

# scGPS introduction

Quan Nguyen

2018-03-21

## Installation instruction

```
#Prior to install 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 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 install 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 compile in R, using devtools
devtools::document()

#and then you can load the package:
devtools::load_all()
```

**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$dat2_counts, GeneMetadata = day2$dat
2geneInfo,
                        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$dat
5geneInfo,
                        CellMetadata = day5$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
suppressWarnings(LSOLDA_dat <- bootstrap_scGPS(nboots = 2,mixedpop1 = mixedpop1, mixe
dpop2 = 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 2.456e-01
#>  [2,]  1  2.972e-02 2.237e-01
#>  [3,]  3  6.442e-02 2.039e-01
#>  [4,]  4  1.056e-01 1.858e-01
#>  [5,]  5  1.458e-01 1.693e-01
#>  [6,]  5  1.822e-01 1.542e-01
#>  [7,]  5  2.139e-01 1.405e-01
#>  [8,]  6  2.417e-01 1.280e-01
#>  [9,]  7  2.700e-01 1.167e-01
#> [10,]  7  2.950e-01 1.063e-01
#> [11,]  7  3.172e-01 9.685e-02
#> [12,]  7  3.369e-01 8.825e-02
#> [13,]  8  3.552e-01 8.041e-02
#> [14,] 10  3.740e-01 7.327e-02
#> [15,] 10  3.913e-01 6.676e-02
#> [16,] 11  4.075e-01 6.083e-02
#> [17,] 13  4.240e-01 5.542e-02
#> [18,] 18  4.439e-01 5.050e-02
#> [19,] 19  4.665e-01 4.601e-02
#> [20,] 19  4.866e-01 4.193e-02
#> [21,] 23  5.077e-01 3.820e-02
#> [22,] 24  5.286e-01 3.481e-02
#> [23,] 26  5.490e-01 3.172e-02
#> [24,] 30  5.685e-01 2.890e-02
#> [25,] 31  5.872e-01 2.633e-02
#> [26,] 33  6.048e-01 2.399e-02
#> [27,] 35  6.217e-01 2.186e-02
#> [28,] 37  6.398e-01 1.992e-02
#> [29,] 39  6.568e-01 1.815e-02
#> [30,] 41  6.734e-01 1.654e-02
#> [31,] 43  6.891e-01 1.507e-02

```

```
#> [32,] 45 7.050e-01 1.373e-02
#> [33,] 45 7.202e-01 1.251e-02
#> [34,] 45 7.342e-01 1.140e-02
#> [35,] 47 7.478e-01 1.039e-02
#> [36,] 48 7.615e-01 9.463e-03
#> [37,] 49 7.744e-01 8.622e-03
#> [38,] 50 7.866e-01 7.856e-03
#> [39,] 55 7.985e-01 7.158e-03
#> [40,] 56 8.100e-01 6.522e-03
#> [41,] 57 8.211e-01 5.943e-03
#> [42,] 57 8.319e-01 5.415e-03
#> [43,] 60 8.419e-01 4.934e-03
#> [44,] 60 8.517e-01 4.496e-03
#> [45,] 60 8.610e-01 4.096e-03
#> [46,] 62 8.702e-01 3.732e-03
#> [47,] 64 8.790e-01 3.401e-03
#> [48,] 65 8.877e-01 3.099e-03
#> [49,] 64 8.960e-01 2.823e-03
#> [50,] 64 9.036e-01 2.573e-03
#> [51,] 64 9.109e-01 2.344e-03
#> [52,] 64 9.177e-01 2.136e-03
#> [53,] 64 9.240e-01 1.946e-03
#> [54,] 65 9.300e-01 1.773e-03
#> [55,] 66 9.357e-01 1.616e-03
#> [56,] 65 9.409e-01 1.472e-03
#> [57,] 65 9.457e-01 1.341e-03
#> [58,] 64 9.501e-01 1.222e-03
#> [59,] 64 9.543e-01 1.114e-03
#> [60,] 63 9.581e-01 1.015e-03
#> [61,] 64 9.617e-01 9.245e-04
#> [62,] 64 9.650e-01 8.424e-04
#> [63,] 65 9.681e-01 7.676e-04
#> [64,] 65 9.708e-01 6.994e-04
#> [65,] 65 9.734e-01 6.372e-04
#> [66,] 65 9.757e-01 5.806e-04
#> [67,] 65 9.778e-01 5.290e-04
#> [68,] 67 9.798e-01 4.820e-04
#> [69,] 67 9.816e-01 4.392e-04
#> [70,] 67 9.832e-01 4.002e-04
#> [71,] 66 9.847e-01 3.647e-04
#> [72,] 65 9.860e-01 3.323e-04
#> [73,] 65 9.873e-01 3.027e-04
#> [74,] 65 9.884e-01 2.758e-04
#> [75,] 65 9.894e-01 2.513e-04
#> [76,] 65 9.903e-01 2.290e-04
#> [77,] 65 9.912e-01 2.087e-04
#> [78,] 66 9.920e-01 1.901e-04
#> [79,] 66 9.927e-01 1.732e-04
#> [80,] 66 9.933e-01 1.578e-04
#> [81,] 67 9.939e-01 1.438e-04
#> [82,] 67 9.945e-01 1.310e-04
#> [83,] 67 9.949e-01 1.194e-04
#> [84,] 67 9.954e-01 1.088e-04
#> [85,] 67 9.958e-01 9.913e-05
#> [86,] 67 9.962e-01 9.033e-05
#> [87,] 67 9.965e-01 8.230e-05
#> [88,] 67 9.968e-01 7.499e-05
#> [89,] 67 9.971e-01 6.833e-05
```

```

#> [90,] 67 9.973e-01 6.226e-05
#> [91,] 67 9.976e-01 5.673e-05
#> [92,] 67 9.978e-01 5.169e-05
#> [93,] 67 9.980e-01 4.710e-05
#> [94,] 67 9.982e-01 4.291e-05
#> [95,] 68 9.983e-01 3.910e-05
#> [96,] 68 9.985e-01 3.563e-05
#> [97,] 68 9.986e-01 3.246e-05
#> [98,] 67 9.987e-01 2.958e-05
#> [99,] 67 9.988e-01 2.695e-05
#> [100,] 67 9.989e-01 2.456e-05
#> [1] "done 1"
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>      Df      %Dev   Lambda
#> [1,]  0 -2.563e-15 2.581e-01
#> [2,]  2  3.383e-02 2.351e-01
#> [3,]  2  7.253e-02 2.143e-01
#> [4,]  2  1.056e-01 1.952e-01
#> [5,]  2  1.343e-01 1.779e-01
#> [6,]  4  1.654e-01 1.621e-01
#> [7,]  4  1.973e-01 1.477e-01
#> [8,]  4  2.253e-01 1.346e-01
#> [9,]  4  2.500e-01 1.226e-01
#> [10,] 5  2.753e-01 1.117e-01
#> [11,] 6  2.987e-01 1.018e-01
#> [12,] 6  3.196e-01 9.274e-02
#> [13,] 6  3.381e-01 8.451e-02
#> [14,] 6  3.546e-01 7.700e-02
#> [15,] 8  3.701e-01 7.016e-02
#> [16,] 10 3.876e-01 6.393e-02
#> [17,] 10 4.036e-01 5.825e-02
#> [18,] 12 4.198e-01 5.307e-02
#> [19,] 13 4.365e-01 4.836e-02
#> [20,] 15 4.540e-01 4.406e-02
#> [21,] 19 4.713e-01 4.015e-02
#> [22,] 23 4.906e-01 3.658e-02
#> [23,] 24 5.098e-01 3.333e-02
#> [24,] 26 5.284e-01 3.037e-02
#> [25,] 27 5.455e-01 2.767e-02
#> [26,] 29 5.612e-01 2.521e-02
#> [27,] 32 5.759e-01 2.297e-02
#> [28,] 36 5.903e-01 2.093e-02
#> [29,] 39 6.050e-01 1.907e-02
#> [30,] 41 6.190e-01 1.738e-02
#> [31,] 44 6.325e-01 1.583e-02
#> [32,] 46 6.455e-01 1.443e-02
#> [33,] 48 6.582e-01 1.315e-02
#> [34,] 53 6.709e-01 1.198e-02
#> [35,] 55 6.839e-01 1.091e-02
#> [36,] 56 6.959e-01 9.945e-03
#> [37,] 60 7.079e-01 9.061e-03
#> [38,] 60 7.200e-01 8.256e-03
#> [39,] 62 7.314e-01 7.523e-03
#> [40,] 63 7.424e-01 6.855e-03
#> [41,] 63 7.527e-01 6.246e-03
#> [42,] 65 7.624e-01 5.691e-03

```

```
#> [43,] 65 7.718e-01 5.185e-03
#> [44,] 68 7.813e-01 4.725e-03
#> [45,] 68 7.907e-01 4.305e-03
#> [46,] 69 7.998e-01 3.922e-03
#> [47,] 69 8.089e-01 3.574e-03
#> [48,] 67 8.176e-01 3.256e-03
#> [49,] 67 8.262e-01 2.967e-03
#> [50,] 67 8.345e-01 2.704e-03
#> [51,] 68 8.434e-01 2.463e-03
#> [52,] 69 8.521e-01 2.245e-03
#> [53,] 71 8.608e-01 2.045e-03
#> [54,] 73 8.698e-01 1.863e-03
#> [55,] 74 8.785e-01 1.698e-03
#> [56,] 72 8.869e-01 1.547e-03
#> [57,] 74 8.949e-01 1.410e-03
#> [58,] 74 9.028e-01 1.284e-03
#> [59,] 74 9.098e-01 1.170e-03
#> [60,] 75 9.168e-01 1.066e-03
#> [61,] 75 9.231e-01 9.716e-04
#> [62,] 76 9.297e-01 8.853e-04
#> [63,] 77 9.355e-01 8.066e-04
#> [64,] 77 9.412e-01 7.350e-04
#> [65,] 76 9.466e-01 6.697e-04
#> [66,] 75 9.515e-01 6.102e-04
#> [67,] 75 9.559e-01 5.560e-04
#> [68,] 75 9.599e-01 5.066e-04
#> [69,] 76 9.636e-01 4.616e-04
#> [70,] 76 9.669e-01 4.206e-04
#> [71,] 76 9.699e-01 3.832e-04
#> [72,] 75 9.727e-01 3.492e-04
#> [73,] 75 9.752e-01 3.182e-04
#> [74,] 76 9.774e-01 2.899e-04
#> [75,] 75 9.795e-01 2.641e-04
#> [76,] 74 9.814e-01 2.407e-04
#> [77,] 74 9.830e-01 2.193e-04
#> [78,] 74 9.846e-01 1.998e-04
#> [79,] 74 9.859e-01 1.821e-04
#> [80,] 74 9.872e-01 1.659e-04
#> [81,] 74 9.883e-01 1.512e-04
#> [82,] 74 9.894e-01 1.377e-04
#> [83,] 74 9.903e-01 1.255e-04
#> [84,] 74 9.912e-01 1.143e-04
#> [85,] 74 9.920e-01 1.042e-04
#> [86,] 74 9.927e-01 9.493e-05
#> [87,] 74 9.933e-01 8.649e-05
#> [88,] 74 9.939e-01 7.881e-05
#> [89,] 74 9.945e-01 7.181e-05
#> [90,] 73 9.949e-01 6.543e-05
#> [91,] 73 9.954e-01 5.962e-05
#> [92,] 73 9.958e-01 5.432e-05
#> [93,] 73 9.962e-01 4.950e-05
#> [94,] 73 9.965e-01 4.510e-05
#> [95,] 73 9.968e-01 4.109e-05
#> [96,] 73 9.971e-01 3.744e-05
#> [97,] 73 9.973e-01 3.412e-05
#> [98,] 73 9.976e-01 3.108e-05
#> [99,] 73 9.978e-01 2.832e-05
#> [100,] 73 9.980e-01 2.581e-05
```

```
#> [1] "done 2"

#display some results
names(LSOLDA_dat)
#> [1] "Accuracy"      "LassoGenes"    "Deviance"      "LassoFit"
#> [5] "LDAFit"        "predictor_S1"  "LassoPredict"  "LDAPredict"
LSOLDA_dat$LassoPredict
#> [[1]]
#> [[1]][[1]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[1]][[2]]
#> [1] 99.2674
#>
#> [[1]][[3]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 89.43089
#>
#> [[1]][[5]]
#> [1] "LASSO for subpop1 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 96.72131
#>
#> [[1]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 88.77551
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LASSO for subpop2 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 95.9707
#>
#> [[2]][[3]]
#> [1] "LASSO for subpop3 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 74.39024
#>
#> [[2]][[5]]
#> [1] "LASSO for subpop1 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 88.52459
#>
#> [[2]][[7]]
#> [1] "LASSO for subpop4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 85.71429
LSOLDA_dat$LDAPredict
```

```

#> [[1]]
#> [[1]][[1]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[1]][[2]]
#> [1] 72.16117
#>
#> [[1]][[3]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[1]][[4]]
#> [1] 21.54472
#>
#> [[1]][[5]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[1]][[6]]
#> [1] 26.50273
#>
#> [[1]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[1]][[8]]
#> [1] 26.53061
#>
#>
#> [[2]]
#> [[2]][[1]]
#> [1] "LDA for subpop 2 in target mixedpop2"
#>
#> [[2]][[2]]
#> [1] 48.71795
#>
#> [[2]][[3]]
#> [1] "LDA for subpop 3 in target mixedpop2"
#>
#> [[2]][[4]]
#> [1] 11.38211
#>
#> [[2]][[5]]
#> [1] "LDA for subpop 1 in target mixedpop2"
#>
#> [[2]][[6]]
#> [1] 2.459016
#>
#> [[2]][[7]]
#> [1] "LDA for subpop 4 in target mixedpop2"
#>
#> [[2]][[8]]
#> [1] 14.28571

#summary results LDA
summary_prediction_lda(LSOLDA_dat=LSOLDA_dat, nPredSubpop = 4)
#>
#>          V1          V2          names
#> 1 72.1611721611722 48.7179487179487 LDA for subpop 2 in target mixedpop2
#> 2 21.5447154471545 11.3821138211382 LDA for subpop 3 in target mixedpop2
#> 3 26.5027322404372 2.45901639344262 LDA for subpop 1 in target mixedpop2
#> 4 26.530612244898 14.2857142857143 LDA for subpop 4 in target mixedpop2

```

```

#summary results Lasso
summary_prediction_lasso(LSOLDA_dat=LSOLDA_dat, nPredSubpop = 4)
#>
#>      V1      V2      names
#> 1 99.2673992673993 95.970695970696 LASSO for subpop2 in target mixedpop2
#> 2 89.4308943089431 74.390243902439 LASSO for subpop3 in target mixedpop2
#> 3 96.7213114754098 88.5245901639344 LASSO for subpop1 in target mixedpop2
#> 4 88.7755102040816 85.7142857142857 LASSO for subpop4 in target mixedpop2

#summary deviance
summary_deviance(LSOLDA_dat)
#> $allDeviance
#> [1] "0.3172" "0.3381"
#>
#> $DeviMax
#>      Dfd      Deviance      DEgenes
#> 1      0 -2.563e-15 genes_cluster1
#> 2      2      0.1343 genes_cluster1
#> 3      4      0.25 genes_cluster1
#> 4      5      0.2753 genes_cluster1
#> 5      6      0.3381 genes_cluster1
#> 6 remaining      1      DEgenes
#>
#> $LassoGenesMax
#>
#>      1      name
#> (Intercept)      0.08193993      (Intercept)
#> FN1_ENSG00000115414 -0.04457009 FN1_ENSG00000115414
#> FOXC1_ENSG00000054598 0.02843923 FOXC1_ENSG00000054598
#> T_ENSG00000164458      0.13784303      T_ENSG00000164458
#> SOX17_ENSG00000164736 -0.03576542 SOX17_ENSG00000164736
#> MESP1_ENSG00000166823 0.03616270 MESP1_ENSG00000166823
#> SNAI1_ENSG00000124216 0.14396048 SNAI1_ENSG00000124216

```

**A complete workflow of the scGPS: given an unknown mixed population, find clusters and estimate relationship between clusters**



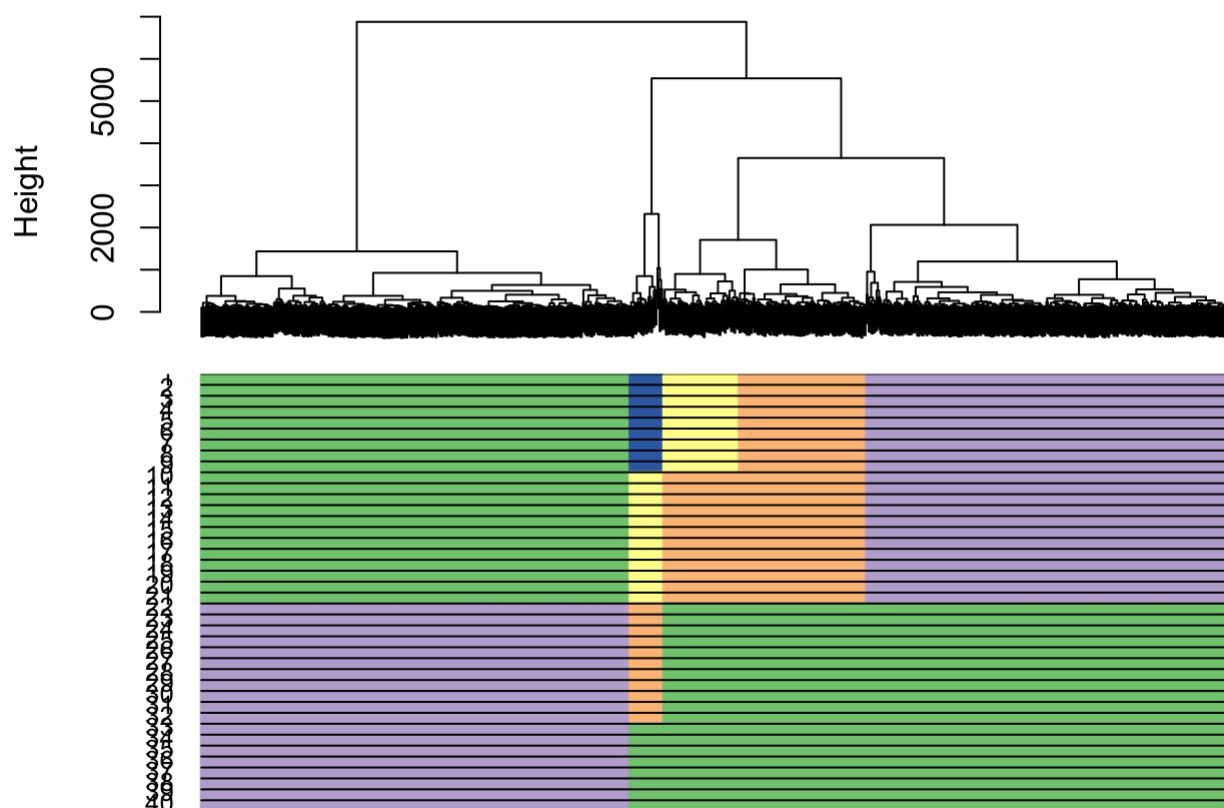
```
#given a single cell expression matrix, without clustering information
day5 <- sample2
cellnames <- colnames(day5$dat5_counts)
cluster <- day5$dat5_clusters
cellnames <- data.frame("Cluster"=cluster, "cellBarcodes" = cellnames)

mixedpop2 <- NewscGPS_SME(ExpressionMatrix = day5$dat5_counts, GeneMetadata = day5$dat
5geneInfo, CellMetadata = cellnames )

#let's find the clusters
CORE_cluster <- CORE_scGPS(mixedpop2, remove_outlier = c(0))
#> [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)
```

## Cluster Dendrogram



```

#let's plot just the optimal clustering result (with colored dendrogram)
optimal_index = which(CORE_cluster$optimalClust$KeyStats$Height == CORE_cluster$optimalClust$OptimalRes)
dev.off()
#> null device
#> 1
plot_optimal_CORE(original_tree= CORE_cluster$tree, optimal_cluster = unlist(CORE_cluster$Cluster[optimal_index]), shift = -20)
#> [1] "Ordering and assigning labels..."
#> [1] 2
#> [1] 204 424 NA NA
#> [1] 3
#> [1] 204 424 536 NA
#> [1] 4
#> [1] 204 424 536 808
#> [1] "Plotting the colored dendrogram now...."
#> [1] "Plotting the bar underneath now...."
#> [1] "Users are required to check cluster labels...."

#let's compare with other dimensional reduction methods
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(mixedpop2)[,1])))
#may need to turn off the R graphic device dev.off() before plotting the following
plot(p2)

#load gene list (this can be any lists of user selected genes)

genes <-GeneList
genes <-genes$Merged_unique

#the gene list can also be generated objectively by differential expression analysis
#if the cluster information is in the mixedpop2 object, run this (if not run the CORE)
#as described below
DEgenes <- findMarkers_scGPS(expression_matrix=assay(mixedpop2), cluster = mixedpop2@colData$Cluster)
#> [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..."
#> [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 4..."
#> [1] "Done estimate dispersions. Start nbinom test for cluster 4..."
#> [1] "Done nbinom test for cluster 4 ..."
#> [1] "Adjust foldchange by subtracting basemean to 1..."
names(DEgenes)
#> [1] "DE_Subpop2vsRemaining" "DE_Subpop3vsRemaining" "DE_Subpop1vsRemaining"
#> [4] "DE_Subpop4vsRemaining"

#users need to check the format of the gene input to make sure they are consistent to

#the gene names in the expression matrix

#add the CORE cluster information into the scGPS object
Optimal_index <- which( CORE_cluster$OptimalClust$KeyStats$Height == CORE_cluster$OptimalClust$OptimalRes)
mixedpop2@colData$Cluster <- unlist(CORE_cluster$Cluster[[Optimal_index]])
#select a subpopulation
c_selectID <- 1

#run the test bootstrap

suppressWarnings(LSOLDA_dat <- bootstrap_scGPS(nboots = 2,mixedpop1 = mixedpop2, mixedpop2 = mixedpop2, genes=genes, c_selectID, listData =list()))
#>
#> Call:  glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>           Df          %Dev      Lambda
#>  [1,]      0 -2.883e-15  2.720e-01
#>  [2,]      2  4.262e-02  2.478e-01
#>  [3,]      2  8.682e-02  2.258e-01
#>  [4,]      2  1.249e-01  2.057e-01
#>  [5,]      2  1.581e-01  1.875e-01
#>  [6,]      3  1.876e-01  1.708e-01
#>  [7,]      3  2.141e-01  1.556e-01
#>  [8,]      4  2.387e-01  1.418e-01
#>  [9,]      4  2.608e-01  1.292e-01
#> [10,]      4  2.803e-01  1.177e-01
#> [11,]      5  2.977e-01  1.073e-01
#> [12,]      5  3.137e-01  9.774e-02
#> [13,]      5  3.278e-01  8.906e-02
#> [14,]      6  3.411e-01  8.115e-02
#> [15,]      6  3.538e-01  7.394e-02
#> [16,]      7  3.651e-01  6.737e-02
#> [17,]     10  3.770e-01  6.138e-02
#> [18,]     11  3.886e-01  5.593e-02
#> [19,]     14  3.992e-01  5.096e-02
#> [20,]     16  4.106e-01  4.643e-02
#> [21,]     18  4.232e-01  4.231e-02
#> [22,]     22  4.362e-01  3.855e-02
#> [23,]     26  4.499e-01  3.513e-02
#> [24,]     27  4.634e-01  3.201e-02
#> [25,]     28  4.756e-01  2.916e-02
#> [26,]     31  4.873e-01  2.657e-02
#> [27,]     36  5.004e-01  2.421e-02
#> [28,]     37  5.138e-01  2.206e-02
#> [29,]     39  5.266e-01  2.010e-02
#> [30,]     41  5.389e-01  1.831e-02

```

```
#> [31,] 43 5.507e-01 1.669e-02
#> [32,] 46 5.625e-01 1.521e-02
#> [33,] 47 5.742e-01 1.385e-02
#> [34,] 48 5.847e-01 1.262e-02
#> [35,] 53 5.951e-01 1.150e-02
#> [36,] 54 6.053e-01 1.048e-02
#> [37,] 56 6.147e-01 9.549e-03
#> [38,] 59 6.239e-01 8.701e-03
#> [39,] 61 6.324e-01 7.928e-03
#> [40,] 64 6.412e-01 7.224e-03
#> [41,] 70 6.497e-01 6.582e-03
#> [42,] 71 6.586e-01 5.997e-03
#> [43,] 74 6.671e-01 5.464e-03
#> [44,] 74 6.750e-01 4.979e-03
#> [45,] 75 6.824e-01 4.537e-03
#> [46,] 78 6.894e-01 4.134e-03
#> [47,] 79 6.959e-01 3.766e-03
#> [48,] 79 7.018e-01 3.432e-03
#> [49,] 81 7.074e-01 3.127e-03
#> [50,] 83 7.125e-01 2.849e-03
#> [51,] 84 7.173e-01 2.596e-03
#> [52,] 86 7.218e-01 2.365e-03
#> [53,] 89 7.268e-01 2.155e-03
#> [54,] 91 7.312e-01 1.964e-03
#> [55,] 91 7.351e-01 1.789e-03
#> [56,] 96 7.388e-01 1.630e-03
#> [57,] 98 7.423e-01 1.486e-03
#> [58,] 98 7.455e-01 1.354e-03
#> [59,] 99 7.486e-01 1.233e-03
#> [60,] 101 7.514e-01 1.124e-03
#> [61,] 103 7.550e-01 1.024e-03
#> [62,] 103 7.582e-01 9.330e-04
#> [63,] 103 7.611e-01 8.501e-04
#> [64,] 105 7.637e-01 7.746e-04
#> [65,] 106 7.663e-01 7.058e-04
#> [66,] 107 7.687e-01 6.431e-04
#> [67,] 107 7.712e-01 5.859e-04
#> [68,] 106 7.743e-01 5.339e-04
#> [69,] 108 7.781e-01 4.865e-04
#> [70,] 109 7.820e-01 4.432e-04
#> [71,] 110 7.859e-01 4.039e-04
#> [72,] 110 7.895e-01 3.680e-04
#> [73,] 109 7.927e-01 3.353e-04
#> [74,] 111 7.958e-01 3.055e-04
#> [75,] 111 7.988e-01 2.784e-04
#> [76,] 111 8.016e-01 2.536e-04
#> [77,] 112 8.043e-01 2.311e-04
#> [78,] 111 8.068e-01 2.106e-04
#> [79,] 112 8.092e-01 1.919e-04
#> [80,] 110 8.115e-01 1.748e-04
#> [81,] 110 8.138e-01 1.593e-04
#> [82,] 110 8.158e-01 1.451e-04
#> [83,] 109 8.177e-01 1.322e-04
#> [84,] 109 8.195e-01 1.205e-04
#> [85,] 109 8.211e-01 1.098e-04
#> [86,] 108 8.227e-01 1.000e-04
#> [87,] 109 8.241e-01 9.115e-05
#> [88,] 109 8.254e-01 8.305e-05
```

```

#> [89,] 109 8.267e-01 7.568e-05
#> [90,] 109 8.279e-01 6.895e-05
#> [91,] 109 8.290e-01 6.283e-05
#> [92,] 109 8.301e-01 5.725e-05
#> [93,] 109 8.312e-01 5.216e-05
#> [94,] 110 8.322e-01 4.753e-05
#> [95,] 109 8.331e-01 4.330e-05
#> [96,] 109 8.341e-01 3.946e-05
#> [97,] 109 8.350e-01 3.595e-05
#> [98,] 110 8.360e-01 3.276e-05
#> [99,] 111 8.369e-01 2.985e-05
#> [100,] 111 8.378e-01 2.720e-05
#> [1] "done 1"
#>
#> Call: glmnet(x = t(predictor_S1), y = y_cat, family = "binomial")
#>
#>           Df      %Dev   Lambda
#> [1,]      0 -2.883e-15 2.901e-01
#> [2,]      2  4.174e-02 2.644e-01
#> [3,]      2  9.127e-02 2.409e-01
#> [4,]      2  1.337e-01 2.195e-01
#> [5,]      3  1.718e-01 2.000e-01
#> [6,]      3  2.062e-01 1.822e-01
#> [7,]      4  2.390e-01 1.660e-01
#> [8,]      4  2.680e-01 1.513e-01
#> [9,]      4  2.937e-01 1.378e-01
#> [10,]     4  3.164e-01 1.256e-01
#> [11,]     4  3.365e-01 1.144e-01
#> [12,]     4  3.544e-01 1.043e-01
#> [13,]     4  3.702e-01 9.501e-02
#> [14,]     4  3.842e-01 8.657e-02
#> [15,]     4  3.966e-01 7.888e-02
#> [16,]     4  4.076e-01 7.187e-02
#> [17,]     6  4.208e-01 6.548e-02
#> [18,]     6  4.357e-01 5.967e-02
#> [19,]     7  4.494e-01 5.437e-02
#> [20,]     7  4.618e-01 4.954e-02
#> [21,]     7  4.727e-01 4.514e-02
#> [22,]    11  4.839e-01 4.113e-02
#> [23,]    13  4.959e-01 3.747e-02
#> [24,]    18  5.073e-01 3.414e-02
#> [25,]    21  5.203e-01 3.111e-02
#> [26,]    22  5.324e-01 2.835e-02
#> [27,]    24  5.435e-01 2.583e-02
#> [28,]    28  5.550e-01 2.353e-02
#> [29,]    31  5.657e-01 2.144e-02
#> [30,]    32  5.759e-01 1.954e-02
#> [31,]    34  5.850e-01 1.780e-02
#> [32,]    38  5.937e-01 1.622e-02
#> [33,]    42  6.031e-01 1.478e-02
#> [34,]    46  6.121e-01 1.347e-02
#> [35,]    53  6.223e-01 1.227e-02
#> [36,]    59  6.336e-01 1.118e-02
#> [37,]    63  6.454e-01 1.019e-02
#> [38,]    66  6.570e-01 9.282e-03
#> [39,]    69  6.674e-01 8.458e-03
#> [40,]    74  6.777e-01 7.706e-03
#> [41,]    76  6.877e-01 7.022e-03

```

```
#> [42,] 77 6.967e-01 6.398e-03
#> [43,] 79 7.052e-01 5.830e-03
#> [44,] 81 7.134e-01 5.312e-03
#> [45,] 83 7.212e-01 4.840e-03
#> [46,] 84 7.285e-01 4.410e-03
#> [47,] 83 7.351e-01 4.018e-03
#> [48,] 86 7.416e-01 3.661e-03
#> [49,] 89 7.481e-01 3.336e-03
#> [50,] 88 7.539e-01 3.040e-03
#> [51,] 85 7.593e-01 2.769e-03
#> [52,] 86 7.642e-01 2.523e-03
#> [53,] 87 7.689e-01 2.299e-03
#> [54,] 88 7.731e-01 2.095e-03
#> [55,] 92 7.773e-01 1.909e-03
#> [56,] 93 7.812e-01 1.739e-03
#> [57,] 94 7.854e-01 1.585e-03
#> [58,] 97 7.900e-01 1.444e-03
#> [59,] 99 7.944e-01 1.316e-03
#> [60,] 101 7.989e-01 1.199e-03
#> [61,] 102 8.030e-01 1.092e-03
#> [62,] 100 8.071e-01 9.953e-04
#> [63,] 102 8.110e-01 9.069e-04
#> [64,] 107 8.147e-01 8.263e-04
#> [65,] 107 8.186e-01 7.529e-04
#> [66,] 108 8.224e-01 6.860e-04
#> [67,] 110 8.261e-01 6.251e-04
#> [68,] 110 8.302e-01 5.695e-04
#> [69,] 110 8.342e-01 5.190e-04
#> [70,] 110 8.389e-01 4.728e-04
#> [71,] 110 8.435e-01 4.308e-04
#> [72,] 109 8.479e-01 3.926e-04
#> [73,] 110 8.522e-01 3.577e-04
#> [74,] 109 8.565e-01 3.259e-04
#> [75,] 109 8.612e-01 2.970e-04
#> [76,] 110 8.658e-01 2.706e-04
#> [77,] 109 8.704e-01 2.465e-04
#> [78,] 110 8.753e-01 2.246e-04
#> [79,] 109 8.806e-01 2.047e-04
#> [80,] 110 8.860e-01 1.865e-04
#> [81,] 109 8.917e-01 1.699e-04
#> [82,] 111 8.977e-01 1.548e-04
#> [83,] 111 9.041e-01 1.411e-04
#> [84,] 111 9.107e-01 1.285e-04
#> [85,] 111 9.173e-01 1.171e-04
#> [86,] 111 9.239e-01 1.067e-04
#> [87,] 111 9.303e-01 9.724e-05
#> [88,] 111 9.366e-01 8.860e-05
#> [89,] 109 9.424e-01 8.073e-05
#> [90,] 110 9.477e-01 7.356e-05
#> [91,] 109 9.527e-01 6.703e-05
#> [92,] 108 9.571e-01 6.107e-05
#> [93,] 108 9.610e-01 5.565e-05
#> [94,] 108 9.647e-01 5.070e-05
#> [95,] 108 9.679e-01 4.620e-05
#> [96,] 109 9.709e-01 4.209e-05
#> [97,] 108 9.735e-01 3.835e-05
#> [98,] 107 9.759e-01 3.495e-05
#> [99,] 107 9.781e-01 3.184e-05
```

```

#> [100,] 108 9.800e-01 2.901e-05
#> [1] "done 2"

#display some results
#summary results LDA
row_cluster <-length(unique(mixedpop2@colData$Cluster))
summary_prediction_lda(LSOLDA_dat=LSOLDA_dat, nPredSubpop = row_cluster )
#>
#>          V1          V2          names
#> 1 82.3529411764706 82.3529411764706 LDA for subpop 1 in target mixedpop2
#> 2 14.2857142857143 16.2857142857143 LDA for subpop 2 in target mixedpop2
#> 3 12.9533678756477 9.32642487046632 LDA for subpop 3 in target mixedpop2
#> 4          25          25 LDA for subpop 4 in target mixedpop2

#summary results Lasso
summary_prediction_lasso(LSOLDA_dat=LSOLDA_dat, nPredSubpop = row_cluster)
#>
#>          V1          V2          names
#> 1 83.578431372549 83.0882352941177 LASSO for subpop1 in target mixedpop2
#> 2 21.7142857142857 17.7142857142857 LASSO for subpop2 in target mixedpop2
#> 3 14.5077720207254 10.3626943005181 LASSO for subpop3 in target mixedpop2
#> 4          43.75          31.25 LASSO for subpop4 in target mixedpop2

#summary deviance
summary_deviance(LSOLDA_dat)
#> $allDeviance
#> [1] "0.377" "0.5759"
#>
#> $DeviMax
#>
#>      Dfd   Deviance    DEgenes
#> 1      0 -2.883e-15 genes_cluster1
#> 2      2  0.1337 genes_cluster1
#> 3      3  0.2062 genes_cluster1
#> 4      4  0.4076 genes_cluster1
#> 5      6  0.4357 genes_cluster1
#> 6      7  0.4727 genes_cluster1
#> 7     11  0.4839 genes_cluster1
#> 8     13  0.4959 genes_cluster1
#> 9     18  0.5073 genes_cluster1
#> 10    21  0.5203 genes_cluster1
#> 11    22  0.5324 genes_cluster1
#> 12    24  0.5435 genes_cluster1
#> 13    28  0.555 genes_cluster1
#> 14    31  0.5657 genes_cluster1
#> 15    32  0.5759 genes_cluster1
#> 16 remaining      1      DEgenes
#>
#> $LassoGenesMax
#>
#>          1          name
#> (Intercept) -1.3024127796 (Intercept)
#> ABCA4_ENSG00000198691 -0.2816903163 ABCA4_ENSG00000198691
#> OLFM3_ENSG00000118733  0.7813331138 OLFM3_ENSG00000118733
#> TPM3_ENSG00000143549 -0.0102583051 TPM3_ENSG00000143549
#> NR5A2_ENSG00000116833  0.2114527765 NR5A2_ENSG00000116833
#> TNNI1_ENSG00000159173  0.0428508610 TNNI1_ENSG00000159173
#> EOMES_ENSG00000163508 -0.5867072711 EOMES_ENSG00000163508
#> PDGFRA_ENSG00000134853  0.0921765246 PDGFRA_ENSG00000134853
#> KDR_ENSG00000128052  0.1182486931 KDR_ENSG00000128052
#> HAND2_ENSG00000164107  0.0223184138 HAND2_ENSG00000164107
#> IRX2_ENSG00000170561  0.0245239884 IRX2_ENSG00000170561

```



```
#> ISL1_ENSG00000016082 0.0692959293 ISL1_ENSG00000016082
#> MEF2C_ENSG00000081189 0.0355677004 MEF2C_ENSG00000081189
#> HAND1_ENSG000000113196 0.0090977834 HAND1_ENSG000000113196
#> JARID2_ENSG00000008083 0.0759773290 JARID2_ENSG00000008083
#> POU5F1_ENSG000000204531 -0.0419114084 POU5F1_ENSG000000204531
#> TBX20_ENSG000000164532 -0.7725852121 TBX20_ENSG000000164532
#> SOX17_ENSG000000164736 -0.0538974989 SOX17_ENSG000000164736
#> MYC_ENSG000000136997 -0.1291894685 MYC_ENSG000000136997
#> ZBTB16_ENSG000000109906 -0.0919341102 ZBTB16_ENSG000000109906
#> VIM_ENSG00000026025 0.0265021543 VIM_ENSG00000026025
#> HHEX_ENSG000000152804 -0.3511758561 HHEX_ENSG000000152804
#> CD9_ENSG00000010278 -0.6882941587 CD9_ENSG00000010278
#> TBX3_ENSG000000135111 -0.0604726906 TBX3_ENSG000000135111
#> ACTC1_ENSG000000159251 0.0005465365 ACTC1_ENSG000000159251
#> MEIS2_ENSG000000134138 0.1286935397 MEIS2_ENSG000000134138
#> MYLK3_ENSG000000140795 0.1505284692 MYLK3_ENSG000000140795
#> FOXF1_ENSG000000103241 -2.3477950292 FOXF1_ENSG000000103241
#> MYL4_ENSG000000198336 0.0483712309 MYL4_ENSG000000198336
#> PRKCA_ENSG000000154229 0.0801770389 PRKCA_ENSG000000154229
#> SNAI1_ENSG000000124216 0.2604766991 SNAI1_ENSG000000124216
#> FOXA3_ENSG000000170608 -0.4534582121 FOXA3_ENSG000000170608
#> TNNT3_ENSG000000129991 -0.1814444365 TNNT3_ENSG000000129991
```