scGPS introduction

Quan Nguyen

2018-04-20

Contents

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')

# R/3.4.1 or above is required

# 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
```

A simple workflow of the scGPS:

Given a mixed population with known subpopulations, estimate transition scores between these subpopulation

```
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

# run the test bootstrap
sink("temp")
LSOLDA_dat <- bootstrap_scGPS(nboots = 2, mixedpop1 = mixedpop1,
    mixedpop2 = mixedpop2, genes = genes, c_selectID, listData = list())
#>
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>         Df       %Dev    Lambda
#>    [1,]  0 -2.563e-15 2.692e-01
#>    [2,]  1  3.572e-02 2.453e-01
#>    [3,]  2  7.492e-02 2.235e-01
#>    [4,]  2  1.105e-01 2.036e-01
#>    [5,]  2  1.415e-01 1.855e-01
#>    [6,]  3  1.733e-01 1.690e-01
#>    [7,]  3  2.025e-01 1.540e-01
#>    [8,]  3  2.286e-01 1.403e-01
#>    [9,]  3  2.520e-01 1.279e-01
#>   [10,]  4  2.743e-01 1.165e-01
#>   [11,]  6  2.966e-01 1.062e-01
#>   [12,]  6  3.186e-01 9.674e-02
#>   [13,]  7  3.386e-01 8.814e-02
#>   [14,]  8  3.574e-01 8.031e-02
#>  [15,]  8  3.746e-01 7.318e-02
#>  [16,]  8  3.900e-01 6.668e-02
#>  [17,]  9  4.043e-01 6.075e-02
#>  [18,] 11  4.203e-01 5.536e-02
#>  [19,] 14  4.374e-01 5.044e-02
#>  [20,] 17  4.556e-01 4.596e-02
#>  [21,] 19  4.735e-01 4.187e-02
#>  [22,] 22  4.906e-01 3.815e-02
#>  [23,] 27  5.080e-01 3.476e-02
#>  [24,] 27  5.257e-01 3.168e-02
#>  [25,] 30  5.428e-01 2.886e-02
#>  [26,] 34  5.601e-01 2.630e-02
#>  [27,] 37  5.776e-01 2.396e-02
#>  [28,] 36  5.953e-01 2.183e-02
#>  [29,] 37  6.107e-01 1.989e-02
#>  [30,] 39  6.253e-01 1.813e-02
#>  [31,] 40  6.398e-01 1.652e-02
#>  [32,] 44  6.544e-01 1.505e-02
#>  [33,] 46  6.684e-01 1.371e-02
#>  [34,] 47  6.817e-01 1.249e-02
#>  [35,] 51  6.952e-01 1.138e-02
#>  [36,] 52  7.086e-01 1.037e-02
#>  [37,] 53  7.212e-01 9.451e-03
#>  [38,] 54  7.329e-01 8.612e-03
#>  [39,] 56  7.441e-01 7.846e-03
#>  [40,] 58  7.554e-01 7.149e-03
#>  [41,] 58  7.661e-01 6.514e-03
#>  [42,] 60  7.766e-01 5.936e-03
#>  [43,] 63  7.871e-01 5.408e-03
#>  [44,] 65  7.975e-01 4.928e-03
#>  [45,] 64  8.075e-01 4.490e-03
#>  [46,] 67  8.169e-01 4.091e-03
#>  [47,] 67  8.258e-01 3.728e-03
#>  [48,] 65  8.342e-01 3.397e-03
#>  [49,] 66  8.419e-01 3.095e-03
#>  [50,] 67  8.493e-01 2.820e-03
#>  [51,] 66  8.565e-01 2.569e-03
#>  [52,] 67  8.635e-01 2.341e-03
#>  [53,] 68  8.704e-01 2.133e-03
#>  [54,] 67  8.772e-01 1.944e-03
#>  [55,] 67  8.838e-01 1.771e-03
#>  [56,] 69  8.908e-01 1.614e-03
#>  [57,] 69  8.976e-01 1.470e-03
#>  [58,] 69  9.043e-01 1.340e-03
#>  [59,] 68  9.108e-01 1.221e-03
#>  [60,] 67  9.171e-01 1.112e-03
#>  [61,] 66  9.232e-01 1.013e-03
#>  [62,] 66  9.291e-01 9.234e-04
#>  [63,] 67  9.349e-01 8.414e-04
#>  [64,] 68  9.405e-01 7.666e-04
#>  [65,] 68  9.458e-01 6.985e-04
#>  [66,] 69  9.507e-01 6.365e-04
#>  [67,] 69  9.551e-01 5.799e-04
#>  [68,] 69  9.593e-01 5.284e-04
#>  [69,] 69  9.630e-01 4.815e-04
#>  [70,] 69  9.664e-01 4.387e-04
#>  [71,] 69  9.695e-01 3.997e-04
#>  [72,] 69  9.723e-01 3.642e-04
#>  [73,] 69  9.748e-01 3.318e-04
#>  [74,] 69  9.771e-01 3.024e-04
#>  [75,] 69  9.792e-01 2.755e-04
#>  [76,] 69  9.811e-01 2.510e-04
#>  [77,] 68  9.828e-01 2.287e-04
#>  [78,] 68  9.843e-01 2.084e-04
#>  [79,] 68  9.858e-01 1.899e-04
#>  [80,] 68  9.870e-01 1.730e-04
#>  [81,] 68  9.882e-01 1.577e-04
#>  [82,] 68  9.893e-01 1.436e-04
#>  [83,] 68  9.902e-01 1.309e-04
#>  [84,] 68  9.911e-01 1.193e-04
#>  [85,] 68  9.919e-01 1.087e-04
#>  [86,] 68  9.926e-01 9.901e-05
#>  [87,] 69  9.933e-01 9.022e-05
#>  [88,] 69  9.939e-01 8.220e-05
#>  [89,] 69  9.944e-01 7.490e-05
#>  [90,] 69  9.949e-01 6.824e-05
#>  [91,] 69  9.954e-01 6.218e-05
#>  [92,] 69  9.958e-01 5.666e-05
#>  [93,] 69  9.961e-01 5.162e-05
#>  [94,] 68  9.965e-01 4.704e-05
#>  [95,] 69  9.968e-01 4.286e-05
#>  [96,] 69  9.971e-01 3.905e-05
#>  [97,] 69  9.973e-01 3.558e-05
#>  [98,] 69  9.976e-01 3.242e-05
#>  [99,] 69  9.978e-01 2.954e-05
#> [100,] 69  9.980e-01 2.692e-05
#> [1] "done bootstrap 1"
#> 
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial") 
#> 
#>        Df       %Dev    Lambda
#>   [1,]  0 -2.563e-15 2.735e-01
#>   [2,]  2  4.502e-02 2.492e-01
#>   [3,]  2  8.746e-02 2.270e-01
#>   [4,]  2  1.242e-01 2.069e-01
#>   [5,]  2  1.566e-01 1.885e-01
#>   [6,]  2  1.853e-01 1.718e-01
#>   [7,]  2  2.110e-01 1.565e-01
#>   [8,]  2  2.341e-01 1.426e-01
#>   [9,]  2  2.550e-01 1.299e-01
#>  [10,]  4  2.765e-01 1.184e-01
#>  [11,]  7  2.987e-01 1.079e-01
#>  [12,]  8  3.254e-01 9.828e-02
#>  [13,]  9  3.505e-01 8.955e-02
#>  [14,]  9  3.740e-01 8.160e-02
#>  [15,]  9  3.950e-01 7.435e-02
#>  [16,] 10  4.147e-01 6.774e-02
#>  [17,] 11  4.328e-01 6.172e-02
#>  [18,] 12  4.497e-01 5.624e-02
#>  [19,] 15  4.668e-01 5.124e-02
#>  [20,] 15  4.839e-01 4.669e-02
#>  [21,] 16  5.001e-01 4.254e-02
#>  [22,] 19  5.196e-01 3.876e-02
#>  [23,] 20  5.392e-01 3.532e-02
#>  [24,] 22  5.582e-01 3.218e-02
#>  [25,] 25  5.766e-01 2.932e-02
#>  [26,] 25  5.939e-01 2.672e-02
#>  [27,] 26  6.105e-01 2.435e-02
#>  [28,] 27  6.262e-01 2.218e-02
#>  [29,] 34  6.421e-01 2.021e-02
#>  [30,] 35  6.582e-01 1.842e-02
#>  [31,] 35  6.733e-01 1.678e-02
#>  [32,] 37  6.874e-01 1.529e-02
#>  [33,] 42  7.023e-01 1.393e-02
#>  [34,] 43  7.170e-01 1.269e-02
#>  [35,] 43  7.309e-01 1.157e-02
#>  [36,] 44  7.436e-01 1.054e-02
#>  [37,] 46  7.555e-01 9.602e-03
#>  [38,] 47  7.666e-01 8.749e-03
#>  [39,] 46  7.772e-01 7.972e-03
#>  [40,] 47  7.873e-01 7.264e-03
#>  [41,] 49  7.977e-01 6.618e-03
#>  [42,] 51  8.076e-01 6.031e-03
#>  [43,] 53  8.170e-01 5.495e-03
#>  [44,] 54  8.261e-01 5.007e-03
#>  [45,] 59  8.350e-01 4.562e-03
#>  [46,] 61  8.438e-01 4.157e-03
#>  [47,] 61  8.522e-01 3.787e-03
#>  [48,] 60  8.602e-01 3.451e-03
#>  [49,] 62  8.680e-01 3.144e-03
#>  [50,] 64  8.758e-01 2.865e-03
#>  [51,] 63  8.835e-01 2.610e-03
#>  [52,] 63  8.909e-01 2.379e-03
#>  [53,] 63  8.982e-01 2.167e-03
#>  [54,] 64  9.050e-01 1.975e-03
#>  [55,] 62  9.115e-01 1.799e-03
#>  [56,] 62  9.175e-01 1.639e-03
#>  [57,] 63  9.232e-01 1.494e-03
#>  [58,] 65  9.288e-01 1.361e-03
#>  [59,] 67  9.342e-01 1.240e-03
#>  [60,] 67  9.394e-01 1.130e-03
#>  [61,] 68  9.444e-01 1.030e-03
#>  [62,] 67  9.490e-01 9.382e-04
#>  [63,] 69  9.533e-01 8.548e-04
#>  [64,] 69  9.573e-01 7.789e-04
#>  [65,] 68  9.611e-01 7.097e-04
#>  [66,] 68  9.645e-01 6.466e-04
#>  [67,] 67  9.677e-01 5.892e-04
#>  [68,] 69  9.706e-01 5.368e-04
#>  [69,] 70  9.733e-01 4.892e-04
#>  [70,] 70  9.757e-01 4.457e-04
#>  [71,] 70  9.780e-01 4.061e-04
#>  [72,] 70  9.800e-01 3.700e-04
#>  [73,] 70  9.818e-01 3.372e-04
#>  [74,] 69  9.834e-01 3.072e-04
#>  [75,] 68  9.849e-01 2.799e-04
#>  [76,] 68  9.863e-01 2.550e-04
#>  [77,] 68  9.875e-01 2.324e-04
#>  [78,] 68  9.886e-01 2.117e-04
#>  [79,] 68  9.897e-01 1.929e-04
#>  [80,] 68  9.906e-01 1.758e-04
#>  [81,] 68  9.914e-01 1.602e-04
#>  [82,] 68  9.922e-01 1.459e-04
#>  [83,] 69  9.929e-01 1.330e-04
#>  [84,] 69  9.935e-01 1.212e-04
#>  [85,] 69  9.941e-01 1.104e-04
#>  [86,] 69  9.946e-01 1.006e-04
#>  [87,] 69  9.951e-01 9.166e-05
#>  [88,] 69  9.955e-01 8.352e-05
#>  [89,] 69  9.959e-01 7.610e-05
#>  [90,] 69  9.963e-01 6.934e-05
#>  [91,] 69  9.966e-01 6.318e-05
#>  [92,] 69  9.969e-01 5.756e-05
#>  [93,] 69  9.972e-01 5.245e-05
#>  [94,] 69  9.974e-01 4.779e-05
#>  [95,] 69  9.977e-01 4.355e-05
#>  [96,] 69  9.979e-01 3.968e-05
#>  [97,] 69  9.980e-01 3.615e-05
#>  [98,] 69  9.982e-01 3.294e-05
#>  [99,] 68  9.984e-01 3.001e-05
#> [100,] 68  9.985e-01 2.735e-05
#> [1] "done bootstrap 2"
sink()

# 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] 98.93048
#>
#> [[1]][[3]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 100
#>
```

```
#> [[1]][[5]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 98.49624
#>
#> [[1]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 100
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LASSO for subpop1 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 100
#>
#> [[2]][[3]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 99.28571
#>
#> [[2]][[5]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 97.74436
#>
#> [[2]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 100
LSOLDA_dat$LDAPredict
#> [[1]]
#> [[1]][[1]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[1]][[2]]
#> [1] 83.42246
#>
#> [[1]][[3]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 70
#>
#> [[1]][[5]]
#> [1] "LDA for subpop 3 in target mixedpop2"
```

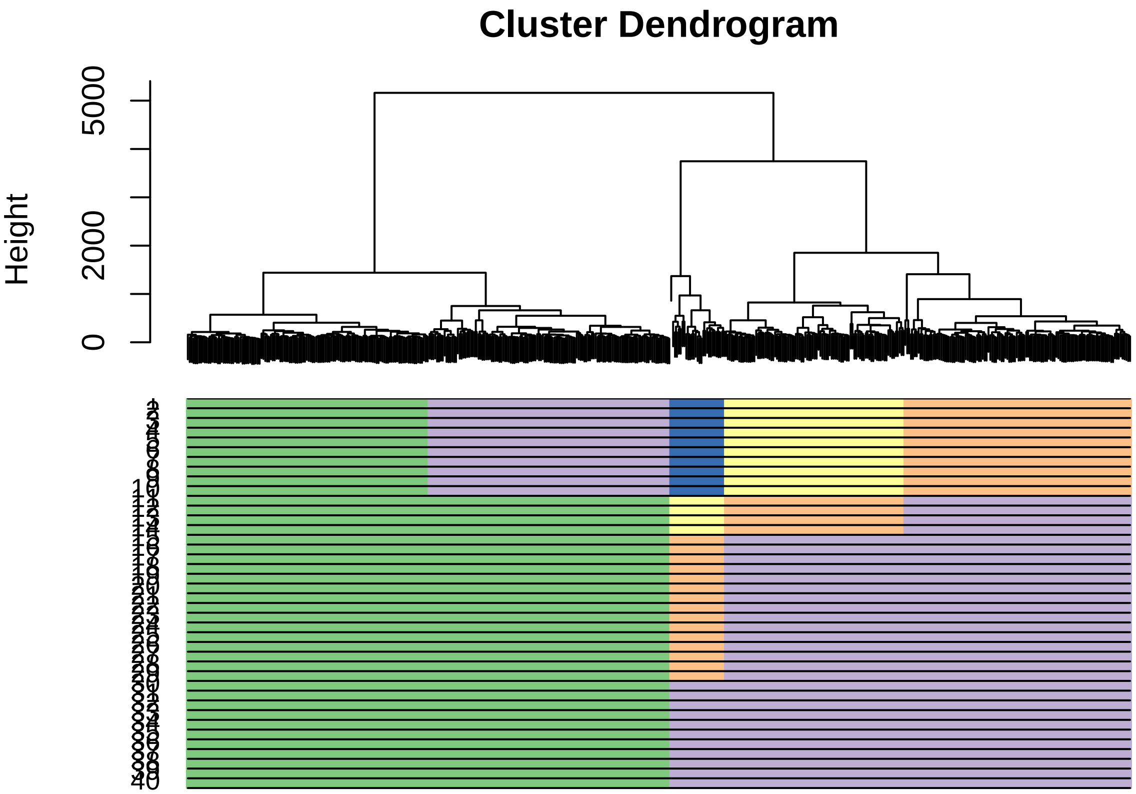
```
#>
#> [[1]][[6]]
#> [1] 54.88722
#>
#> [[1]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 65
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 33.15508
#>
#> [[2]][[3]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 54.28571
#>
#> [[2]][[5]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 30.82707
#>
#> [[2]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 45

# summary results LDA
summary_prediction_lda(LSOLDA_dat = LSOLDA_dat, nPredSubpop = 4)
#>                  V1               V2                                names
#> 1 83.4224598930481 33.1550802139037 LDA for subpop 1 in target mixedpop2
#> 2               70 54.2857142857143 LDA for subpop 2 in target mixedpop2
#> 3 54.8872180451128 30.8270676691729 LDA for subpop 3 in target mixedpop2
#> 4               65               45 LDA for subpop 4 in target mixedpop2

# summary results Lasso
summary_prediction_lasso(LSOLDA_dat = LSOLDA_dat, nPredSubpop = 4)
#>                  V1               V2                                 names
#> 1 98.9304812834225              100 LASSO for subpop1 in target mixedpop2
#> 2              100 99.2857142857143 LASSO for subpop2 in target mixedpop2
#> 3 98.4962406015038 97.7443609022556 LASSO for subpop3 in target mixedpop2
#> 4              100              100 LASSO for subpop4 in target mixedpop2

# summary deviance
```

```
summary_deviance(object = LSOLDA_dat)
#> $allDeviance
#> [1] "0.1733" "0.374"
#>
#> $DeviMax
#>         Dfd   Deviance        DEgenes
#> 1         0 -2.563e-15 genes_cluster1
#> 2         2      0.255 genes_cluster1
#> 3         4     0.2765 genes_cluster1
#> 4         7     0.2987 genes_cluster1
#> 5         8     0.3254 genes_cluster1
#> 6         9      0.374 genes_cluster1
#> 7 remaining          1        DEgenes
#>
#> $LassoGenesMax
#>                                  1                  name
#> (Intercept)            0.05419448           (Intercept)
#> CXCR4_ENSG00000121966 -0.02224918 CXCR4_ENSG00000121966
#> FN1_ENSG00000115414   -0.01035844   FN1_ENSG00000115414
#> FOXC1_ENSG00000054598  0.10397767 FOXC1_ENSG00000054598
#> GJA1_ENSG00000152661  -0.04297188  GJA1_ENSG00000152661
#> T_ENSG00000164458      0.06802307     T_ENSG00000164458
#> SOX17_ENSG00000164736 -0.03171682 SOX17_ENSG00000164736
#> TPM1_ENSG00000140416  -0.05214643  TPM1_ENSG00000140416
#> MESP1_ENSG00000166823  0.07837061 MESP1_ENSG00000166823
#> FOXA2_ENSG00000125798 -0.01473921 FOXA2_ENSG00000125798
```

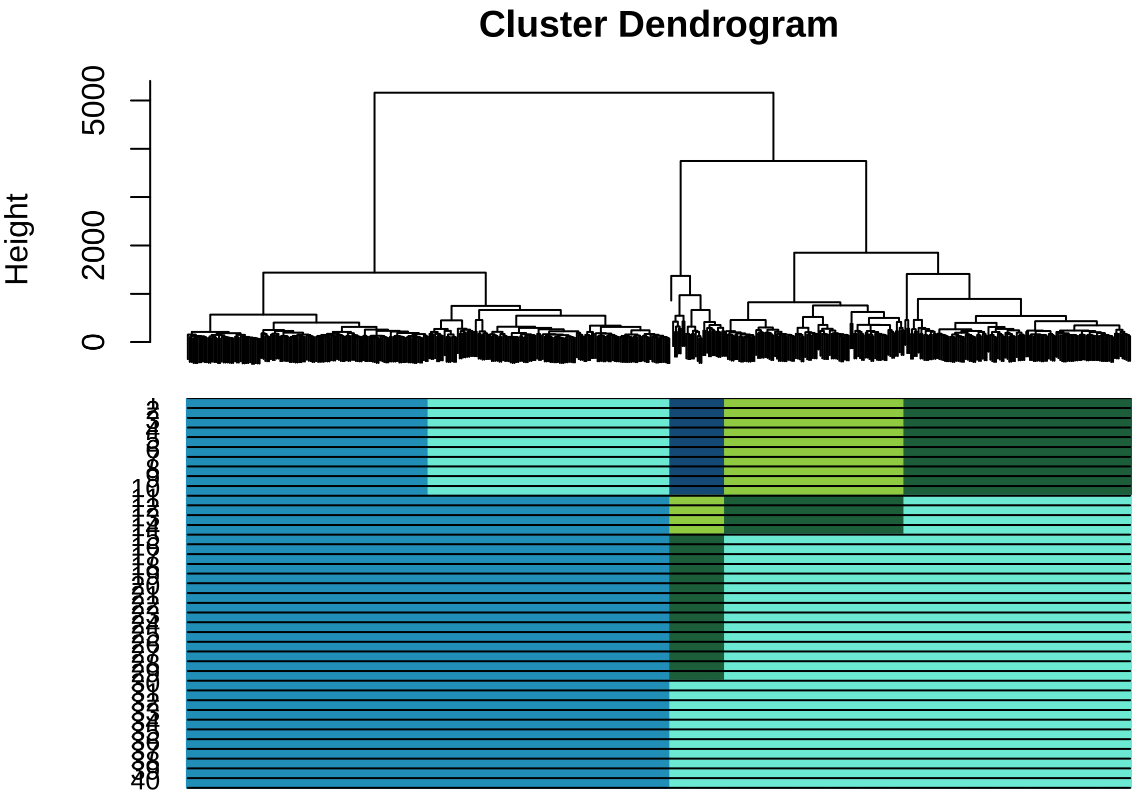
A complete workflow of the scGPS:

Given an unknown mixed population, find clusters and estimate relationship between clusters

```
#Let's find clustering information using CORE (skip this if clusters are known)
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

#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)
```

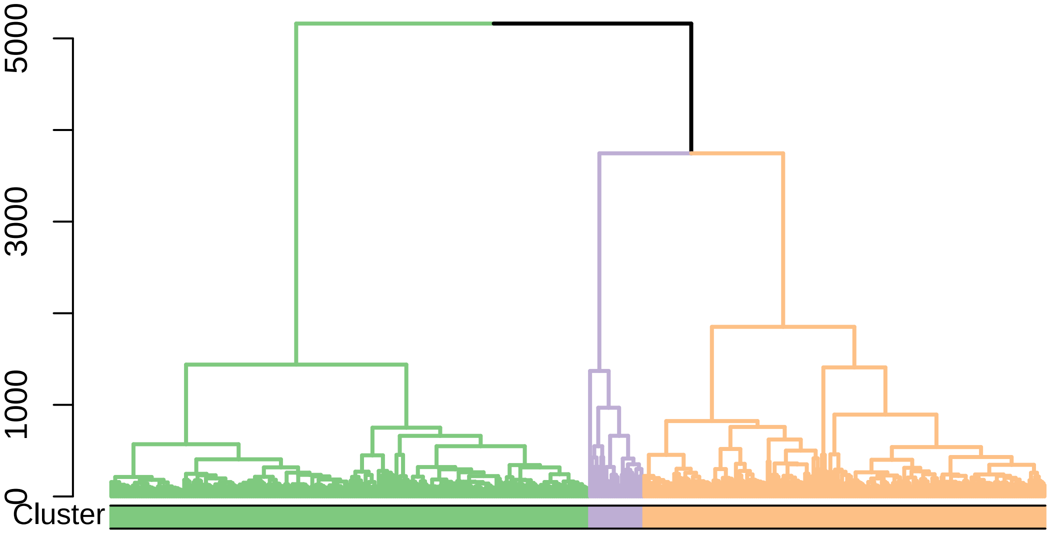


```
#you can customise the cluster color bars (provide color_branch values)
plot_CORE(CORE_cluster$tree, CORE_cluster$Cluster, color_branch = c("#208eb7", "#6ce9d3", "#1c5e39", "#8
```



Let’s plot just the optimal clustering result (with colored dendrogram)

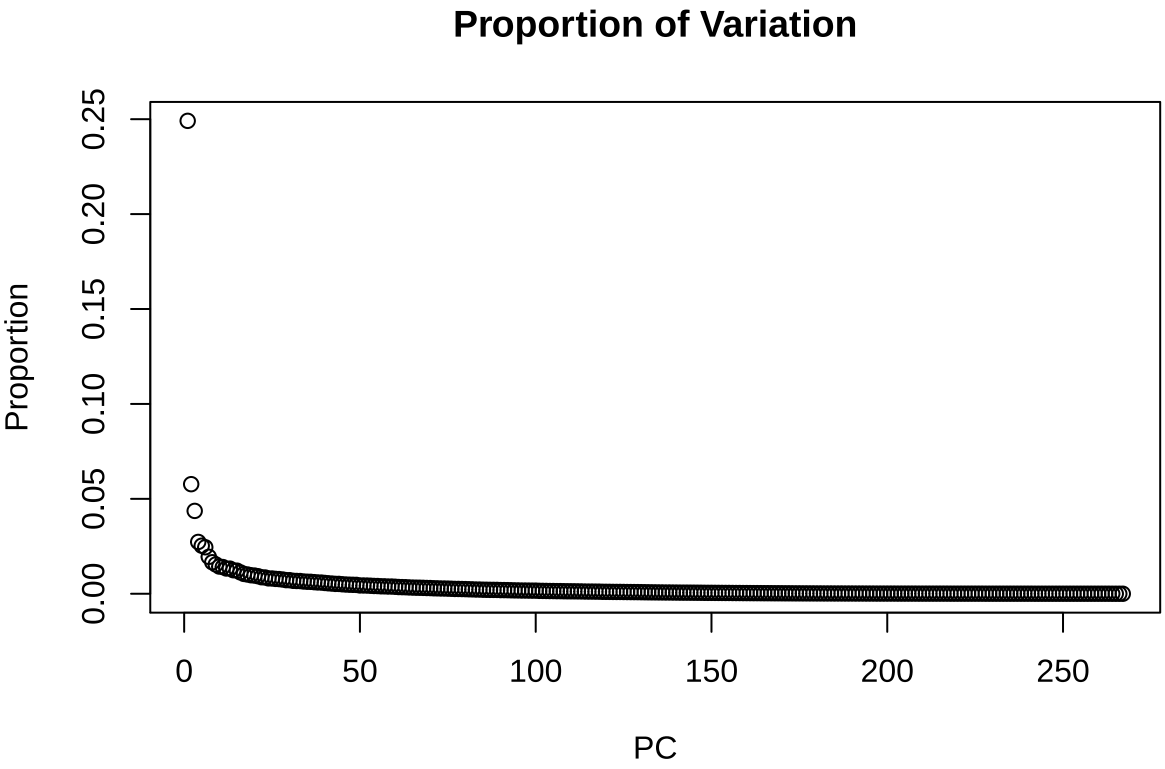
```
#extract optimal index identified by CORE_scGPS
optimal_index = which(CORE_cluster$optimalClust$KeyStats$Height == CORE_cluster$optimalClust$OptimalRes)

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...."
```



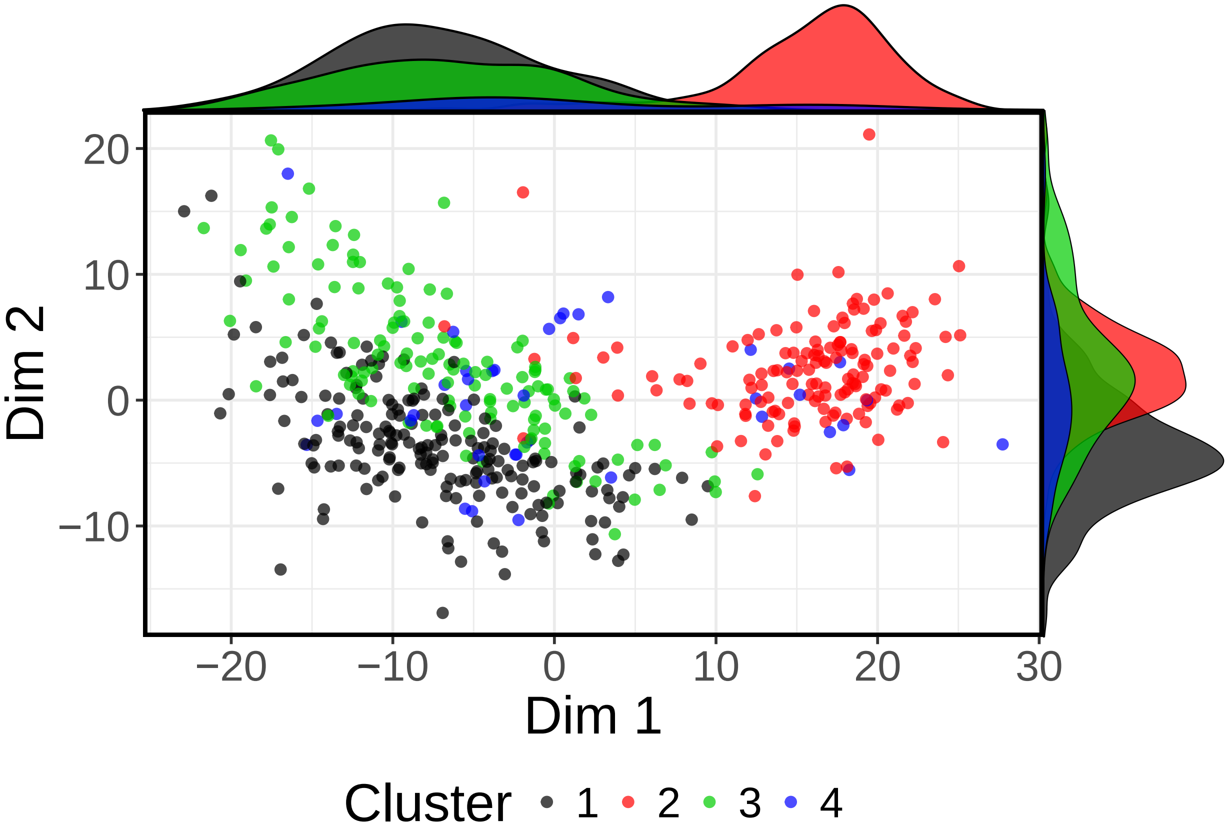
```
#> [1] "Plotting the bar underneath now...."
```

Let’s 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..."
```

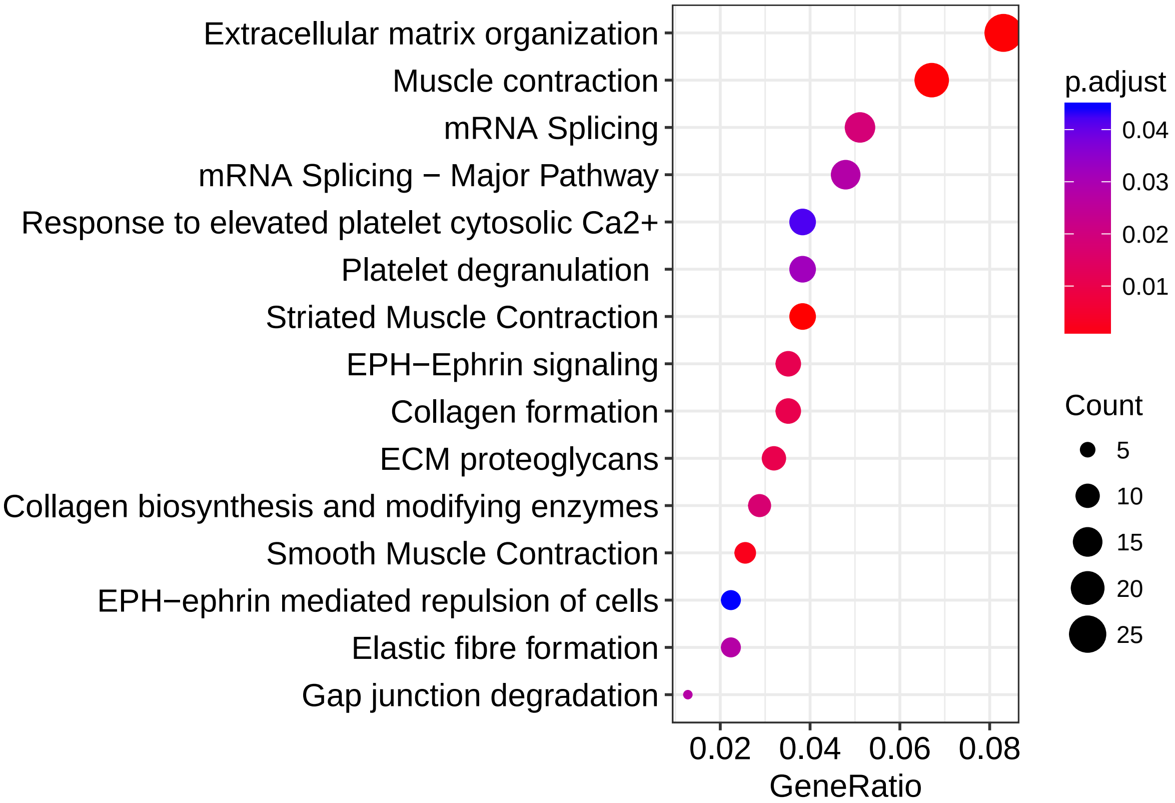


```
#> [1] "find the number of PC..."
#> [1] "perform clustering..."
p2 <-plotReduced_scGPS(t, color_fac = factor(colData(mixedpop2)[,1]),palletes =1:length(unique(colData(m
#>
#> Attaching package: 'cowplot'
#> The following object is masked from 'package:ggplot2':
#>
#>     ggsave
p2
```



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)
```



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 0.358100
#>   [2,]  1  3.293e-02 0.341800
#>   [3,]  1  6.331e-02 0.326300
#>   [4,]  1  9.161e-02 0.311400
#>   [5,]  1  1.182e-01 0.297300
#>   [6,]  1  1.434e-01 0.283800
#>   [7,]  1  1.675e-01 0.270900
#>   [8,]  1  1.905e-01 0.258600
#>   [9,]  1  2.126e-01 0.246800
#>  [10,]  1  2.340e-01 0.235600
#>  [11,]  1  2.547e-01 0.224900
#>  [12,]  1  2.747e-01 0.214700
#>  [13,]  1  2.942e-01 0.204900
#>  [14,]  1  3.131e-01 0.195600
#>  [15,]  1  3.315e-01 0.186700
#>  [16,]  1  3.493e-01 0.178200
#>  [17,]  1  3.667e-01 0.170100
#>  [18,]  1  3.835e-01 0.162400
#>  [19,]  1  3.999e-01 0.155000
#>  [20,]  1  4.158e-01 0.148000
#>  [21,]  1  4.313e-01 0.141200
#>  [22,]  1  4.462e-01 0.134800
#>  [23,]  1  4.607e-01 0.128700
#>  [24,]  1  4.748e-01 0.122800
#>  [25,]  2  4.888e-01 0.117300
#>  [26,]  2  5.029e-01 0.111900
#>  [27,]  2  5.165e-01 0.106800
#>  [28,]  2  5.295e-01 0.102000
#>  [29,]  2  5.421e-01 0.097350
#>  [30,]  2  5.542e-01 0.092920
#>  [31,]  2  5.659e-01 0.088700
#>  [32,]  2  5.771e-01 0.084670
#>  [33,]  3  5.880e-01 0.080820
#>  [34,]  3  5.988e-01 0.077150
#>  [35,]  3  6.092e-01 0.073640
#>  [36,]  4  6.193e-01 0.070290
#>  [37,]  4  6.297e-01 0.067100
#>  [38,]  4  6.397e-01 0.064050
#>  [39,]  4  6.494e-01 0.061140
#>  [40,]  5  6.588e-01 0.058360
#>  [41,]  5  6.682e-01 0.055710
#>  [42,]  6  6.772e-01 0.053170
#>  [43,]  8  6.864e-01 0.050760
#>  [44,]  9  6.956e-01 0.048450
#>  [45,] 10  7.049e-01 0.046250
#>  [46,] 10  7.138e-01 0.044150
#>  [47,] 11  7.226e-01 0.042140
#>  [48,] 11  7.311e-01 0.040220
#>  [49,] 11  7.393e-01 0.038400
#>  [50,] 12  7.474e-01 0.036650
#>  [51,] 12  7.554e-01 0.034980
#>  [52,] 13  7.632e-01 0.033390
#>  [53,] 13  7.707e-01 0.031880
#>  [54,] 14  7.779e-01 0.030430
#>  [55,] 15  7.849e-01 0.029040
#>  [56,] 15  7.918e-01 0.027720
#>  [57,] 15  7.985e-01 0.026460
#>  [58,] 16  8.049e-01 0.025260
#>  [59,] 16  8.111e-01 0.024110
#>  [60,] 19  8.173e-01 0.023020
#>  [61,] 20  8.237e-01 0.021970
#>  [62,] 20  8.298e-01 0.020970
#>  [63,] 21  8.358e-01 0.020020
#>  [64,] 21  8.416e-01 0.019110
#>  [65,] 23  8.473e-01 0.018240
#>  [66,] 25  8.530e-01 0.017410
#>  [67,] 25  8.586e-01 0.016620
#>  [68,] 25  8.639e-01 0.015870
#>  [69,] 26  8.691e-01 0.015140
#>  [70,] 27  8.741e-01 0.014460
#>  [71,] 27  8.789e-01 0.013800
#>  [72,] 28  8.836e-01 0.013170
#>  [73,] 30  8.881e-01 0.012570
#>  [74,] 32  8.925e-01 0.012000
#>  [75,] 36  8.970e-01 0.011460
#>  [76,] 36  9.014e-01 0.010940
#>  [77,] 36  9.056e-01 0.010440
#>  [78,] 37  9.097e-01 0.009964
#>  [79,] 37  9.135e-01 0.009511
#>  [80,] 37  9.173e-01 0.009079
#>  [81,] 36  9.208e-01 0.008666
#>  [82,] 36  9.242e-01 0.008272
#>  [83,] 36  9.274e-01 0.007896
#>  [84,] 36  9.305e-01 0.007537
#>  [85,] 36  9.335e-01 0.007195
#>  [86,] 38  9.364e-01 0.006868
#>  [87,] 39  9.391e-01 0.006555
#>  [88,] 39  9.417e-01 0.006258
#>  [89,] 41  9.443e-01 0.005973
#>  [90,] 40  9.467e-01 0.005702
#>  [91,] 41  9.490e-01 0.005442
#>  [92,] 41  9.512e-01 0.005195
#>  [93,] 42  9.533e-01 0.004959
#>  [94,] 42  9.554e-01 0.004734
#>  [95,] 43  9.573e-01 0.004518
#>  [96,] 45  9.592e-01 0.004313
#>  [97,] 47  9.611e-01 0.004117
#>  [98,] 49  9.628e-01 0.003930
#>  [99,] 49  9.645e-01 0.003751
#> [100,] 49  9.661e-01 0.003581
#> [1] "done bootstrap 1"
#> 
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial") 
#> 
#>        Df       %Dev   Lambda
#>   [1,]  0 -1.922e-15 0.336200
#>   [2,]  1  2.909e-02 0.320900
#>   [3,]  1  5.618e-02 0.306400
#>   [4,]  1  8.186e-02 0.292400
#>   [5,]  1  1.065e-01 0.279100
#>   [6,]  1  1.304e-01 0.266500
#>   [7,]  1  1.536e-01 0.254300
#>   [8,]  1  1.762e-01 0.242800
#>   [9,]  1  1.982e-01 0.231700
#>  [10,]  1  2.197e-01 0.221200
#>  [11,]  1  2.407e-01 0.211200
#>  [12,]  1  2.611e-01 0.201600
#>  [13,]  1  2.811e-01 0.192400
#>  [14,]  2  3.011e-01 0.183700
#>  [15,]  2  3.222e-01 0.175300
#>  [16,]  2  3.425e-01 0.167300
#>  [17,]  3  3.622e-01 0.159700
#>  [18,]  4  3.815e-01 0.152500
#>  [19,]  4  4.000e-01 0.145500
#>  [20,]  4  4.178e-01 0.138900
#>  [21,]  4  4.348e-01 0.132600
#>  [22,]  4  4.512e-01 0.126600
#>  [23,]  4  4.669e-01 0.120800
#>  [24,]  4  4.820e-01 0.115300
#>  [25,]  4  4.965e-01 0.110100
#>  [26,]  4  5.104e-01 0.105100
#>  [27,]  4  5.238e-01 0.100300
#>  [28,]  4  5.366e-01 0.095760
#>  [29,]  4  5.489e-01 0.091410
#>  [30,]  4  5.607e-01 0.087250
#>  [31,]  5  5.735e-01 0.083290
#>  [32,]  5  5.857e-01 0.079500
#>  [33,]  7  5.977e-01 0.075890
#>  [34,]  8  6.096e-01 0.072440
#>  [35,]  7  6.211e-01 0.069150
#>  [36,]  7  6.321e-01 0.066000
#>  [37,]  9  6.428e-01 0.063000
#>  [38,] 10  6.532e-01 0.060140
#>  [39,] 10  6.636e-01 0.057410
#>  [40,] 11  6.739e-01 0.054800
#>  [41,] 11  6.837e-01 0.052310
#>  [42,] 12  6.933e-01 0.049930
#>  [43,] 13  7.026e-01 0.047660
#>  [44,] 12  7.115e-01 0.045490
#>  [45,] 12  7.200e-01 0.043430
#>  [46,] 11  7.282e-01 0.041450
#>  [47,] 11  7.360e-01 0.039570
#>  [48,] 12  7.436e-01 0.037770
#>  [49,] 13  7.510e-01 0.036050
#>  [50,] 14  7.583e-01 0.034410
#>  [51,] 15  7.654e-01 0.032850
#>  [52,] 15  7.724e-01 0.031360
#>  [53,] 16  7.792e-01 0.029930
#>  [54,] 16  7.858e-01 0.028570
#>  [55,] 18  7.923e-01 0.027270
#>  [56,] 18  7.985e-01 0.026030
#>  [57,] 20  8.046e-01 0.024850
#>  [58,] 20  8.107e-01 0.023720
#>  [59,] 20  8.166e-01 0.022640
#>  [60,] 21  8.223e-01 0.021610
#>  [61,] 24  8.279e-01 0.020630
#>  [62,] 25  8.334e-01 0.019690
#>  [63,] 28  8.390e-01 0.018800
#>  [64,] 28  8.446e-01 0.017940
#>  [65,] 29  8.501e-01 0.017130
#>  [66,] 30  8.555e-01 0.016350
#>  [67,] 34  8.609e-01 0.015610
#>  [68,] 36  8.665e-01 0.014900
#>  [69,] 37  8.718e-01 0.014220
#>  [70,] 36  8.770e-01 0.013570
#>  [71,] 36  8.819e-01 0.012960
#>  [72,] 37  8.867e-01 0.012370
#>  [73,] 39  8.914e-01 0.011810
#>  [74,] 40  8.960e-01 0.011270
#>  [75,] 41  9.005e-01 0.010760
#>  [76,] 42  9.048e-01 0.010270
#>  [77,] 43  9.089e-01 0.009801
#>  [78,] 43  9.128e-01 0.009356
#>  [79,] 44  9.166e-01 0.008930
#>  [80,] 45  9.202e-01 0.008525
#>  [81,] 46  9.237e-01 0.008137
#>  [82,] 47  9.270e-01 0.007767
#>  [83,] 46  9.302e-01 0.007414
#>  [84,] 45  9.333e-01 0.007077
#>  [85,] 45  9.362e-01 0.006756
#>  [86,] 44  9.390e-01 0.006449
#>  [87,] 44  9.416e-01 0.006155
#>  [88,] 44  9.442e-01 0.005876
#>  [89,] 44  9.466e-01 0.005609
#>  [90,] 43  9.490e-01 0.005354
#>  [91,] 43  9.512e-01 0.005110
#>  [92,] 44  9.533e-01 0.004878
#>  [93,] 45  9.553e-01 0.004656
#>  [94,] 46  9.573e-01 0.004445
#>  [95,] 48  9.592e-01 0.004243
#>  [96,] 48  9.610e-01 0.004050
#>  [97,] 48  9.627e-01 0.003866
#>  [98,] 49  9.643e-01 0.003690
#>  [99,] 49  9.659e-01 0.003522
#> [100,] 49  9.674e-01 0.003362
#> [1] "done bootstrap 2"
sink()
```

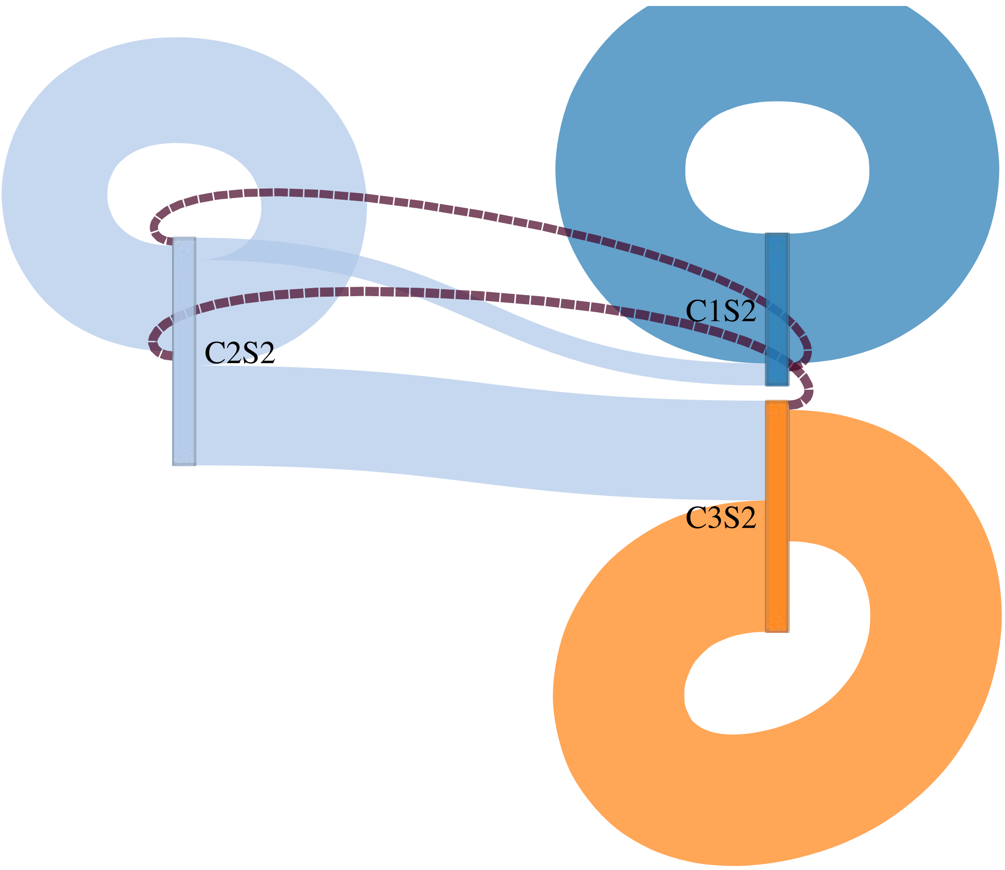
Display summary results for the prediction

```
#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           84.375 LDA for subpop 1 in target mixedpop2
#> 2 8.83720930232558 10.6976744186047 LDA for subpop 2 in target mixedpop2
#> 3 3.44827586206897 3.44827586206897 LDA for subpop 3 in target mixedpop2

#summary results Lasso
summary_prediction_lasso(LSOLDA_dat=LSOLDA_dat, nPredSubpop = row_cluster)
```

```
#>                  V1               V2                                 names
#> 1         98.828125        99.609375 LASSO for subpop1 in target mixedpop2
#> 2 7.44186046511628 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] "0.5988" "0.4"
#>
#> $DeviMax
#>         Dfd   Deviance        DEgenes
#> 1         0 -1.922e-15 genes_cluster1
#> 2         1     0.4748 genes_cluster1
#> 3         2     0.5771 genes_cluster1
#> 4         3     0.5988 genes_cluster1
#> 5 remaining          1        DEgenes
#>
#> $LassoGenesMax
#>                                  1                  name
#> (Intercept)            -3.020620419           (Intercept)
#> PPIC_ENSG00000168938    0.006559629   PPIC_ENSG00000168938
#> BEX3_ENSG00000166681    0.013616897   BEX3_ENSG00000166681
#> MALAT1_ENSG00000251562  0.009911637 MALAT1_ENSG00000251562
```

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)

sink("temp")
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 3.537e-01
#>  [2,]  1  6.185e-02 3.223e-01
#>  [3,]  1  1.159e-01 2.936e-01
#>  [4,]  1  1.649e-01 2.676e-01
#>  [5,]  1  2.103e-01 2.438e-01
#>  [6,]  1  2.530e-01 2.221e-01
#>  [7,]  1  2.932e-01 2.024e-01
#>  [8,]  1  3.309e-01 1.844e-01
#>  [9,]  1  3.663e-01 1.680e-01
#> [10,]  1  3.993e-01 1.531e-01
#> [11,]  1  4.300e-01 1.395e-01
#> [12,]  1  4.585e-01 1.271e-01
#> [13,]  1  4.848e-01 1.158e-01
#> [14,]  2  5.105e-01 1.055e-01
#> [15,]  2  5.349e-01 9.615e-02
#> [16,]  3  5.581e-01 8.761e-02
#> [17,]  4  5.813e-01 7.983e-02
#> [18,]  5  6.028e-01 7.274e-02
#> [19,]  6  6.232e-01 6.628e-02
#> [20,]  7  6.424e-01 6.039e-02
#> [21,]  7  6.600e-01 5.502e-02
#> [22,]  8  6.766e-01 5.013e-02
#> [23,]  9  6.920e-01 4.568e-02
#> [24,] 10  7.066e-01 4.162e-02
#> [25,] 12  7.210e-01 3.793e-02
#> [26,] 12  7.345e-01 3.456e-02
#> [27,] 14  7.476e-01 3.149e-02
#> [28,] 15  7.599e-01 2.869e-02
#> [29,] 16  7.714e-01 2.614e-02
#> [30,] 16  7.822e-01 2.382e-02
#> [31,] 17  7.928e-01 2.170e-02
#> [32,] 22  8.031e-01 1.977e-02
#> [33,] 21  8.140e-01 1.802e-02
#> [34,] 22  8.242e-01 1.642e-02
#> [35,] 23  8.341e-01 1.496e-02
#> [36,] 26  8.433e-01 1.363e-02
#> [37,] 28  8.534e-01 1.242e-02
#> [38,] 28  8.634e-01 1.132e-02
#> [39,] 28  8.727e-01 1.031e-02
#> [40,] 29  8.815e-01 9.394e-03
#> [41,] 30  8.899e-01 8.560e-03
#> [42,] 31  8.980e-01 7.799e-03
#> [43,] 29  9.055e-01 7.106e-03
#> [44,] 32  9.128e-01 6.475e-03
#> [45,] 34  9.197e-01 5.900e-03
#> [46,] 39  9.262e-01 5.376e-03
#> [47,] 39  9.324e-01 4.898e-03
#> [48,] 39  9.381e-01 4.463e-03
#> [49,] 41  9.434e-01 4.067e-03
#> [50,] 42  9.483e-01 3.705e-03
#> [51,] 42  9.528e-01 3.376e-03
#> [52,] 43  9.569e-01 3.076e-03
#> [53,] 44  9.606e-01 2.803e-03
#> [54,] 46  9.641e-01 2.554e-03
#> [55,] 48  9.672e-01 2.327e-03
#> [56,] 48  9.701e-01 2.120e-03
#> [57,] 49  9.727e-01 1.932e-03
#> [58,] 50  9.751e-01 1.760e-03
#> [59,] 50  9.773e-01 1.604e-03
#> [60,] 52  9.793e-01 1.461e-03
#> [61,] 53  9.812e-01 1.332e-03
#> [62,] 53  9.828e-01 1.213e-03
#> [63,] 54  9.844e-01 1.106e-03
#> [64,] 54  9.857e-01 1.007e-03
#> [65,] 54  9.870e-01 9.178e-04
#> [66,] 54  9.881e-01 8.363e-04
#> [67,] 54  9.892e-01 7.620e-04
#> [68,] 55  9.902e-01 6.943e-04
#> [69,] 55  9.910e-01 6.326e-04
#> [70,] 55  9.918e-01 5.764e-04
#> [71,] 55  9.925e-01 5.252e-04
#> [72,] 56  9.932e-01 4.786e-04
#> [73,] 56  9.938e-01 4.360e-04
#> [74,] 56  9.944e-01 3.973e-04
#> [75,] 56  9.949e-01 3.620e-04
#> [76,] 57  9.953e-01 3.299e-04
#> [77,] 58  9.957e-01 3.006e-04
#> [78,] 58  9.961e-01 2.739e-04
#> [79,] 58  9.965e-01 2.495e-04
#> [80,] 58  9.968e-01 2.274e-04
#> [81,] 58  9.971e-01 2.072e-04
#> [82,] 58  9.973e-01 1.888e-04
#> [83,] 59  9.976e-01 1.720e-04
#> [84,] 59  9.978e-01 1.567e-04
#> [85,] 59  9.980e-01 1.428e-04
#> [86,] 59  9.981e-01 1.301e-04
#> [87,] 59  9.983e-01 1.185e-04
#> [88,] 59  9.985e-01 1.080e-04
#> [89,] 60  9.986e-01 9.842e-05
#> [90,] 60  9.987e-01 8.967e-05
#> [91,] 60  9.988e-01 8.171e-05
#> [92,] 60  9.989e-01 7.445e-05
#> [93,] 60  9.990e-01 6.783e-05
#> [1] "done bootstrap 1"
sink()

c_selectID <- 2
genes = DEgenes$DE_Subpop2vsRemaining$id[1:200]
genes <- gsub("_.*", "", genes)
sink("temp")
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 2.140e-01
#>  [2,]  3  2.846e-02 1.950e-01
#>  [3,]  5  6.381e-02 1.777e-01
#>  [4,]  6  9.598e-02 1.619e-01
#>  [5,]  6  1.256e-01 1.475e-01
#>  [6,]  6  1.510e-01 1.344e-01
#>  [7,] 10  1.745e-01 1.225e-01
#>  [8,] 13  1.986e-01 1.116e-01
#>  [9,] 16  2.242e-01 1.017e-01
#> [10,] 16  2.492e-01 9.263e-02
#> [11,] 19  2.732e-01 8.440e-02
#> [12,] 21  2.959e-01 7.690e-02
#> [13,] 24  3.181e-01 7.007e-02
#> [14,] 24  3.391e-01 6.385e-02
#> [15,] 29  3.601e-01 5.817e-02
#> [16,] 30  3.797e-01 5.301e-02
#> [17,] 34  3.980e-01 4.830e-02
#> [18,] 35  4.155e-01 4.401e-02
#> [19,] 35  4.310e-01 4.010e-02
#> [20,] 36  4.450e-01 3.653e-02
#> [21,] 37  4.609e-01 3.329e-02
#> [22,] 39  4.771e-01 3.033e-02
#> [23,] 38  4.935e-01 2.764e-02
#> [24,] 45  5.144e-01 2.518e-02
#> [25,] 49  5.383e-01 2.294e-02
#> [26,] 52  5.626e-01 2.091e-02
#> [27,] 57  5.881e-01 1.905e-02
#> [28,] 63  6.146e-01 1.736e-02
#> [29,] 63  6.410e-01 1.582e-02
#> [30,] 65  6.659e-01 1.441e-02
#> [31,] 67  6.911e-01 1.313e-02
#> [32,] 69  7.144e-01 1.196e-02
#> [33,] 69  7.359e-01 1.090e-02
#> [34,] 70  7.558e-01 9.932e-03
#> [35,] 75  7.750e-01 9.050e-03
#> [36,] 80  7.932e-01 8.246e-03
#> [37,] 83  8.103e-01 7.513e-03
#> [38,] 83  8.263e-01 6.846e-03
#> [39,] 84  8.410e-01 6.238e-03
#> [40,] 84  8.544e-01 5.684e-03
#> [41,] 85  8.667e-01 5.179e-03
#> [42,] 88  8.781e-01 4.719e-03
#> [43,] 87  8.885e-01 4.299e-03
#> [44,] 87  8.979e-01 3.918e-03
#> [45,] 89  9.067e-01 3.569e-03
#> [46,] 88  9.148e-01 3.252e-03
#> [47,] 88  9.222e-01 2.963e-03
#> [48,] 88  9.289e-01 2.700e-03
#> [49,] 88  9.351e-01 2.460e-03
#> [50,] 88  9.409e-01 2.242e-03
#> [51,] 91  9.460e-01 2.043e-03
#> [52,] 91  9.509e-01 1.861e-03
#> [53,] 92  9.553e-01 1.696e-03
#> [54,] 93  9.591e-01 1.545e-03
#> [55,] 92  9.628e-01 1.408e-03
#> [56,] 92  9.661e-01 1.283e-03
#> [57,] 95  9.691e-01 1.169e-03
#> [58,] 96  9.718e-01 1.065e-03
#> [59,] 96  9.743e-01 9.704e-04
#> [60,] 96  9.766e-01 8.842e-04
#> [61,] 96  9.787e-01 8.056e-04
#> [62,] 96  9.806e-01 7.341e-04
#> [63,] 94  9.823e-01 6.689e-04
#> [64,] 95  9.839e-01 6.094e-04
#> [65,] 95  9.853e-01 5.553e-04
#> [66,] 95  9.866e-01 5.060e-04
#> [67,] 95  9.878e-01 4.610e-04
#> [68,] 96  9.889e-01 4.201e-04
#> [69,] 96  9.899e-01 3.827e-04
#> [70,] 97  9.907e-01 3.487e-04
#> [71,] 98  9.916e-01 3.178e-04
#> [72,] 98  9.923e-01 2.895e-04
#> [73,] 97  9.930e-01 2.638e-04
#> [74,] 97  9.936e-01 2.404e-04
#> [75,] 97  9.942e-01 2.190e-04
#> [76,] 97  9.947e-01 1.996e-04
#> [77,] 97  9.952e-01 1.818e-04
#> [78,] 97  9.956e-01 1.657e-04
#> [79,] 97  9.960e-01 1.510e-04
#> [80,] 98  9.963e-01 1.376e-04
#> [81,] 98  9.967e-01 1.253e-04
#> [82,] 98  9.970e-01 1.142e-04
#> [83,] 96  9.972e-01 1.041e-04
#> [84,] 97  9.975e-01 9.481e-05
#> [85,] 97  9.977e-01 8.639e-05
#> [86,] 97  9.979e-01 7.871e-05
#> [87,] 97  9.981e-01 7.172e-05
#> [88,] 97  9.983e-01 6.535e-05
#> [89,] 97  9.984e-01 5.954e-05
#> [90,] 97  9.985e-01 5.425e-05
#> [91,] 97  9.987e-01 4.943e-05
#> [92,] 97  9.988e-01 4.504e-05
#> [93,] 97  9.989e-01 4.104e-05
#> [94,] 98  9.990e-01 3.739e-05
#> [95,] 98  9.991e-01 3.407e-05
#> [1] "done bootstrap 1"
sink()


c_selectID <- 3
genes = DEgenes$DE_Subpop3vsRemaining$id[1:200]
genes <- gsub("_.*", "", genes)
sink("temp")
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 0.465900
#>    [2,]  1 0.05569 0.444700
#>    [3,]  1 0.10670 0.424500
#>    [4,]  1 0.15370 0.405200
#>    [5,]  1 0.19710 0.386800
#>    [6,]  1 0.23740 0.369200
#>    [7,]  1 0.27500 0.352400
#>    [8,]  1 0.31000 0.336400
#>    [9,]  1 0.34280 0.321100
#>   [10,]  1 0.37370 0.306500
#>  [11,]  1 0.40260 0.292600
#>  [12,]  1 0.43000 0.279300
#>  [13,]  1 0.45580 0.266600
#>  [14,]  1 0.48010 0.254500
#>  [15,]  1 0.50320 0.242900
#>  [16,]  1 0.52510 0.231900
#>  [17,]  1 0.54580 0.221300
#>  [18,]  1 0.56550 0.211300
#>  [19,]  1 0.58430 0.201700
#>  [20,]  1 0.60210 0.192500
#>  [21,]  1 0.61910 0.183800
#>  [22,]  1 0.63530 0.175400
#>  [23,]  1 0.65070 0.167400
#>  [24,]  1 0.66540 0.159800
#>  [25,]  1 0.67940 0.152600
#>  [26,]  1 0.69280 0.145600
#>  [27,]  1 0.70560 0.139000
#>  [28,]  1 0.71780 0.132700
#>  [29,]  1 0.72950 0.126700
#>  [30,]  1 0.74070 0.120900
#>  [31,]  1 0.75140 0.115400
#>  [32,]  1 0.76160 0.110200
#>  [33,]  1 0.77140 0.105200
#>  [34,]  1 0.78080 0.100400
#>  [35,]  1 0.78980 0.095810
#>  [36,]  1 0.79830 0.091450
#>  [37,]  1 0.80660 0.087300
#>  [38,]  1 0.81440 0.083330
#>  [39,]  1 0.82200 0.079540
#>  [40,]  1 0.82920 0.075930
#>  [41,]  1 0.83620 0.072480
#>  [42,]  1 0.84280 0.069180
#>  [43,]  1 0.84920 0.066040
#>  [44,]  1 0.85530 0.063040
#>  [45,]  2 0.86130 0.060170
#>  [46,]  2 0.86720 0.057440
#>  [47,]  2 0.87280 0.054830
#>  [48,]  2 0.87810 0.052330
#>  [49,]  2 0.88320 0.049950
#>  [50,]  2 0.88810 0.047680
#>  [51,]  2 0.89280 0.045520
#>  [52,]  4 0.89750 0.043450
#>  [53,]  4 0.90200 0.041470
#>  [54,]  5 0.90640 0.039590
#>  [55,]  5 0.91070 0.037790
#>  [56,]  5 0.91470 0.036070
#>  [57,]  5 0.91860 0.034430
#>  [58,]  5 0.92230 0.032870
#>  [59,]  5 0.92580 0.031370
#>  [60,]  5 0.92910 0.029950
#>  [61,]  5 0.93230 0.028590
#>  [62,]  5 0.93540 0.027290
#>  [63,]  5 0.93830 0.026050
#>  [64,]  6 0.94110 0.024860
#>  [65,]  6 0.94380 0.023730
#>  [66,]  6 0.94630 0.022650
#>  [67,]  6 0.94870 0.021620
#>  [68,]  6 0.95100 0.020640
#>  [69,]  7 0.95330 0.019700
#>  [70,]  7 0.95540 0.018810
#>  [71,]  7 0.95740 0.017950
#>  [72,]  7 0.95930 0.017140
#>  [73,]  7 0.96120 0.016360
#>  [74,]  7 0.96290 0.015610
#>  [75,]  7 0.96460 0.014900
#>  [76,]  7 0.96620 0.014230
#>  [77,]  7 0.96780 0.013580
#>  [78,]  7 0.96920 0.012960
#>  [79,]  7 0.97060 0.012370
#>  [80,]  7 0.97190 0.011810
#>  [81,]  7 0.97320 0.011270
#>  [82,]  7 0.97440 0.010760
#>  [83,]  7 0.97560 0.010270
#>  [84,]  7 0.97670 0.009806
#>  [85,]  8 0.97770 0.009361
#>  [86,]  8 0.97870 0.008935
#>  [87,]  8 0.97970 0.008529
#>  [88,]  8 0.98060 0.008141
#>  [89,]  8 0.98150 0.007771
#>  [90,]  8 0.98230 0.007418
#>  [91,]  8 0.98310 0.007081
#>  [92,]  8 0.98390 0.006759
#>  [93,]  8 0.98460 0.006452
#>  [94,]  8 0.98530 0.006159
#>  [95,]  8 0.98590 0.005879
#>  [96,]  8 0.98660 0.005612
#>  [97,]  8 0.98720 0.005357
#>  [98,]  8 0.98780 0.005113
#>  [99,]  8 0.98830 0.004881
#> [100,]  8 0.98880 0.004659
#> [1] "done bootstrap 1"
sink()

#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", Li
                  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
```