e-01 3.154e-03
#> [49,] 83  9.307e-01 2.874e-03
#> [50,] 85  9.370e-01 2.619e-03
#> [51,] 87  9.427e-01 2.386e-03
#> [52,] 86  9.480e-01 2.174e-03

```

```

#> [53,] 87 9.527e-01 1.981e-03
#> [54,] 87 9.570e-01 1.805e-03
#> [55,] 87 9.610e-01 1.645e-03
#> [56,] 88 9.645e-01 1.498e-03
#> [57,] 89 9.677e-01 1.365e-03
#> [58,] 89 9.707e-01 1.244e-03
#> [59,] 90 9.733e-01 1.134e-03
#> [60,] 90 9.757e-01 1.033e-03
#> [61,] 91 9.779e-01 9.411e-04
#> [62,] 91 9.799e-01 8.575e-04
#> [63,] 92 9.817e-01 7.813e-04
#> [64,] 92 9.833e-01 7.119e-04
#> [65,] 92 9.848e-01 6.487e-04
#> [66,] 92 9.862e-01 5.910e-04
#> [67,] 92 9.874e-01 5.385e-04
#> [68,] 92 9.885e-01 4.907e-04
#> [69,] 91 9.895e-01 4.471e-04
#> [70,] 91 9.905e-01 4.074e-04
#> [71,] 91 9.913e-01 3.712e-04
#> [72,] 91 9.921e-01 3.382e-04
#> [73,] 90 9.928e-01 3.082e-04
#> [74,] 90 9.934e-01 2.808e-04
#> [75,] 90 9.940e-01 2.558e-04
#> [76,] 90 9.945e-01 2.331e-04
#> [77,] 90 9.950e-01 2.124e-04
#> [78,] 90 9.955e-01 1.935e-04
#> [79,] 90 9.959e-01 1.763e-04
#> [80,] 91 9.962e-01 1.607e-04
#> [81,] 91 9.966e-01 1.464e-04
#> [82,] 90 9.969e-01 1.334e-04
#> [83,] 90 9.971e-01 1.215e-04
#> [84,] 91 9.974e-01 1.107e-04
#> [85,] 91 9.976e-01 1.009e-04
#> [86,] 92 9.978e-01 9.195e-05
#> [87,] 92 9.980e-01 8.378e-05
#> [88,] 92 9.982e-01 7.633e-05
#> [89,] 92 9.984e-01 6.955e-05
#> [90,] 92 9.985e-01 6.337e-05
#> [91,] 92 9.986e-01 5.774e-05
#> [92,] 91 9.988e-01 5.261e-05
#> [93,] 91 9.989e-01 4.794e-05
#> [94,] 90 9.990e-01 4.368e-05
#> [95,] 90 9.991e-01 3.980e-05
#> [1] "done bootstrap 1"

c_selectID <- 3
genes = DEgenes$DE_Subpop3vsRemaining$id[1:200]
genes <- gsub("_.*", "", genes)
LSOLDA_dat3 <- bootstrap_scGPS(nboots = 1, mixedpop1 = mixedpop2, mixedpop2 = mixedpop2, genes=genes, c_
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>

```

```

#>      Df    %Dev   Lambda
#> [1,]  0 0.0000 0.456600
#> [2,]  1 0.0535 0.435900
#> [3,]  1 0.1026 0.416100
#> [4,]  1 0.1478 0.397200
#> [5,]  1 0.1896 0.379100
#> [6,]  1 0.2285 0.361900
#> [7,]  1 0.2649 0.345400
#> [8,]  1 0.2988 0.329700
#> [9,]  1 0.3308 0.314700
#> [10,] 1 0.3608 0.300400
#> [11,] 1 0.3891 0.286800
#> [12,] 1 0.4159 0.273700
#> [13,] 1 0.4412 0.261300
#> [14,] 1 0.4652 0.249400
#> [15,] 1 0.4880 0.238100
#> [16,] 1 0.5097 0.227300
#> [17,] 1 0.5303 0.216900
#> [18,] 1 0.5499 0.207100
#> [19,] 1 0.5685 0.197700
#> [20,] 1 0.5863 0.188700
#> [21,] 1 0.6033 0.180100
#> [22,] 1 0.6195 0.171900
#> [23,] 1 0.6350 0.164100
#> [24,] 1 0.6498 0.156600
#> [25,] 1 0.6639 0.149500
#> [26,] 1 0.6774 0.142700
#> [27,] 1 0.6903 0.136200
#> [28,] 1 0.7027 0.130100
#> [29,] 1 0.7145 0.124100
#> [30,] 1 0.7258 0.118500
#> [31,] 1 0.7367 0.113100
#> [32,] 1 0.7470 0.108000
#> [33,] 2 0.7573 0.103100
#> [34,] 2 0.7676 0.098380
#> [35,] 2 0.7773 0.093910
#> [36,] 2 0.7866 0.089640
#> [37,] 3 0.7959 0.085570
#> [38,] 3 0.8049 0.081680
#> [39,] 3 0.8135 0.077960
#> [40,] 3 0.8217 0.074420
#> [41,] 3 0.8296 0.071040
#> [42,] 3 0.8371 0.067810
#> [43,] 4 0.8443 0.064730
#> [44,] 4 0.8512 0.061790
#> [45,] 4 0.8578 0.058980
#> [46,] 4 0.8642 0.056300
#> [47,] 4 0.8702 0.053740
#> [48,] 4 0.8759 0.051300
#> [49,] 4 0.8814 0.048960
#> [50,] 4 0.8867 0.046740
#> [51,] 5 0.8919 0.044610
#> [52,] 5 0.8968 0.042590

```

```

#> [53,] 5 0.9016 0.040650
#> [54,] 5 0.9061 0.038800
#> [55,] 5 0.9104 0.037040
#> [56,] 5 0.9145 0.035360
#> [57,] 5 0.9184 0.033750
#> [58,] 5 0.9221 0.032210
#> [59,] 5 0.9257 0.030750
#> [60,] 6 0.9291 0.029350
#> [61,] 6 0.9323 0.028020
#> [62,] 7 0.9354 0.026750
#> [63,] 7 0.9383 0.025530
#> [64,] 7 0.9412 0.024370
#> [65,] 7 0.9438 0.023260
#> [66,] 7 0.9464 0.022200
#> [67,] 7 0.9489 0.021200
#> [68,] 7 0.9512 0.020230
#> [69,] 7 0.9534 0.019310
#> [70,] 7 0.9555 0.018430
#> [71,] 7 0.9575 0.017600
#> [72,] 8 0.9595 0.016800
#> [73,] 8 0.9613 0.016030
#> [74,] 8 0.9631 0.015300
#> [75,] 8 0.9648 0.014610
#> [76,] 8 0.9664 0.013950
#> [77,] 8 0.9679 0.013310
#> [78,] 8 0.9693 0.012710
#> [79,] 8 0.9707 0.012130
#> [80,] 8 0.9721 0.011580
#> [81,] 8 0.9733 0.011050
#> [82,] 8 0.9745 0.010550
#> [83,] 8 0.9757 0.010070
#> [84,] 8 0.9768 0.009612
#> [85,] 8 0.9778 0.009175
#> [86,] 8 0.9788 0.008758
#> [87,] 8 0.9798 0.008360
#> [88,] 8 0.9807 0.007980
#> [89,] 8 0.9816 0.007617
#> [90,] 8 0.9824 0.007271
#> [91,] 8 0.9832 0.006940
#> [92,] 8 0.9840 0.006625
#> [93,] 8 0.9847 0.006324
#> [94,] 8 0.9854 0.006036
#> [95,] 8 0.9860 0.005762
#> [96,] 8 0.9867 0.005500
#> [97,] 8 0.9873 0.005250
#> [98,] 8 0.9878 0.005012
#> [99,] 8 0.9884 0.004784
#> [100,] 8 0.9889 0.004566
#> [1] "done bootstrap 1"

#prepare table input for sankey plot

reformat_LASSO <-function(c_selectID = NULL, s_selectID = NULL, LSOLDA_dat = NULL,

```



```

        nPredSubpop = row_cluster, Nodes_group = "#7570b3"){
LASSO_out <- summary_prediction_lasso(LSOLDA_dat=LSOLDA_dat, nPredSubpop = nPredSubpop)
LASSO_out <- as.data.frame(LASSO_out)
temp_name <- gsub("LASSO for subpop", "C", LASSO_out$names)
temp_name <- gsub(" in target mixedpop", "S", temp_name)
LASSO_out$names <- temp_name
source <- rep(paste0("C",c_selectID,"S",s_selectID), length(temp_name))
LASSO_out$Source <- source
LASSO_out$Node <- source
LASSO_out$Nodes_group <- rep(Nodes_group, length(temp_name))
colnames(LASSO_out) <- c("Value", "Target", "Source", "Node", "NodeGroup")
LASSO_out$Value <- as.numeric(as.vector(LASSO_out$Value))
return(LASSO_out)
}

LASSO_C1S2 <- reformat_LASSO(c_selectID=1, s_selectID =2, LSOLDA_dat=LSOLDA_dat1,
        nPredSubpop = row_cluster, Nodes_group = "#7570b3")

LASSO_C2S2 <- reformat_LASSO(c_selectID=2, s_selectID =2, LSOLDA_dat=LSOLDA_dat2,
        nPredSubpop = row_cluster, Nodes_group = "#1b9e77")

LASSO_C3S2 <- reformat_LASSO(c_selectID=3, s_selectID =2, LSOLDA_dat=LSOLDA_dat3,
        nPredSubpop = row_cluster, Nodes_group = "#e7298a")

combined <- rbind(LASSO_C1S2,LASSO_C2S2,LASSO_C3S2 )
combined <- combined[is.na(combined$Value) != TRUE,]
combined_D3obj <- list(Nodes=combined[,4:5], Links=combined[,c(3,2,1)])

library(networkD3)

Node_source <- as.vector(sort(unique(combined_D3obj$Links$Source)))
Node_target <- as.vector(sort(unique(combined_D3obj$Links$Target)))
Node_all <- unique(c(Node_source, Node_target))

#assign IDs for Source (start from 0)
Source <- combined_D3obj$Links$Source
Target <- combined_D3obj$Links$Target

for(i in 1:length(Node_all)){
  Source[Source==Node_all[i]] <- i-1
  Target[Target==Node_all[i]] <- i-1
}

combined_D3obj$Links$Source <- as.numeric(Source)
combined_D3obj$Links$Target <- as.numeric(Target)
combined_D3obj$Links$LinkColor <- combined$NodeGroup

#prepare node info
node_df <- data.frame(Node=Node_all)
node_df$id <- as.numeric(c(0, 1:(length(Node_all)-1)))

suppressMessages(library(dplyr))

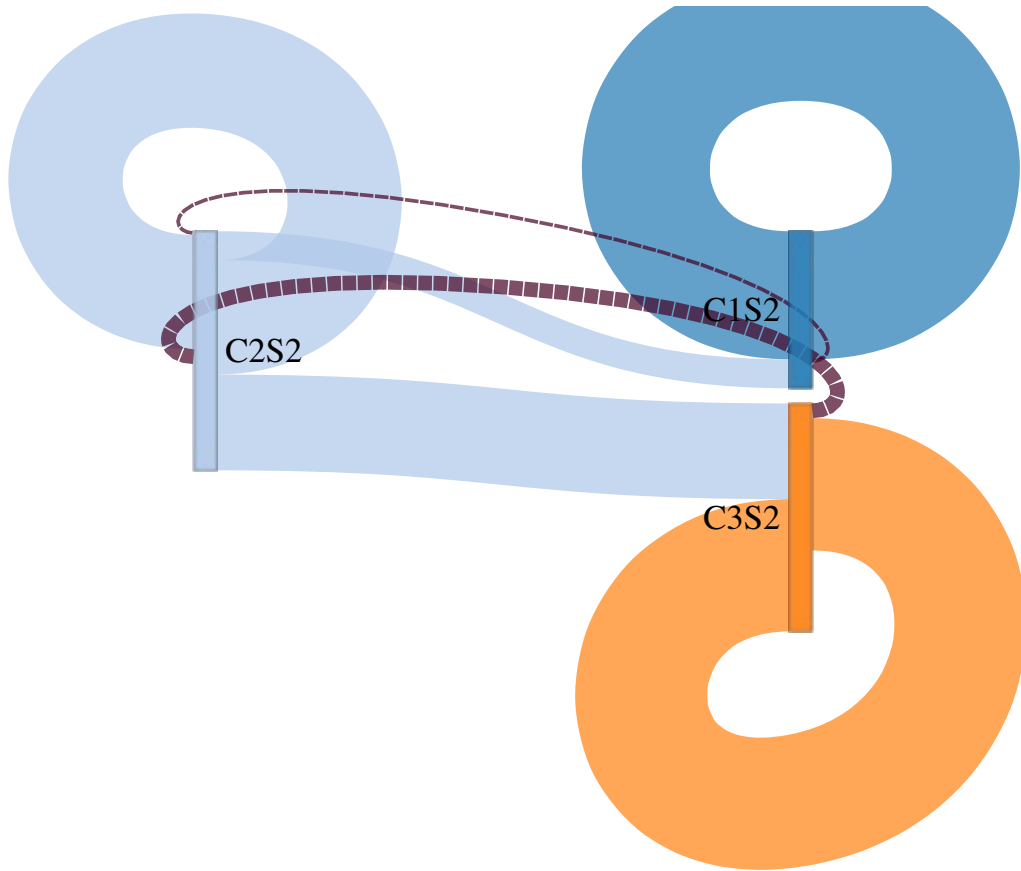
```

```

Color <- combined %>% count(Node, color=NodeGroup) %>% select(2)
node_df$color <- Color$color

suppressMessages(library(networkD3))
p1<-sankeyNetwork(Links =combined_D3obj$Links, Nodes = node_df, Value = "Value", NodeGroup ="color", L
                fontSize = 22 )
p1

```



```

#saveNetwork(p1, file = paste0(path, 'Subpopulation_Net.html'))
##R Setting Information
#sessionInfo()
#rmarkdown::render("/Users/quan.nguyen/Documents/Powell_group_MacQuan/AllCodes/scGPS/vignettes/vignette
#rmarkdown::render("/Users/quan.nguyen/Documents/Powell_group_MacQuan/AllCodes/scGPS/vignettes/vignette

```