# scGPS introduction

*Quan Nguyen*

*2018-04-20*

## Contents



## 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: model fit failed for Fold01.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 22 25 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold03.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 50 81 appear to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 16 85 appear to be constant within groups
#> Warning: model fit failed for Fold09.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 10 58 appear to be constant within groups
#> Warning: model fit failed for Fold10.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
```

```
#>   variable 94 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 10 85 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold03.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 81 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 22 25 appear to be constant within groups
#> Warning: model fit failed for Fold05.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 94 appears to be constant within groups
#> Warning: model fit failed for Fold06.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 58 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold09.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 59 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 16 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 25 appears to be constant within groups
#> Warning: model fit failed for Fold03.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 10 85 appear to be constant within groups
#> Warning: model fit failed for Fold05.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 58 94 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold08.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 16 22 appear to be constant within groups
#> Warning: model fit failed for Fold10.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 81 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 2.472e-01
#>   [2,]  2  3.521e-02 2.252e-01
#>   [3,]  2  7.013e-02 2.052e-01
#>   [4,]  2  9.999e-02 1.870e-01
#>   [5,]  5  1.330e-01 1.704e-01
#>   [6,]  5  1.679e-01 1.552e-01
#>   [7,]  5  1.982e-01 1.414e-01
#>   [8,]  5  2.247e-01 1.289e-01
#>   [9,]  5  2.479e-01 1.174e-01
#>  [10,]  7  2.704e-01 1.070e-01
#>  [11,]  7  2.924e-01 9.748e-02
#>  [12,]  7  3.119e-01 8.882e-02
#>  [13,]  9  3.301e-01 8.093e-02
#>  [14,]  9  3.474e-01 7.374e-02
#>  [15,] 10  3.631e-01 6.719e-02
#>  [16,] 12  3.791e-01 6.122e-02
#>  [17,] 14  3.966e-01 5.578e-02
#>  [18,] 16  4.159e-01 5.083e-02
#>  [19,] 20  4.350e-01 4.631e-02
```

```
#>  [20,] 20  4.541e-01 4.220e-02
#>  [21,] 24  4.732e-01 3.845e-02
#>  [22,] 27  4.920e-01 3.503e-02
#>  [23,] 30  5.110e-01 3.192e-02
#>  [24,] 33  5.295e-01 2.909e-02
#>  [25,] 35  5.491e-01 2.650e-02
#>  [26,] 35  5.678e-01 2.415e-02
#>  [27,] 36  5.848e-01 2.200e-02
#>  [28,] 39  6.007e-01 2.005e-02
#>  [29,] 42  6.166e-01 1.827e-02
#>  [30,] 43  6.329e-01 1.664e-02
#>  [31,] 46  6.487e-01 1.517e-02
#>  [32,] 49  6.640e-01 1.382e-02
#>  [33,] 49  6.782e-01 1.259e-02
#>  [34,] 51  6.914e-01 1.147e-02
#>  [35,] 52  7.043e-01 1.045e-02
#>  [36,] 56  7.164e-01 9.524e-03
#>  [37,] 59  7.292e-01 8.678e-03
#>  [38,] 59  7.418e-01 7.907e-03
#>  [39,] 62  7.542e-01 7.205e-03
#>  [40,] 63  7.663e-01 6.565e-03
#>  [41,] 62  7.778e-01 5.982e-03
#>  [42,] 62  7.888e-01 5.450e-03
#>  [43,] 64  7.993e-01 4.966e-03
#>  [44,] 65  8.098e-01 4.525e-03
#>  [45,] 65  8.200e-01 4.123e-03
#>  [46,] 65  8.298e-01 3.757e-03
#>  [47,] 67  8.399e-01 3.423e-03
#>  [48,] 67  8.501e-01 3.119e-03
#>  [49,] 68  8.599e-01 2.842e-03
#>  [50,] 70  8.696e-01 2.589e-03
#>  [51,] 71  8.791e-01 2.359e-03
#>  [52,] 70  8.884e-01 2.150e-03
#>  [53,] 70  8.971e-01 1.959e-03
#>  [54,] 71  9.052e-01 1.785e-03
#>  [55,] 71  9.129e-01 1.626e-03
#>  [56,] 72  9.201e-01 1.482e-03
#>  [57,] 72  9.268e-01 1.350e-03
#>  [58,] 72  9.331e-01 1.230e-03
#>  [59,] 73  9.389e-01 1.121e-03
#>  [60,] 73  9.443e-01 1.021e-03
#>  [61,] 73  9.491e-01 9.305e-04
#>  [62,] 73  9.537e-01 8.479e-04
#>  [63,] 72  9.579e-01 7.726e-04
#>  [64,] 72  9.617e-01 7.039e-04
#>  [65,] 72  9.651e-01 6.414e-04
#>  [66,] 73  9.683e-01 5.844e-04
#>  [67,] 73  9.711e-01 5.325e-04
#>  [68,] 74  9.737e-01 4.852e-04
#>  [69,] 75  9.761e-01 4.421e-04
#>  [70,] 75  9.783e-01 4.028e-04
#>  [71,] 75  9.802e-01 3.670e-04
#>  [72,] 75  9.820e-01 3.344e-04
```

```
#>  [73,] 75  9.836e-01 3.047e-04
#>  [74,] 75  9.851e-01 2.776e-04
#>  [75,] 75  9.864e-01 2.530e-04
#>  [76,] 75  9.876e-01 2.305e-04
#>  [77,] 75  9.887e-01 2.100e-04
#>  [78,] 75  9.897e-01 1.914e-04
#>  [79,] 75  9.907e-01 1.744e-04
#>  [80,] 75  9.915e-01 1.589e-04
#>  [81,] 75  9.923e-01 1.448e-04
#>  [82,] 75  9.929e-01 1.319e-04
#>  [83,] 75  9.936e-01 1.202e-04
#>  [84,] 75  9.941e-01 1.095e-04
#>  [85,] 75  9.947e-01 9.978e-05
#>  [86,] 75  9.951e-01 9.091e-05
#>  [87,] 75  9.956e-01 8.284e-05
#>  [88,] 75  9.959e-01 7.548e-05
#>  [89,] 75  9.963e-01 6.877e-05
#>  [90,] 76  9.966e-01 6.266e-05
#>  [91,] 76  9.969e-01 5.710e-05
#>  [92,] 76  9.972e-01 5.202e-05
#>  [93,] 76  9.975e-01 4.740e-05
#>  [94,] 76  9.977e-01 4.319e-05
#>  [95,] 76  9.979e-01 3.935e-05
#>  [96,] 76  9.981e-01 3.586e-05
#>  [97,] 76  9.982e-01 3.267e-05
#>  [98,] 76  9.984e-01 2.977e-05
#>  [99,] 76  9.985e-01 2.713e-05
#> [100,] 76  9.987e-01 2.472e-05
#> [1] "done bootstrap 1"
#> Warning: model fit failed for Fold01.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 85 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 Fold04.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 11 93 appear to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  5 78 appear to be constant within groups
#> Warning: model fit failed for Fold06.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 13 appears to be constant within groups
#> Warning: model fit failed for Fold07.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 83 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  7 47 79 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold10.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 55 71 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 Fold03.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 55 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
```

```
#>   variable 71 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 47 79 appear to be constant within groups
#> Warning: model fit failed for Fold06.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  7 13 appear to be constant within groups
#> Warning: model fit failed for Fold07.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 85 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 83 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  5 93 appear to be constant within groups
#> Warning: model fit failed for Fold10.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 11 78 appear to be constant within groups
#> Warning: model fit failed for Fold01.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 78 83 85 appear to be constant within groups
#> Warning: model fit failed for Fold02.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  7 13 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 93 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 47 79 appear 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, ...) :
#>   variable 5 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold08.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 11 55 appear to be constant within groups
#> Warning: model fit failed for Fold09.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 71 appears to be constant within groups
#> 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 2.405e-01
#>   [2,]  1  2.848e-02 2.191e-01
#>   [3,]  1  5.284e-02 1.997e-01
#>   [4,]  2  7.650e-02 1.819e-01
#>   [5,]  2  1.035e-01 1.658e-01
#>   [6,]  4  1.340e-01 1.510e-01
#>   [7,]  4  1.629e-01 1.376e-01
#>   [8,]  6  1.893e-01 1.254e-01
#>   [9,]  6  2.132e-01 1.143e-01
#>  [10,]  9  2.369e-01 1.041e-01
#>  [11,] 11  2.594e-01 9.485e-02
#>  [12,] 10  2.834e-01 8.643e-02
#>  [13,] 11  3.062e-01 7.875e-02
#>  [14,] 12  3.274e-01 7.175e-02
#>  [15,] 12  3.467e-01 6.538e-02
#>  [16,] 14  3.651e-01 5.957e-02
```

```
#>  [17,] 16  3.840e-01 5.428e-02
#>  [18,] 18  4.044e-01 4.946e-02
#>  [19,] 20  4.260e-01 4.506e-02
#>  [20,] 22  4.461e-01 4.106e-02
#>  [21,] 25  4.681e-01 3.741e-02
#>  [22,] 26  4.886e-01 3.409e-02
#>  [23,] 28  5.082e-01 3.106e-02
#>  [24,] 31  5.266e-01 2.830e-02
#>  [25,] 32  5.443e-01 2.579e-02
#>  [26,] 36  5.624e-01 2.350e-02
#>  [27,] 37  5.798e-01 2.141e-02
#>  [28,] 40  5.966e-01 1.951e-02
#>  [29,] 40  6.128e-01 1.777e-02
#>  [30,] 43  6.278e-01 1.619e-02
#>  [31,] 46  6.423e-01 1.476e-02
#>  [32,] 50  6.565e-01 1.345e-02
#>  [33,] 52  6.714e-01 1.225e-02
#>  [34,] 55  6.860e-01 1.116e-02
#>  [35,] 57  6.999e-01 1.017e-02
#>  [36,] 59  7.130e-01 9.267e-03
#>  [37,] 63  7.277e-01 8.444e-03
#>  [38,] 64  7.428e-01 7.694e-03
#>  [39,] 66  7.575e-01 7.010e-03
#>  [40,] 65  7.720e-01 6.388e-03
#>  [41,] 66  7.860e-01 5.820e-03
#>  [42,] 67  7.995e-01 5.303e-03
#>  [43,] 67  8.123e-01 4.832e-03
#>  [44,] 71  8.247e-01 4.403e-03
#>  [45,] 74  8.367e-01 4.012e-03
#>  [46,] 73  8.481e-01 3.655e-03
#>  [47,] 73  8.586e-01 3.330e-03
#>  [48,] 73  8.686e-01 3.035e-03
#>  [49,] 72  8.781e-01 2.765e-03
#>  [50,] 72  8.867e-01 2.519e-03
#>  [51,] 72  8.947e-01 2.296e-03
#>  [52,] 73  9.023e-01 2.092e-03
#>  [53,] 74  9.096e-01 1.906e-03
#>  [54,] 74  9.165e-01 1.737e-03
#>  [55,] 74  9.229e-01 1.582e-03
#>  [56,] 74  9.288e-01 1.442e-03
#>  [57,] 75  9.343e-01 1.314e-03
#>  [58,] 74  9.398e-01 1.197e-03
#>  [59,] 76  9.448e-01 1.091e-03
#>  [60,] 76  9.496e-01 9.937e-04
#>  [61,] 76  9.539e-01 9.054e-04
#>  [62,] 76  9.580e-01 8.250e-04
#>  [63,] 76  9.618e-01 7.517e-04
#>  [64,] 76  9.652e-01 6.849e-04
#>  [65,] 76  9.683e-01 6.241e-04
#>  [66,] 77  9.711e-01 5.686e-04
#>  [67,] 77  9.737e-01 5.181e-04
#>  [68,] 76  9.760e-01 4.721e-04
#>  [69,] 76  9.781e-01 4.301e-04
```

```
#>  [70,] 76  9.801e-01 3.919e-04
#>  [71,] 76  9.818e-01 3.571e-04
#>  [72,] 76  9.834e-01 3.254e-04
#>  [73,] 76  9.849e-01 2.965e-04
#>  [74,] 76  9.862e-01 2.701e-04
#>  [75,] 76  9.875e-01 2.461e-04
#>  [76,] 76  9.886e-01 2.243e-04
#>  [77,] 75  9.896e-01 2.044e-04
#>  [78,] 76  9.905e-01 1.862e-04
#>  [79,] 76  9.913e-01 1.697e-04
#>  [80,] 76  9.921e-01 1.546e-04
#>  [81,] 76  9.928e-01 1.409e-04
#>  [82,] 76  9.934e-01 1.283e-04
#>  [83,] 76  9.940e-01 1.169e-04
#>  [84,] 76  9.945e-01 1.066e-04
#>  [85,] 76  9.950e-01 9.709e-05
#>  [86,] 76  9.955e-01 8.846e-05
#>  [87,] 76  9.958e-01 8.060e-05
#>  [88,] 77  9.962e-01 7.344e-05
#>  [89,] 77  9.965e-01 6.692e-05
#>  [90,] 77  9.968e-01 6.097e-05
#>  [91,] 77  9.971e-01 5.556e-05
#>  [92,] 77  9.974e-01 5.062e-05
#>  [93,] 77  9.976e-01 4.612e-05
#>  [94,] 77  9.978e-01 4.203e-05
#>  [95,] 77  9.980e-01 3.829e-05
#>  [96,] 77  9.982e-01 3.489e-05
#>  [97,] 77  9.983e-01 3.179e-05
#>  [98,] 77  9.985e-01 2.897e-05
#>  [99,] 77  9.986e-01 2.639e-05
#> [100,] 77  9.987e-01 2.405e-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: model fit failed for Fold02.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  4 16 60 70 appear to be constant within groups
#> Warning: model fit failed for Fold03.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 1 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 15 83 appear to be constant within groups
#> Warning: model fit failed for Fold06.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 81 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 54 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 29 appears to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold04.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
```

```
#>   variable 81 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 29 60 appear to be constant within groups
#> Warning: model fit failed for Fold07.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  4 15 54 appear to be constant within groups
#> Warning: model fit failed for Fold08.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 70 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 16 83 appear to be constant within groups
#> Warning: model fit failed for Fold01.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 29 appears to be constant within groups
#> Warning: model fit failed for Fold02.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 54 appears to be constant within groups
#> Warning: model fit failed for Fold03.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 60 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 70 appears to be constant within groups
#> Warning: model fit failed for Fold05.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  4 81 appear to be constant within groups
#> Warning: model fit failed for Fold06.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 83 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 15 16 appear 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 2.953e-01
#>  [2,]  1  4.286e-02 2.690e-01
#>  [3,]  2  9.037e-02 2.451e-01
#>  [4,]  2  1.330e-01 2.234e-01
#>  [5,]  4  1.770e-01 2.035e-01
#>  [6,]  4  2.222e-01 1.854e-01
#>  [7,]  5  2.639e-01 1.690e-01
#>  [8,]  5  3.020e-01 1.540e-01
#>  [9,]  5  3.359e-01 1.403e-01
#> [10,]  6  3.665e-01 1.278e-01
#> [11,]  6  3.955e-01 1.165e-01
#> [12,]  7  4.223e-01 1.061e-01
#> [13,]  7  4.468e-01 9.669e-02
#> [14,]  7  4.690e-01 8.810e-02
#> [15,]  7  4.889e-01 8.027e-02
#> [16,]  7  5.070e-01 7.314e-02
#> [17,]  9  5.237e-01 6.664e-02
#> [18,]  9  5.416e-01 6.072e-02
#> [19,]  9  5.578e-01 5.533e-02
#> [20,] 10  5.730e-01 5.041e-02
#> [21,] 11  5.873e-01 4.593e-02
#> [22,] 13  6.008e-01 4.185e-02
#> [23,] 14  6.176e-01 3.814e-02
#> [24,] 14  6.328e-01 3.475e-02
```

```
#> [25,] 15  6.465e-01 3.166e-02
#> [26,] 19  6.603e-01 2.885e-02
#> [27,] 20  6.742e-01 2.629e-02
#> [28,] 21  6.872e-01 2.395e-02
#> [29,] 23  7.016e-01 2.182e-02
#> [30,] 24  7.157e-01 1.988e-02
#> [31,] 25  7.285e-01 1.812e-02
#> [32,] 26  7.401e-01 1.651e-02
#> [33,] 28  7.513e-01 1.504e-02
#> [34,] 31  7.620e-01 1.371e-02
#> [35,] 34  7.736e-01 1.249e-02
#> [36,] 38  7.860e-01 1.138e-02
#> [37,] 38  7.982e-01 1.037e-02
#> [38,] 41  8.102e-01 9.447e-03
#> [39,] 43  8.220e-01 8.607e-03
#> [40,] 45  8.334e-01 7.843e-03
#> [41,] 46  8.442e-01 7.146e-03
#> [42,] 48  8.544e-01 6.511e-03
#> [43,] 49  8.641e-01 5.933e-03
#> [44,] 51  8.735e-01 5.406e-03
#> [45,] 53  8.828e-01 4.925e-03
#> [46,] 53  8.916e-01 4.488e-03
#> [47,] 54  9.000e-01 4.089e-03
#> [48,] 54  9.078e-01 3.726e-03
#> [49,] 56  9.151e-01 3.395e-03
#> [50,] 56  9.221e-01 3.093e-03
#> [51,] 58  9.288e-01 2.819e-03
#> [52,] 57  9.349e-01 2.568e-03
#> [53,] 57  9.405e-01 2.340e-03
#> [54,] 57  9.457e-01 2.132e-03
#> [55,] 57  9.504e-01 1.943e-03
#> [56,] 57  9.548e-01 1.770e-03
#> [57,] 57  9.587e-01 1.613e-03
#> [58,] 57  9.624e-01 1.470e-03
#> [59,] 57  9.658e-01 1.339e-03
#> [60,] 57  9.687e-01 1.220e-03
#> [61,] 57  9.715e-01 1.112e-03
#> [62,] 59  9.741e-01 1.013e-03
#> [63,] 59  9.764e-01 9.229e-04
#> [64,] 60  9.784e-01 8.409e-04
#> [65,] 60  9.803e-01 7.662e-04
#> [66,] 60  9.821e-01 6.982e-04
#> [67,] 60  9.837e-01 6.361e-04
#> [68,] 59  9.851e-01 5.796e-04
#> [69,] 59  9.865e-01 5.281e-04
#> [70,] 59  9.877e-01 4.812e-04
#> [71,] 59  9.888e-01 4.385e-04
#> [72,] 59  9.897e-01 3.995e-04
#> [73,] 59  9.907e-01 3.640e-04
#> [74,] 59  9.915e-01 3.317e-04
#> [75,] 59  9.922e-01 3.022e-04
#> [76,] 59  9.929e-01 2.754e-04
#> [77,] 59  9.935e-01 2.509e-04
```

```
#> [78,] 58  9.941e-01 2.286e-04
#> [79,] 58  9.946e-01 2.083e-04
#> [80,] 58  9.951e-01 1.898e-04
#> [81,] 58  9.955e-01 1.729e-04
#> [82,] 58  9.959e-01 1.576e-04
#> [83,] 58  9.963e-01 1.436e-04
#> [84,] 58  9.966e-01 1.308e-04
#> [85,] 58  9.969e-01 1.192e-04
#> [86,] 58  9.972e-01 1.086e-04
#> [87,] 57  9.974e-01 9.896e-05
#> [88,] 57  9.976e-01 9.017e-05
#> [89,] 57  9.979e-01 8.216e-05
#> [90,] 57  9.980e-01 7.486e-05
#> [91,] 57  9.982e-01 6.821e-05
#> [92,] 57  9.984e-01 6.215e-05
#> [93,] 58  9.985e-01 5.663e-05
#> [94,] 58  9.986e-01 5.160e-05
#> [95,] 58  9.988e-01 4.702e-05
#> [96,] 58  9.989e-01 4.284e-05
#> [97,] 58  9.990e-01 3.903e-05
#> [98,] 58  9.991e-01 3.557e-05
#> [1] "done bootstrap 1"
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold02.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 25 81 appear to be constant within groups
#> 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 11 46 57 appear to be constant within groups
#> Warning: model fit failed for Fold05.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 17 74 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 lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold09.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  1 49 82 appear to be constant within groups
#> Warning: model fit failed for Fold10.Rep1: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 64 appears to be constant within groups
#> Warning: model fit failed for Fold01.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 11 81 appear to be constant within groups
#> Warning in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold03.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 25 appears to be constant within groups
#> Warning: model fit failed for Fold04.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  1 49 82 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.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 57 appears to be constant within groups
#> Warning: model fit failed for Fold08.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 17 64 appear to be constant within groups
```

```
#> Warning: model fit failed for Fold09.Rep2: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables 46 74 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 lda.default(x, grouping, ...): variables are collinear

#> 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 17 74 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 in lda.default(x, grouping, ...): variables are collinear
#> Warning: model fit failed for Fold08.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 64 appears to be constant within groups
#> Warning: model fit failed for Fold09.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variable 11 appears to be constant within groups
#> Warning: model fit failed for Fold10.Rep3: parameter=none Error in lda.default(x, grouping, ...) :
#>   variables  1 25 49 82 appear 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 2.317e-01
#>   [2,]  2  3.212e-02 2.111e-01
#>   [3,]  2  6.366e-02 1.924e-01
#>   [4,]  3  9.392e-02 1.753e-01
#>   [5,]  4  1.244e-01 1.597e-01
#>   [6,]  4  1.536e-01 1.455e-01
#>   [7,]  4  1.793e-01 1.326e-01
#>   [8,]  4  2.022e-01 1.208e-01
#>   [9,]  5  2.228e-01 1.101e-01
#>  [10,]  6  2.439e-01 1.003e-01
#>  [11,]  6  2.631e-01 9.138e-02
#>  [12,]  9  2.813e-01 8.326e-02
#>  [13,] 10  3.032e-01 7.587e-02
#>  [14,] 12  3.243e-01 6.913e-02
#>  [15,] 13  3.464e-01 6.299e-02
#>  [16,] 14  3.684e-01 5.739e-02
#>  [17,] 16  3.913e-01 5.229e-02
#>  [18,] 18  4.132e-01 4.765e-02
#>  [19,] 20  4.349e-01 4.341e-02
#>  [20,] 23  4.568e-01 3.956e-02
#>  [21,] 25  4.785e-01 3.604e-02
#>  [22,] 24  4.988e-01 3.284e-02
#>  [23,] 26  5.173e-01 2.992e-02
#>  [24,] 26  5.362e-01 2.727e-02
```

```
#>  [25,] 30  5.553e-01 2.484e-02
#>  [26,] 34  5.748e-01 2.264e-02
#>  [27,] 35  5.940e-01 2.063e-02
#>  [28,] 37  6.122e-01 1.879e-02
#>  [29,] 38  6.298e-01 1.712e-02
#>  [30,] 40  6.464e-01 1.560e-02
#>  [31,] 42  6.619e-01 1.422e-02
#>  [32,] 42  6.766e-01 1.295e-02
#>  [33,] 47  6.909e-01 1.180e-02
#>  [34,] 48  7.052e-01 1.075e-02
#>  [35,] 51  7.192e-01 9.799e-03
#>  [36,] 55  7.344e-01 8.928e-03
#>  [37,] 56  7.491e-01 8.135e-03
#>  [38,] 57  7.627e-01 7.412e-03
#>  [39,] 57  7.758e-01 6.754e-03
#>  [40,] 57  7.880e-01 6.154e-03
#>  [41,] 58  7.992e-01 5.607e-03
#>  [42,] 60  8.099e-01 5.109e-03
#>  [43,] 61  8.201e-01 4.655e-03
#>  [44,] 65  8.300e-01 4.242e-03
#>  [45,] 65  8.394e-01 3.865e-03
#>  [46,] 65  8.482e-01 3.521e-03
#>  [47,] 65  8.563e-01 3.209e-03
#>  [48,] 66  8.641e-01 2.924e-03
#>  [49,] 67  8.717e-01 2.664e-03
#>  [50,] 68  8.789e-01 2.427e-03
#>  [51,] 69  8.856e-01 2.212e-03
#>  [52,] 69  8.922e-01 2.015e-03
#>  [53,] 69  8.986e-01 1.836e-03
#>  [54,] 69  9.048e-01 1.673e-03
#>  [55,] 69  9.111e-01 1.524e-03
#>  [56,] 69  9.171e-01 1.389e-03
#>  [57,] 69  9.229e-01 1.266e-03
#>  [58,] 70  9.284e-01 1.153e-03
#>  [59,] 70  9.339e-01 1.051e-03
#>  [60,] 70  9.391e-01 9.573e-04
#>  [61,] 71  9.441e-01 8.723e-04
#>  [62,] 71  9.488e-01 7.948e-04
#>  [63,] 70  9.532e-01 7.242e-04
#>  [64,] 70  9.573e-01 6.599e-04
#>  [65,] 70  9.611e-01 6.012e-04
#>  [66,] 70  9.645e-01 5.478e-04
#>  [67,] 69  9.677e-01 4.992e-04
#>  [68,] 69  9.705e-01 4.548e-04
#>  [69,] 67  9.732e-01 4.144e-04
#>  [70,] 67  9.755e-01 3.776e-04
#>  [71,] 68  9.777e-01 3.441e-04
#>  [72,] 68  9.796e-01 3.135e-04
#>  [73,] 69  9.814e-01 2.856e-04
#>  [74,] 70  9.831e-01 2.603e-04
#>  [75,] 70  9.846e-01 2.371e-04
#>  [76,] 70  9.860e-01 2.161e-04
#>  [77,] 70  9.872e-01 1.969e-04
```

```
#>  [78,] 70  9.883e-01 1.794e-04
#>  [79,] 70  9.894e-01 1.635e-04
#>  [80,] 70  9.903e-01 1.489e-04
#>  [81,] 70  9.912e-01 1.357e-04
#>  [82,] 70  9.920e-01 1.236e-04
#>  [83,] 70  9.927e-01 1.127e-04
#>  [84,] 69  9.933e-01 1.027e-04
#>  [85,] 69  9.939e-01 9.353e-05
#>  [86,] 68  9.944e-01 8.522e-05
#>  [87,] 67  9.949e-01 7.765e-05
#>  [88,] 67  9.954e-01 7.075e-05
#>  [89,] 67  9.958e-01 6.447e-05
#>  [90,] 67  9.961e-01 5.874e-05
#>  [91,] 67  9.965e-01 5.352e-05
#>  [92,] 67  9.968e-01 4.877e-05
#>  [93,] 67  9.971e-01 4.444e-05
#>  [94,] 67  9.973e-01 4.049e-05
#>  [95,] 67  9.976e-01 3.689e-05
#>  [96,] 67  9.978e-01 3.361e-05
#>  [97,] 67  9.980e-01 3.063e-05
#>  [98,] 68  9.981e-01 2.791e-05
#>  [99,] 68  9.983e-01 2.543e-05
#> [100,] 68  9.984e-01 2.317e-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] 55.08021
#>
#> [[1]][[3]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 80
#>
#> [[1]][[5]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 36.09023
#>
#> [[1]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[1]][[8]]
```

```
#> [1] 55
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LASSO for subpop1 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 72.72727
#>
#> [[2]][[3]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 86.42857
#>
#> [[2]][[5]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 45.11278
#>
#> [[2]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 65
LSOLDA_dat$LDAPredict
#> [[1]]
#> [[1]][[1]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[1]][[2]]
#> [1] 79.14439
#>
#> [[1]][[3]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 42.14286
#>
#> [[1]][[5]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 39.84962
#>
#> [[1]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 50
#>
```

```
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 72.72727
#>
#> [[2]][[3]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 62.85714
#>
#> [[2]][[5]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 51.8797
#>
#> [[2]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 62.5

# summary results LDA
summary_prediction_lda(LSOLDA_dat = LSOLDA_dat, nPredSubpop = 4)
#>                 V1                V2                                    names
#> 1  79.144385026738 72.72727272727272 LDA for subpop 1 in target mixedpop2
#> 2 42.14285714285714 62.85714285714286 LDA for subpop 2 in target mixedpop2
#> 3 39.84962406015038 51.87969924812031 LDA for subpop 3 in target mixedpop2
#> 4                50              62.5 LDA for subpop 4 in target mixedpop2

# summary results Lasso
summary_prediction_lasso(LSOLDA_dat = LSOLDA_dat, nPredSubpop = 4)
#>                 V1                V2                                     names
#> 1 55.08021390374332 72.72727272727272 LASSO for subpop1 in target mixedpop2
#> 2                80 86.42857142857143 LASSO for subpop2 in target mixedpop2
#> 3 36.09022556390977 45.11278195488722 LASSO for subpop3 in target mixedpop2
#> 4                55                65 LASSO for subpop4 in target mixedpop2

# summary deviance
summary_deviance(object = LSOLDA_dat)
#> $allDeviance
#> [1] "0.6465" "0.4132"
#>
#> $DeviMax
#>          Dfd   Deviance        DEgenes
#> 1          0 -2.563e-15 genes_cluster1
#> 2          1    0.04286 genes_cluster1
#> 3          2      0.133 genes_cluster1
#> 4          4     0.2222 genes_cluster1
```

```
#> 5           5     0.3359 genes_cluster1
#> 6           6     0.3955 genes_cluster1
#> 7           7      0.507 genes_cluster1
#> 8           9     0.5578 genes_cluster1
#> 9          10      0.573 genes_cluster1
#> 10         11     0.5873 genes_cluster1
#> 11         13     0.6008 genes_cluster1
#> 12         14     0.6328 genes_cluster1
#> 13         15     0.6465 genes_cluster1
#> 14 remaining          1        DEgenes
#>
#> $LassoGenesMax
#>                                      1                   name
#> (Intercept)              1.037164e+00            (Intercept)
#> CXCR4_ENSG00000121966   -1.627047e-01  CXCR4_ENSG00000121966
#> TTN_ENSG00000155657     -9.701268e-01    TTN_ENSG00000155657
#> FN1_ENSG00000115414     -1.230210e-01    FN1_ENSG00000115414
#> PDGFRA_ENSG00000134853   1.455308e-01 PDGFRA_ENSG00000134853
#> FOXC1_ENSG00000054598    3.140784e-01  FOXC1_ENSG00000054598
#> POU5F1_ENSG00000204531 -2.117037e-02 POU5F1_ENSG00000204531
#> GJA1_ENSG00000152661    -8.864896e-02   GJA1_ENSG00000152661
#> T_ENSG00000164458        2.180129e-01      T_ENSG00000164458
#> SOX17_ENSG00000164736   -3.314781e-02  SOX17_ENSG00000164736
#> HEY1_ENSG00000164683    -2.774349e-01   HEY1_ENSG00000164683
#> MESP1_ENSG00000166823    6.780237e-02  MESP1_ENSG00000166823
#> MESP2_ENSG00000188095    5.720167e-05  MESP2_ENSG00000188095
#> FOXF1_ENSG00000103241    3.195643e-01  FOXF1_ENSG00000103241
#> FOXA2_ENSG00000125798   -4.245519e-01  FOXA2_ENSG00000125798
#> TNNI3_ENSG00000129991    4.055489e-01  TNNI3_ENSG00000129991
```

# 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)*

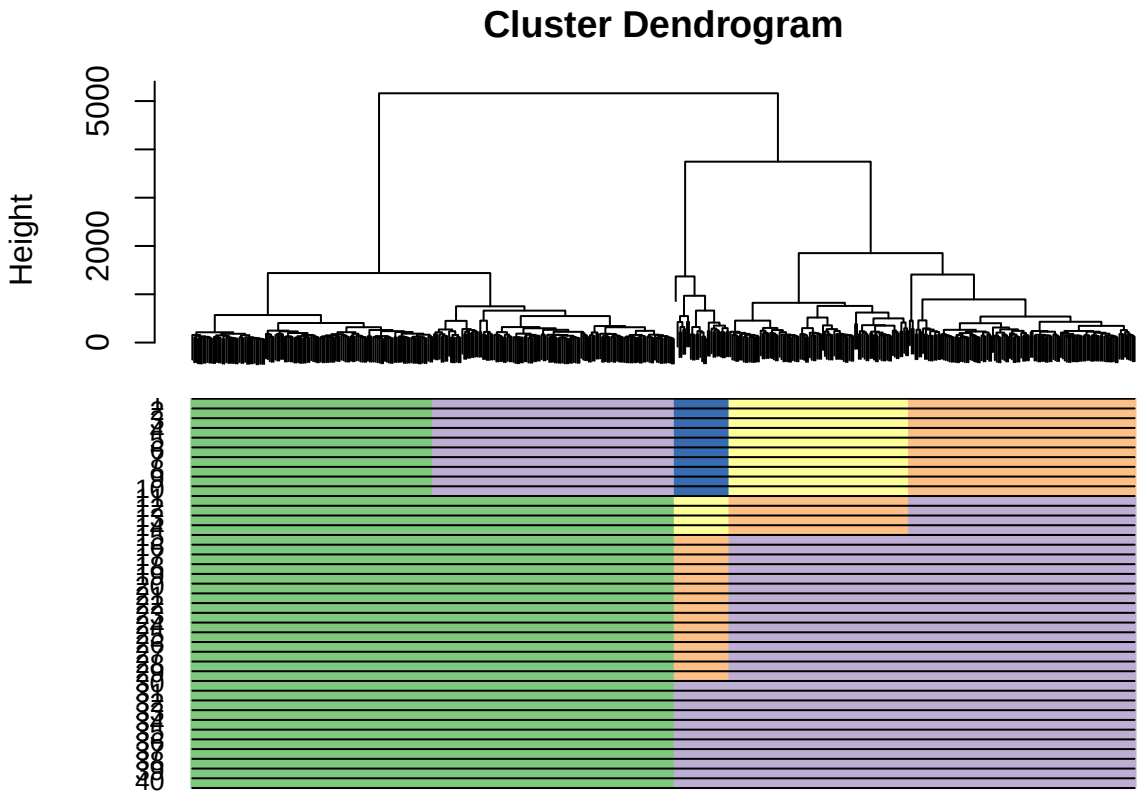
```
#Let's find clustering information in an expresion data
day5 <- sample2
cellnames <- colnames(day5$dat5_counts)
cluster <-day5$dat5_clusters
cellnames <-data.frame("Cluster"=cluster, "cellBarcodes" = cellnames)
mixedpop2 <-NewscGPS_SME(ExpressionMatrix = day5$dat5_counts, GeneMetadata = day5$dat5geneInfo,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

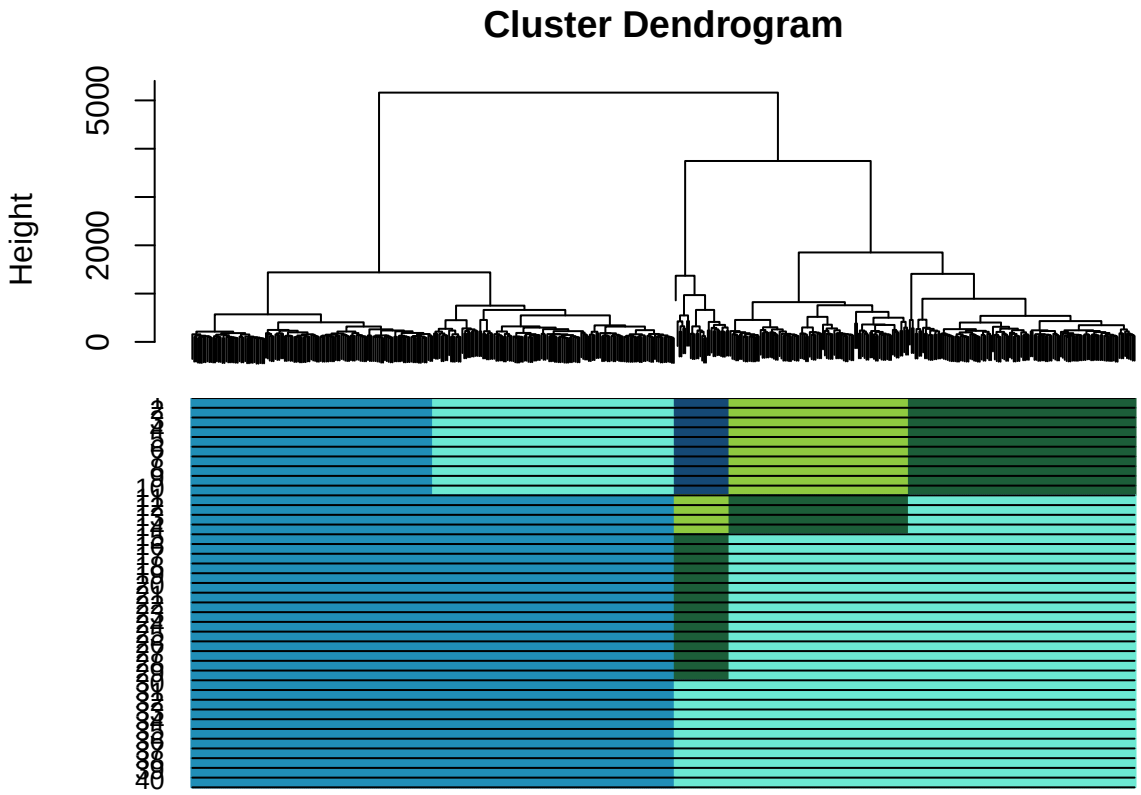
### 3.2 Visualise all cluster results in all iterations

```
#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)
```



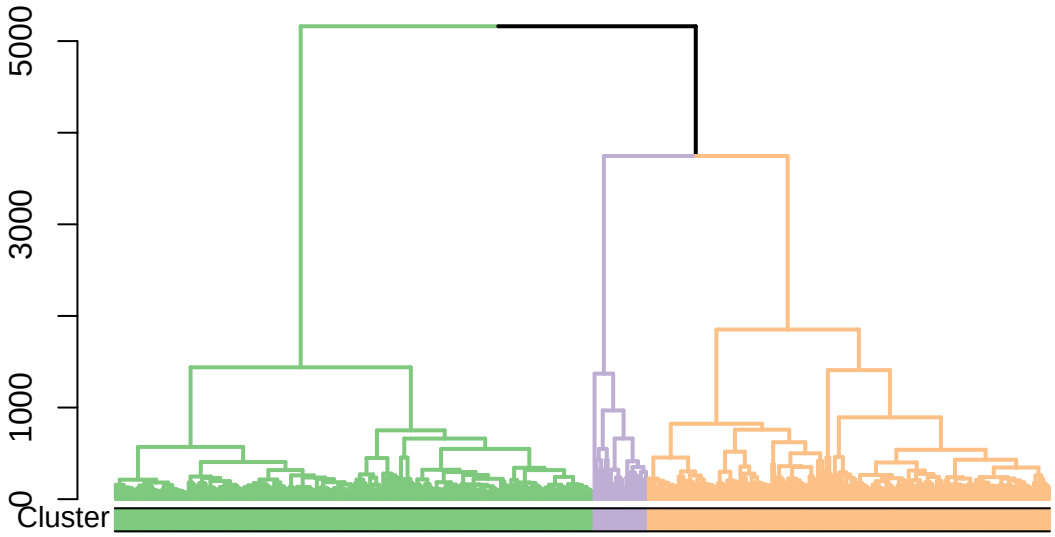

```
#you can customise the cluster color bars (provide color_branch values)
plot_CORE(CORE_cluster$tree, CORE_cluster$Cluster, color_branch = c("#208eb7", "#6ce9d3", "#1c5e39", "#8
```



```
#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", "#8
```

### 3.3 Plot the optimal clustering result

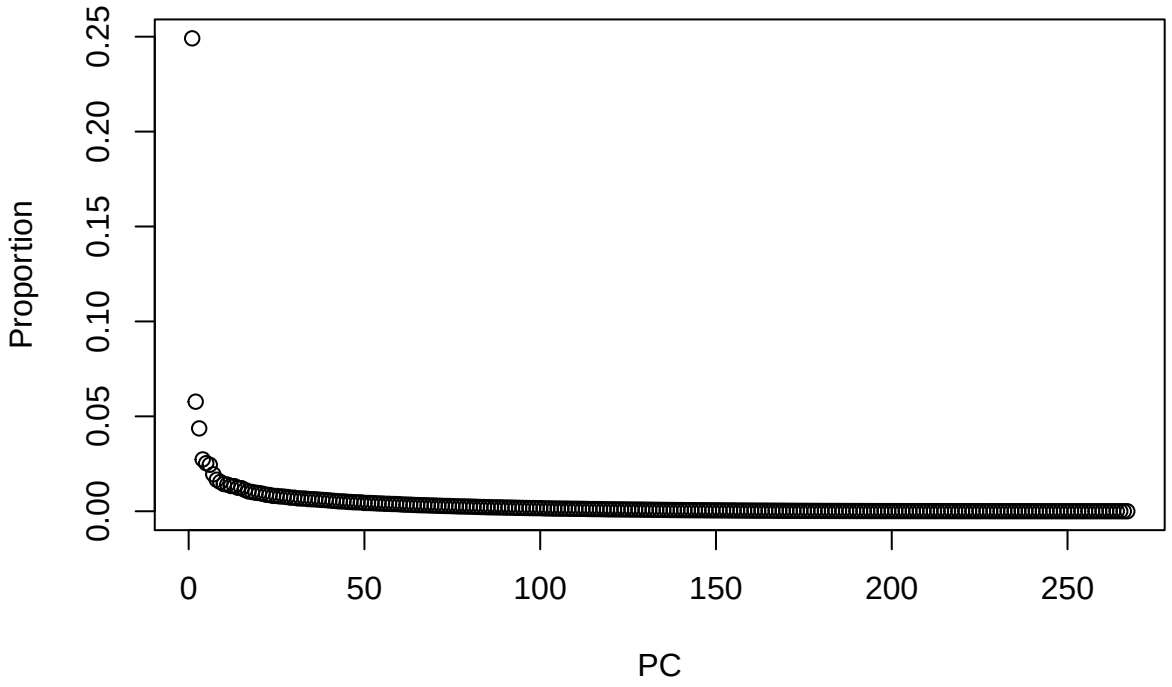
```
#extract optimal index identified by CORE_scGPS
optimal_index = which(CORE_cluster$optimalClust$KeyStats$Height == CORE_cluster$optimalClust$OptimalRes

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



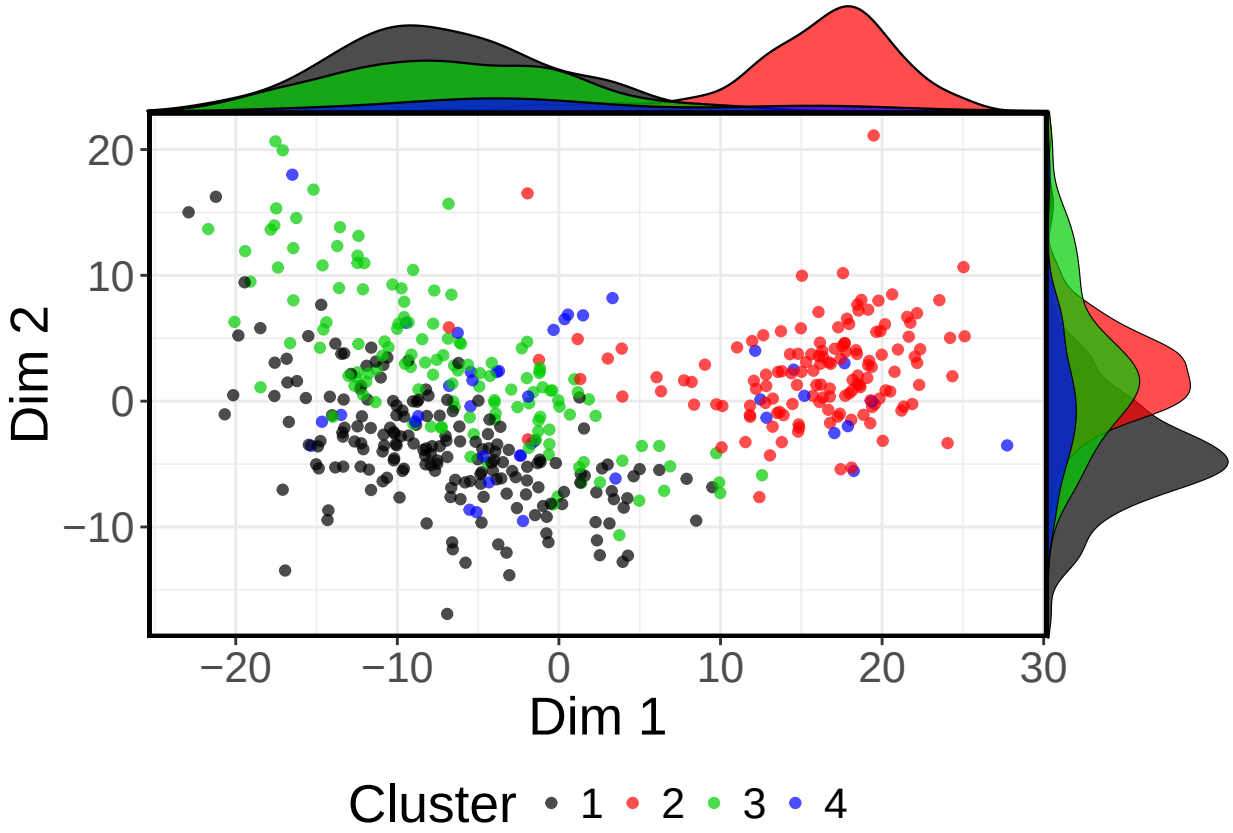

```
#> [1] "Plotting the bar underneath now...."
```

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

```
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..."
```



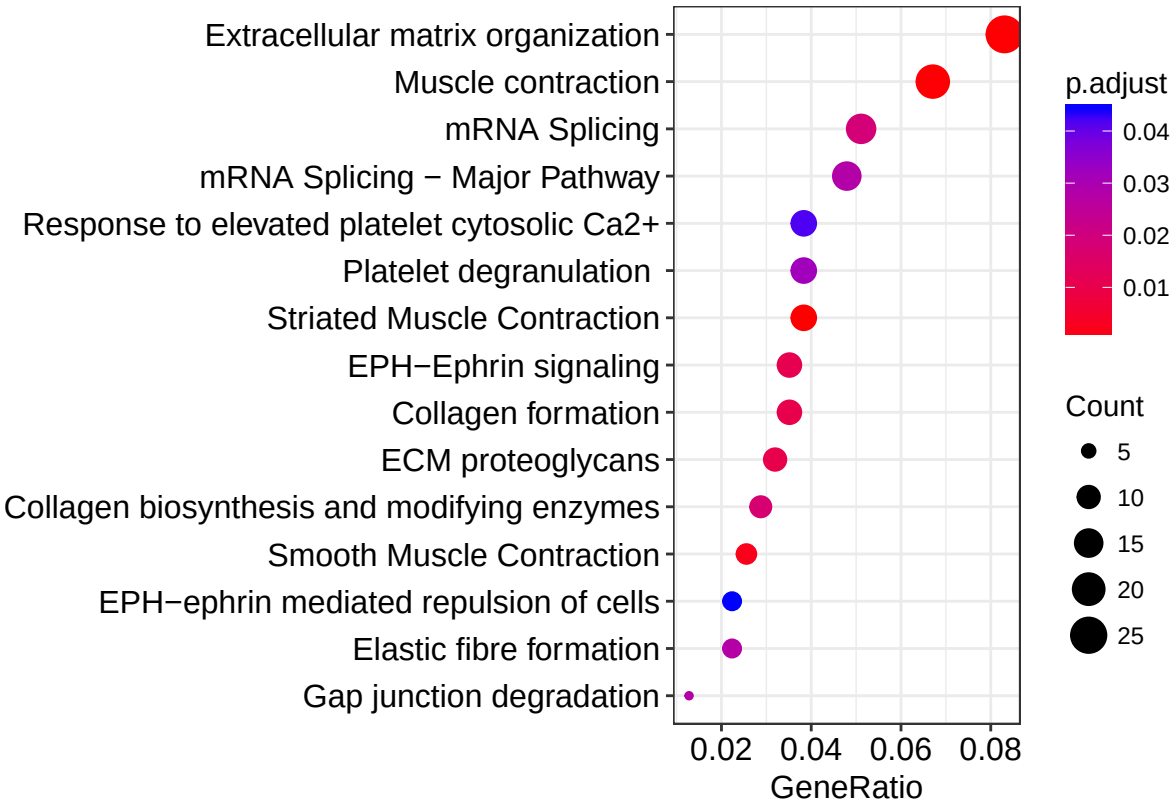

```
#> [1] "find the number of PC..."
#> [1] "perform clustering..."
p2 <-plotReduced_scGPS(t, color_fac = factor(colData(mixedpop2)[,1]),palletes =1:length(unique(colData(m
p2
```




### 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 requied 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,outpu
#> [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 0.358400
```

```
#>   [2,]  1  3.299e-02 0.342100
#>   [3,]  1  6.341e-02 0.326500
#>   [4,]  1  9.176e-02 0.311700
#>   [5,]  1  1.184e-01 0.297500
#>   [6,]  1  1.437e-01 0.284000
#>   [7,]  1  1.677e-01 0.271100
#>   [8,]  1  1.907e-01 0.258800
#>   [9,]  1  2.129e-01 0.247000
#>  [10,]  1  2.342e-01 0.235800
#>  [11,]  1  2.549e-01 0.225100
#>  [12,]  1  2.749e-01 0.214800
#>  [13,]  1  2.943e-01 0.205100
#>  [14,]  1  3.131e-01 0.195800
#>  [15,]  2  3.314e-01 0.186900
#>  [16,]  2  3.516e-01 0.178400
#>  [17,]  2  3.708e-01 0.170300
#>  [18,]  2  3.893e-01 0.162500
#>  [19,]  2  4.070e-01 0.155100
#>  [20,]  2  4.241e-01 0.148100
#>  [21,]  2  4.404e-01 0.141400
#>  [22,]  2  4.562e-01 0.134900
#>  [23,]  2  4.713e-01 0.128800
#>  [24,]  2  4.859e-01 0.122900
#>  [25,]  4  5.009e-01 0.117400
#>  [26,]  4  5.156e-01 0.112000
#>  [27,]  5  5.297e-01 0.106900
#>  [28,]  6  5.433e-01 0.102100
#>  [29,]  6  5.564e-01 0.097430
#>  [30,]  6  5.690e-01 0.093000
#>  [31,]  6  5.811e-01 0.088770
#>  [32,]  7  5.929e-01 0.084740
#>  [33,]  7  6.042e-01 0.080890
#>  [34,]  6  6.151e-01 0.077210
#>  [35,]  6  6.256e-01 0.073700
#>  [36,]  6  6.356e-01 0.070350
#>  [37,]  6  6.454e-01 0.067150
#>  [38,]  7  6.548e-01 0.064100
#>  [39,]  8  6.644e-01 0.061190
#>  [40,]  8  6.738e-01 0.058410
#>  [41,]  8  6.828e-01 0.055750
#>  [42,]  8  6.915e-01 0.053220
#>  [43,]  8  6.999e-01 0.050800
#>  [44,] 10  7.083e-01 0.048490
#>  [45,] 10  7.165e-01 0.046290
#>  [46,] 10  7.245e-01 0.044180
#>  [47,] 10  7.322e-01 0.042170
#>  [48,] 10  7.397e-01 0.040260
#>  [49,] 10  7.469e-01 0.038430
#>  [50,] 11  7.539e-01 0.036680
#>  [51,] 12  7.607e-01 0.035010
#>  [52,] 12  7.674e-01 0.033420
#>  [53,] 13  7.739e-01 0.031900
#>  [54,] 14  7.804e-01 0.030450
```

```
#>  [55,] 17  7.867e-01 0.029070
#>  [56,] 19  7.930e-01 0.027750
#>  [57,] 20  7.992e-01 0.026490
#>  [58,] 23  8.054e-01 0.025280
#>  [59,] 25  8.116e-01 0.024130
#>  [60,] 25  8.175e-01 0.023040
#>  [61,] 27  8.233e-01 0.021990
#>  [62,] 29  8.292e-01 0.020990
#>  [63,] 29  8.350e-01 0.020040
#>  [64,] 29  8.406e-01 0.019130
#>  [65,] 30  8.461e-01 0.018260
#>  [66,] 32  8.517e-01 0.017430
#>  [67,] 33  8.572e-01 0.016630
#>  [68,] 33  8.625e-01 0.015880
#>  [69,] 33  8.676e-01 0.015160
#>  [70,] 33  8.725e-01 0.014470
#>  [71,] 34  8.774e-01 0.013810
#>  [72,] 34  8.820e-01 0.013180
#>  [73,] 33  8.865e-01 0.012580
#>  [74,] 34  8.908e-01 0.012010
#>  [75,] 34  8.949e-01 0.011470
#>  [76,] 34  8.989e-01 0.010940
#>  [77,] 35  9.027e-01 0.010450
#>  [78,] 37  9.065e-01 0.009972
#>  [79,] 38  9.102e-01 0.009519
#>  [80,] 39  9.138e-01 0.009086
#>  [81,] 41  9.174e-01 0.008673
#>  [82,] 42  9.209e-01 0.008279
#>  [83,] 42  9.242e-01 0.007903
#>  [84,] 43  9.274e-01 0.007544
#>  [85,] 46  9.305e-01 0.007201
#>  [86,] 47  9.335e-01 0.006873
#>  [87,] 49  9.365e-01 0.006561
#>  [88,] 51  9.394e-01 0.006263
#>  [89,] 53  9.422e-01 0.005978
#>  [90,] 53  9.449e-01 0.005706
#>  [91,] 53  9.474e-01 0.005447
#>  [92,] 54  9.499e-01 0.005199
#>  [93,] 56  9.522e-01 0.004963
#>  [94,] 57  9.544e-01 0.004738
#>  [95,] 56  9.565e-01 0.004522
#>  [96,] 56  9.586e-01 0.004317
#>  [97,] 57  9.605e-01 0.004120
#>  [98,] 56  9.623e-01 0.003933
#>  [99,] 56  9.640e-01 0.003754
#> [100,] 57  9.657e-01 0.003584
#> [1] "done bootstrap 1"
#>
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>        Df       %Dev   Lambda
#>   [1,]  0 -1.922e-15 0.377100
#>   [2,]  1  3.652e-02 0.360000
```

```
#>   [3,]  1  7.015e-02 0.343600
#>   [4,]  1  1.014e-01 0.328000
#>   [5,]  1  1.307e-01 0.313100
#>   [6,]  1  1.583e-01 0.298900
#>   [7,]  1  1.845e-01 0.285300
#>   [8,]  1  2.095e-01 0.272300
#>   [9,]  1  2.334e-01 0.259900
#>  [10,]  1  2.564e-01 0.248100
#>  [11,]  1  2.786e-01 0.236900
#>  [12,]  1  3.001e-01 0.226100
#>  [13,]  1  3.208e-01 0.215800
#>  [14,]  1  3.410e-01 0.206000
#>  [15,]  1  3.605e-01 0.196600
#>  [16,]  1  3.795e-01 0.187700
#>  [17,]  1  3.979e-01 0.179200
#>  [18,]  1  4.158e-01 0.171000
#>  [19,]  1  4.331e-01 0.163300
#>  [20,]  1  4.499e-01 0.155800
#>  [21,]  1  4.662e-01 0.148800
#>  [22,]  1  4.820e-01 0.142000
#>  [23,]  1  4.972e-01 0.135500
#>  [24,]  1  5.119e-01 0.129400
#>  [25,]  1  5.262e-01 0.123500
#>  [26,]  1  5.399e-01 0.117900
#>  [27,]  1  5.531e-01 0.112500
#>  [28,]  1  5.659e-01 0.107400
#>  [29,]  1  5.782e-01 0.102500
#>  [30,]  1  5.901e-01 0.097870
#>  [31,]  3  6.020e-01 0.093420
#>  [32,]  3  6.137e-01 0.089170
#>  [33,]  3  6.250e-01 0.085120
#>  [34,]  3  6.358e-01 0.081250
#>  [35,]  3  6.462e-01 0.077560
#>  [36,]  3  6.562e-01 0.074030
#>  [37,]  3  6.659e-01 0.070670
#>  [38,]  3  6.752e-01 0.067460
#>  [39,]  3  6.841e-01 0.064390
#>  [40,]  3  6.927e-01 0.061460
#>  [41,]  3  7.010e-01 0.058670
#>  [42,]  5  7.090e-01 0.056000
#>  [43,]  6  7.170e-01 0.053460
#>  [44,]  6  7.250e-01 0.051030
#>  [45,]  7  7.329e-01 0.048710
#>  [46,]  8  7.406e-01 0.046500
#>  [47,]  8  7.482e-01 0.044380
#>  [48,]  8  7.554e-01 0.042370
#>  [49,]  8  7.624e-01 0.040440
#>  [50,]  9  7.692e-01 0.038600
#>  [51,] 10  7.760e-01 0.036850
#>  [52,] 10  7.826e-01 0.035170
#>  [53,] 11  7.892e-01 0.033570
#>  [54,] 12  7.955e-01 0.032050
#>  [55,] 13  8.016e-01 0.030590
```

```
#>  [56,] 13  8.076e-01 0.029200
#>  [57,] 16  8.136e-01 0.027870
#>  [58,] 15  8.196e-01 0.026610
#>  [59,] 15  8.253e-01 0.025400
#>  [60,] 15  8.308e-01 0.024240
#>  [61,] 15  8.361e-01 0.023140
#>  [62,] 17  8.413e-01 0.022090
#>  [63,] 17  8.465e-01 0.021090
#>  [64,] 17  8.515e-01 0.020130
#>  [65,] 19  8.567e-01 0.019210
#>  [66,] 20  8.617e-01 0.018340
#>  [67,] 22  8.666e-01 0.017510
#>  [68,] 23  8.714e-01 0.016710
#>  [69,] 24  8.761e-01 0.015950
#>  [70,] 25  8.806e-01 0.015230
#>  [71,] 25  8.850e-01 0.014530
#>  [72,] 27  8.893e-01 0.013870
#>  [73,] 27  8.936e-01 0.013240
#>  [74,] 27  8.977e-01 0.012640
#>  [75,] 28  9.017e-01 0.012070
#>  [76,] 29  9.056e-01 0.011520
#>  [77,] 30  9.093e-01 0.010990
#>  [78,] 31  9.129e-01 0.010490
#>  [79,] 34  9.165e-01 0.010020
#>  [80,] 34  9.199e-01 0.009562
#>  [81,] 36  9.232e-01 0.009127
#>  [82,] 38  9.264e-01 0.008712
#>  [83,] 38  9.295e-01 0.008316
#>  [84,] 40  9.325e-01 0.007938
#>  [85,] 40  9.354e-01 0.007578
#>  [86,] 42  9.382e-01 0.007233
#>  [87,] 43  9.409e-01 0.006904
#>  [88,] 44  9.435e-01 0.006591
#>  [89,] 44  9.460e-01 0.006291
#>  [90,] 43  9.484e-01 0.006005
#>  [91,] 44  9.507e-01 0.005732
#>  [92,] 44  9.529e-01 0.005472
#>  [93,] 44  9.550e-01 0.005223
#>  [94,] 43  9.570e-01 0.004986
#>  [95,] 44  9.589e-01 0.004759
#>  [96,] 47  9.607e-01 0.004543
#>  [97,] 48  9.625e-01 0.004336
#>  [98,] 49  9.642e-01 0.004139
#>  [99,] 50  9.658e-01 0.003951
#> [100,] 50  9.673e-01 0.003771
#> [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        88.671875         87.890625 LDA for subpop 1 in target mixedpop2
#> 2  12.093023255814 10.6976744186047 LDA for subpop 2 in target mixedpop2
#> 3 6.89655172413793 6.89655172413793 LDA for subpop 3 in target mixedpop2

#summary results Lasso
summary_prediction_lasso(LSOLDA_dat=LSOLDA_dat, nPredSubpop = row_cluster)
#>                  V1                V2                                     names
#> 1           96.875         98.046875 LASSO for subpop1 in target mixedpop2
#> 2 4.65116279069767 5.58139534883721 LASSO for subpop2 in target mixedpop2
#> 3                NA                NA LASSO for subpop3 in target mixedpop2

#summary deviance
summary_deviance(LSOLDA_dat)
#> $allDeviance
#> [1] "0.6042" "0.6137"
#>
#> $DeviMax
#>         Dfd   Deviance        DEgenes
#> 1         0 -1.922e-15 genes_cluster1
#> 2         1     0.5901 genes_cluster1
#> 3         3     0.6137 genes_cluster1
#> 4 remaining          1        DEgenes
#>
#> $LassoGenesMax
#>                                    1                   name
#> (Intercept)            -2.7901242833            (Intercept)
#> BEX3_ENSG00000166681    0.0019194655   BEX3_ENSG00000166681
#> MALAT1_ENSG00000251562  0.0100101702 MALAT1_ENSG00000251562
#> TPM1_ENSG00000140416    0.0003127856   TPM1_ENSG00000140416
```

### 4.3 Plot the relationship between clusters

*Here we look at one example use case to find relationship between clusters within one sample or between two sample*

```
#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 3.609e-01
#>  [2,]  1  6.422e-02 3.288e-01
#>  [3,]  1  1.196e-01 2.996e-01
```

```
#>  [4,]  1  1.686e-01 2.730e-01
#>  [5,]  1  2.130e-01 2.487e-01
#>  [6,]  1  2.538e-01 2.266e-01
#>  [7,]  1  2.915e-01 2.065e-01
#>  [8,]  1  3.268e-01 1.882e-01
#>  [9,]  1  3.597e-01 1.715e-01
#> [10,]  2  3.944e-01 1.562e-01
#> [11,]  3  4.271e-01 1.423e-01
#> [12,]  4  4.583e-01 1.297e-01
#> [13,]  5  4.884e-01 1.182e-01
#> [14,]  5  5.163e-01 1.077e-01
#> [15,]  5  5.421e-01 9.811e-02
#> [16,]  7  5.663e-01 8.939e-02
#> [17,]  7  5.906e-01 8.145e-02
#> [18,]  7  6.131e-01 7.422e-02
#> [19,]  7  6.340e-01 6.762e-02
#> [20,]  9  6.536e-01 6.162e-02
#> [21,]  8  6.718e-01 5.614e-02
#> [22,] 11  6.891e-01 5.115e-02
#> [23,] 11  7.060e-01 4.661e-02
#> [24,] 12  7.218e-01 4.247e-02
#> [25,] 12  7.367e-01 3.870e-02
#> [26,] 12  7.505e-01 3.526e-02
#> [27,] 13  7.634e-01 3.213e-02
#> [28,] 16  7.756e-01 2.927e-02
#> [29,] 16  7.875e-01 2.667e-02
#> [30,] 19  7.988e-01 2.430e-02
#> [31,] 20  8.094e-01 2.214e-02
#> [32,] 20  8.196e-01 2.018e-02
#> [33,] 20  8.291e-01 1.838e-02
#> [34,] 20  8.380e-01 1.675e-02
#> [35,] 22  8.473e-01 1.526e-02
#> [36,] 23  8.568e-01 1.391e-02
#> [37,] 24  8.659e-01 1.267e-02
#> [38,] 25  8.745e-01 1.155e-02
#> [39,] 27  8.830e-01 1.052e-02
#> [40,] 31  8.915e-01 9.585e-03
#> [41,] 33  8.998e-01 8.734e-03
#> [42,] 34  9.076e-01 7.958e-03
#> [43,] 36  9.149e-01 7.251e-03
#> [44,] 35  9.217e-01 6.607e-03
#> [45,] 35  9.280e-01 6.020e-03
#> [46,] 36  9.339e-01 5.485e-03
#> [47,] 37  9.393e-01 4.998e-03
#> [48,] 37  9.444e-01 4.554e-03
#> [49,] 37  9.490e-01 4.149e-03
#> [50,] 38  9.532e-01 3.781e-03
#> [51,] 39  9.572e-01 3.445e-03
#> [52,] 39  9.608e-01 3.139e-03
#> [53,] 41  9.641e-01 2.860e-03
#> [54,] 41  9.672e-01 2.606e-03
#> [55,] 41  9.701e-01 2.374e-03
#> [56,] 43  9.727e-01 2.163e-03
```

```
#> [57,] 44  9.750e-01 1.971e-03
#> [58,] 44  9.772e-01 1.796e-03
#> [59,] 43  9.792e-01 1.637e-03
#> [60,] 44  9.810e-01 1.491e-03
#> [61,] 45  9.827e-01 1.359e-03
#> [62,] 45  9.842e-01 1.238e-03
#> [63,] 45  9.856e-01 1.128e-03
#> [64,] 47  9.869e-01 1.028e-03
#> [65,] 48  9.880e-01 9.365e-04
#> [66,] 48  9.891e-01 8.533e-04
#> [67,] 48  9.901e-01 7.775e-04
#> [68,] 48  9.909e-01 7.084e-04
#> [69,] 50  9.918e-01 6.455e-04
#> [70,] 50  9.925e-01 5.882e-04
#> [71,] 50  9.932e-01 5.359e-04
#> [72,] 50  9.938e-01 4.883e-04
#> [73,] 50  9.943e-01 4.449e-04
#> [74,] 50  9.948e-01 4.054e-04
#> [75,] 50  9.953e-01 3.694e-04
#> [76,] 50  9.957e-01 3.366e-04
#> [77,] 50  9.961e-01 3.067e-04
#> [78,] 50  9.964e-01 2.794e-04
#> [79,] 50  9.967e-01 2.546e-04
#> [80,] 50  9.970e-01 2.320e-04
#> [81,] 50  9.973e-01 2.114e-04
#> [82,] 52  9.975e-01 1.926e-04
#> [83,] 52  9.978e-01 1.755e-04
#> [84,] 52  9.980e-01 1.599e-04
#> [85,] 53  9.981e-01 1.457e-04
#> [86,] 53  9.983e-01 1.327e-04
#> [87,] 51  9.984e-01 1.210e-04
#> [88,] 50  9.986e-01 1.102e-04
#> [89,] 52  9.987e-01 1.004e-04
#> [90,] 52  9.988e-01 9.150e-05
#> [91,] 52  9.989e-01 8.337e-05
#> [92,] 52  9.990e-01 7.596e-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_s
#>
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>        Df       %Dev    Lambda
#>  [1,]   0 -2.403e-15 2.321e-01
#>  [2,]   3  2.878e-02 2.115e-01
#>  [3,]   3  6.314e-02 1.927e-01
#>  [4,]   3  9.220e-02 1.756e-01
#>  [5,]   4  1.171e-01 1.600e-01
#>  [6,]  10  1.460e-01 1.458e-01
```

```
#>  [7,]  12  1.750e-01 1.328e-01
#>  [8,]  14  2.041e-01 1.210e-01
#>  [9,]  16  2.313e-01 1.103e-01
#> [10,]  18  2.574e-01 1.005e-01
#> [11,]  18  2.834e-01 9.154e-02
#> [12,]  21  3.074e-01 8.340e-02
#> [13,]  22  3.305e-01 7.599e-02
#> [14,]  22  3.513e-01 6.924e-02
#> [15,]  24  3.702e-01 6.309e-02
#> [16,]  26  3.883e-01 5.749e-02
#> [17,]  26  4.048e-01 5.238e-02
#> [18,]  29  4.205e-01 4.773e-02
#> [19,]  32  4.364e-01 4.349e-02
#> [20,]  33  4.534e-01 3.962e-02
#> [21,]  33  4.686e-01 3.610e-02
#> [22,]  35  4.826e-01 3.290e-02
#> [23,]  36  4.971e-01 2.997e-02
#> [24,]  39  5.129e-01 2.731e-02
#> [25,]  42  5.280e-01 2.488e-02
#> [26,]  45  5.466e-01 2.267e-02
#> [27,]  50  5.677e-01 2.066e-02
#> [28,]  56  5.923e-01 1.882e-02
#> [29,]  58  6.158e-01 1.715e-02
#> [30,]  61  6.377e-01 1.563e-02
#> [31,]  65  6.583e-01 1.424e-02
#> [32,]  69  6.784e-01 1.297e-02
#> [33,]  69  6.980e-01 1.182e-02
#> [34,]  71  7.161e-01 1.077e-02
#> [35,]  76  7.338e-01 9.815e-03
#> [36,]  78  7.512e-01 8.943e-03
#> [37,]  81  7.679e-01 8.149e-03
#> [38,]  84  7.845e-01 7.425e-03
#> [39,]  86  8.009e-01 6.765e-03
#> [40,]  90  8.167e-01 6.164e-03
#> [41,]  90  8.315e-01 5.617e-03
#> [42,]  96  8.461e-01 5.118e-03
#> [43,]  97  8.599e-01 4.663e-03
#> [44,]  97  8.727e-01 4.249e-03
#> [45,]  96  8.843e-01 3.871e-03
#> [46,]  98  8.947e-01 3.527e-03
#> [47,]  98  9.043e-01 3.214e-03
#> [48,]  99  9.131e-01 2.928e-03
#> [49,] 100  9.209e-01 2.668e-03
#> [50,] 100  9.282e-01 2.431e-03
#> [51,] 101  9.347e-01 2.215e-03
#> [52,] 100  9.406e-01 2.018e-03
#> [53,] 101  9.460e-01 1.839e-03
#> [54,] 102  9.508e-01 1.676e-03
#> [55,] 103  9.553e-01 1.527e-03
#> [56,] 103  9.593e-01 1.391e-03
#> [57,] 103  9.630e-01 1.268e-03
#> [58,] 104  9.663e-01 1.155e-03
#> [59,] 104  9.693e-01 1.052e-03
```

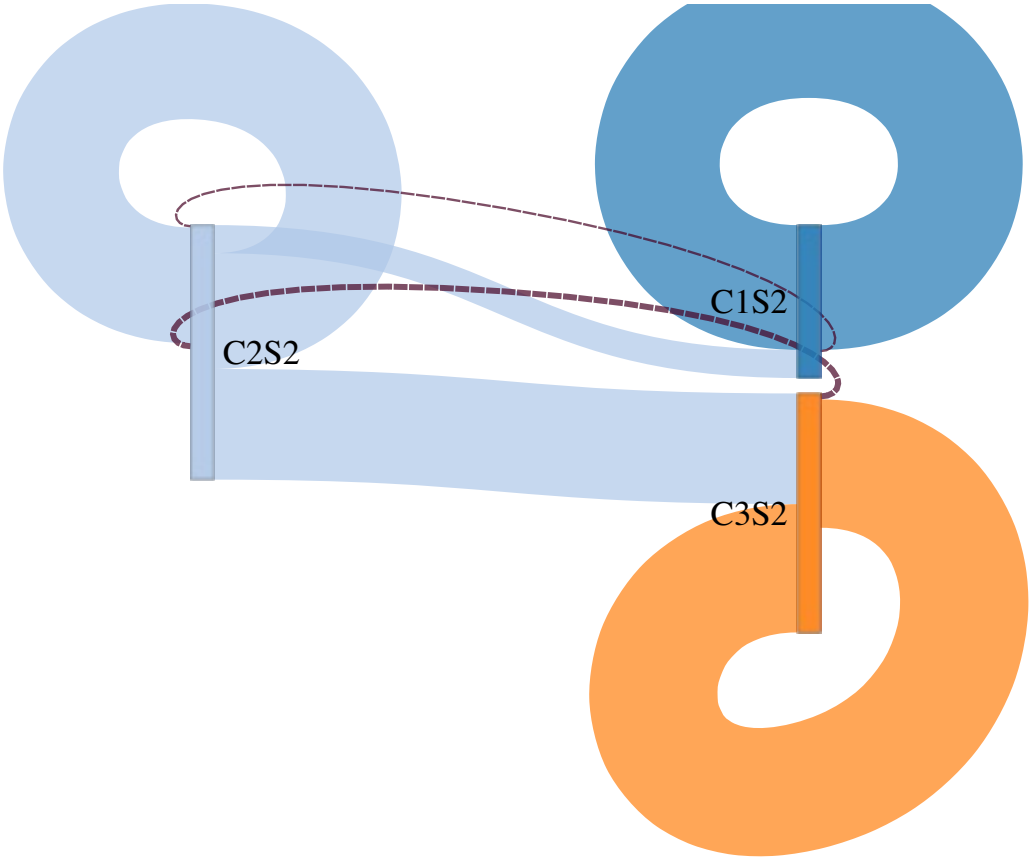
```
#> [60,] 103  9.721e-01 9.589e-04
#> [61,] 103  9.746e-01 8.737e-04
#> [62,] 105  9.768e-01 7.961e-04
#> [63,] 106  9.789e-01 7.254e-04
#> [64,] 106  9.808e-01 6.610e-04
#> [65,] 107  9.825e-01 6.022e-04
#> [66,] 108  9.841e-01 5.487e-04
#> [67,] 108  9.855e-01 5.000e-04
#> [68,] 108  9.868e-01 4.556e-04
#> [69,] 108  9.880e-01 4.151e-04
#> [70,] 108  9.890e-01 3.782e-04
#> [71,] 109  9.900e-01 3.446e-04
#> [72,] 109  9.909e-01 3.140e-04
#> [73,] 108  9.917e-01 2.861e-04
#> [74,] 108  9.924e-01 2.607e-04
#> [75,] 109  9.931e-01 2.375e-04
#> [76,] 109  9.937e-01 2.164e-04
#> [77,] 109  9.943e-01 1.972e-04
#> [78,] 109  9.948e-01 1.797e-04
#> [79,] 110  9.952e-01 1.637e-04
#> [80,] 111  9.956e-01 1.492e-04
#> [81,] 111  9.960e-01 1.359e-04
#> [82,] 111  9.964e-01 1.239e-04
#> [83,] 111  9.967e-01 1.128e-04
#> [84,] 111  9.970e-01 1.028e-04
#> [85,] 111  9.973e-01 9.369e-05
#> [86,] 111  9.975e-01 8.537e-05
#> [87,] 111  9.977e-01 7.778e-05
#> [88,] 111  9.979e-01 7.087e-05
#> [89,] 112  9.981e-01 6.458e-05
#> [90,] 113  9.983e-01 5.884e-05
#> [91,] 113  9.984e-01 5.361e-05
#> [92,] 114  9.986e-01 4.885e-05
#> [93,] 114  9.987e-01 4.451e-05
#> [94,] 114  9.988e-01 4.056e-05
#> [95,] 114  9.989e-01 3.695e-05
#> [96,] 114  9.990e-01 3.367e-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 0.00000 0.445700
#>   [2,]  1 0.05097 0.425400
#>   [3,]  1 0.09771 0.406100
#>   [4,]  1 0.14080 0.387600
#>   [5,]  1 0.18080 0.370000
```

```
#>   [6,]  1 0.21800 0.353200
#>   [7,]  1 0.25280 0.337100
#>   [8,]  1 0.28550 0.321800
#>   [9,]  1 0.31620 0.307200
#>  [10,]  1 0.34510 0.293200
#>  [11,]  1 0.37240 0.279900
#>  [12,]  1 0.39830 0.267200
#>  [13,]  1 0.42280 0.255000
#>  [14,]  1 0.44610 0.243400
#>  [15,]  1 0.46820 0.232400
#>  [16,]  1 0.48930 0.221800
#>  [17,]  1 0.50930 0.211700
#>  [18,]  1 0.52840 0.202100
#>  [19,]  1 0.54660 0.192900
#>  [20,]  1 0.56390 0.184100
#>  [21,]  1 0.58050 0.175800
#>  [22,]  1 0.59630 0.167800
#>  [23,]  1 0.61150 0.160200
#>  [24,]  1 0.62590 0.152900
#>  [25,]  1 0.63980 0.145900
#>  [26,]  1 0.65300 0.139300
#>  [27,]  2 0.66670 0.133000
#>  [28,]  2 0.67990 0.126900
#>  [29,]  2 0.69250 0.121200
#>  [30,]  2 0.70460 0.115600
#>  [31,]  2 0.71610 0.110400
#>  [32,]  2 0.72710 0.105400
#>  [33,]  2 0.73770 0.100600
#>  [34,]  3 0.74840 0.096010
#>  [35,]  3 0.75890 0.091650
#>  [36,]  3 0.76890 0.087480
#>  [37,]  3 0.77850 0.083510
#>  [38,]  3 0.78770 0.079710
#>  [39,]  3 0.79650 0.076090
#>  [40,]  3 0.80480 0.072630
#>  [41,]  3 0.81290 0.069330
#>  [42,]  3 0.82050 0.066180
#>  [43,]  4 0.82830 0.063170
#>  [44,]  4 0.83560 0.060300
#>  [45,]  5 0.84270 0.057560
#>  [46,]  6 0.84970 0.054940
#>  [47,]  6 0.85630 0.052450
#>  [48,]  6 0.86270 0.050060
#>  [49,]  7 0.86880 0.047790
#>  [50,]  7 0.87460 0.045610
#>  [51,]  7 0.88020 0.043540
#>  [52,]  8 0.88570 0.041560
#>  [53,]  8 0.89090 0.039670
#>  [54,]  9 0.89590 0.037870
#>  [55,]  9 0.90080 0.036150
#>  [56,]  9 0.90540 0.034510
#>  [57,]  9 0.90980 0.032940
#>  [58,]  9 0.91390 0.031440
```

```
#>  [59,]  9 0.91790 0.030010
#>  [60,]  9 0.92170 0.028650
#>  [61,]  9 0.92530 0.027350
#>  [62,]  9 0.92870 0.026100
#>  [63,] 11 0.93200 0.024920
#>  [64,] 11 0.93520 0.023780
#>  [65,] 11 0.93820 0.022700
#>  [66,] 11 0.94110 0.021670
#>  [67,] 11 0.94380 0.020690
#>  [68,] 11 0.94640 0.019750
#>  [69,] 11 0.94890 0.018850
#>  [70,] 11 0.95120 0.017990
#>  [71,] 11 0.95350 0.017170
#>  [72,] 11 0.95560 0.016390
#>  [73,] 11 0.95760 0.015650
#>  [74,] 11 0.95960 0.014940
#>  [75,] 11 0.96140 0.014260
#>  [76,] 11 0.96320 0.013610
#>  [77,] 11 0.96490 0.012990
#>  [78,] 11 0.96650 0.012400
#>  [79,] 11 0.96800 0.011840
#>  [80,] 11 0.96950 0.011300
#>  [81,] 11 0.97090 0.010790
#>  [82,] 12 0.97220 0.010300
#>  [83,] 12 0.97350 0.009827
#>  [84,] 12 0.97470 0.009381
#>  [85,] 12 0.97590 0.008954
#>  [86,] 12 0.97700 0.008547
#>  [87,] 12 0.97800 0.008159
#>  [88,] 12 0.97900 0.007788
#>  [89,] 12 0.98000 0.007434
#>  [90,] 12 0.98090 0.007096
#>  [91,] 12 0.98180 0.006774
#>  [92,] 12 0.98260 0.006466
#>  [93,] 12 0.98340 0.006172
#>  [94,] 12 0.98420 0.005891
#>  [95,] 12 0.98490 0.005624
#>  [96,] 12 0.98560 0.005368
#>  [97,] 12 0.98620 0.005124
#>  [98,] 12 0.98690 0.004891
#>  [99,] 12 0.98750 0.004669
#> [100,] 12 0.98800 0.004457
#> [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
```