ATACCoGAPS Tutorial

ATACCoGAPS can be installed from GitHub

```
devtools::install_github("FertigLab/ATACCoGAPS")
```

Attach the ATACCoGAPS package, which attaches CoGAPS as a dependency

```
library(ATACCoGAPS)
```

to outline the ATACCoGAPS pipeline, will use as an example data set single-cell ATAC sequencing data published by Schep et al, 2017. The data was downloaded from GEO accession number GSE99172 and preprocessed using dataSubsetBySparsity() to remove cells and peaks with more than 99% sparsity.

```
data("schepFilteredData")
data("schepCelltypes")
data("schepFilteredPeaks")
```

We use these data to set the hyperparameters of the CoGAPS algorithm. Here we use 7 patterns and 10000 iterations of the algorithm. We use the singleCell and sparseOptimization methods as our data are sparse single-cell data. We run the algorithm distributed across the genome since we have more genomic features than cells (if it was the opposite we would set the distributed pattern to "single-cell"). We then input the peak and cell type information to be returned as part of our result object. Finally, we set distributed parameters so the algorithm will run in parallel across 9 cores.

```
params <- CogapsParams(nPatterns=7, nIterations=10000, seed=42, singleCell=TRUE, sparseOptimization=TRU
params <- setDistributedParams(params, nSets=9)
```

```
## setting distributed parameters - call this again if you change nPatterns
```

```
params
```

```
## -- Standard Parameters --
## nPatterns            7
## nIterations          10000
## seed                 42
## singleCell           TRUE
## sparseOptimization   TRUE
## distributed          genome-wide
##
## -- Sparsity Parameters --
## alpha          0.01
## maxGibbsMass   100
##
## -- Distributed CoGAPS Parameters --
## nSets          9
## cut            7
## minNS          5
## maxNS          14
##
```

```
## 90300 gene names provided
## first gene name: chr1-237588-238087
##
## 1392 sample names provided
## first sample name: Fibroblasts
```

We now call CoGAPS via the R function. CoGAPS is a Bayesian Non-Negative Matrix Factorization algorithm (Fertig et al, 2010). It factorizes count matrices sequencing data and returns patterns which distinguish both features and samples, allowing for the discovery of regulatory differences between samples. In the case of scATAC our features are usually peaks and our samples are indvidual cells.

It is generally not recommended to run CoGAPS locally as it usually requires at least 3 hours for most single-cell data sets, even on powerful servers. The code below is used only as an example and to speed up the generation of this document, will not be run here. We will instead use the example CoGAPS result included in the package which was run on the Batch servers of the AWS cloud.

```r
cogapsResult <- GWCoGAPS(data = schepFilteredData, params = params, nThreads = 9)
```

Loading in the pre-computed CoGAPS result

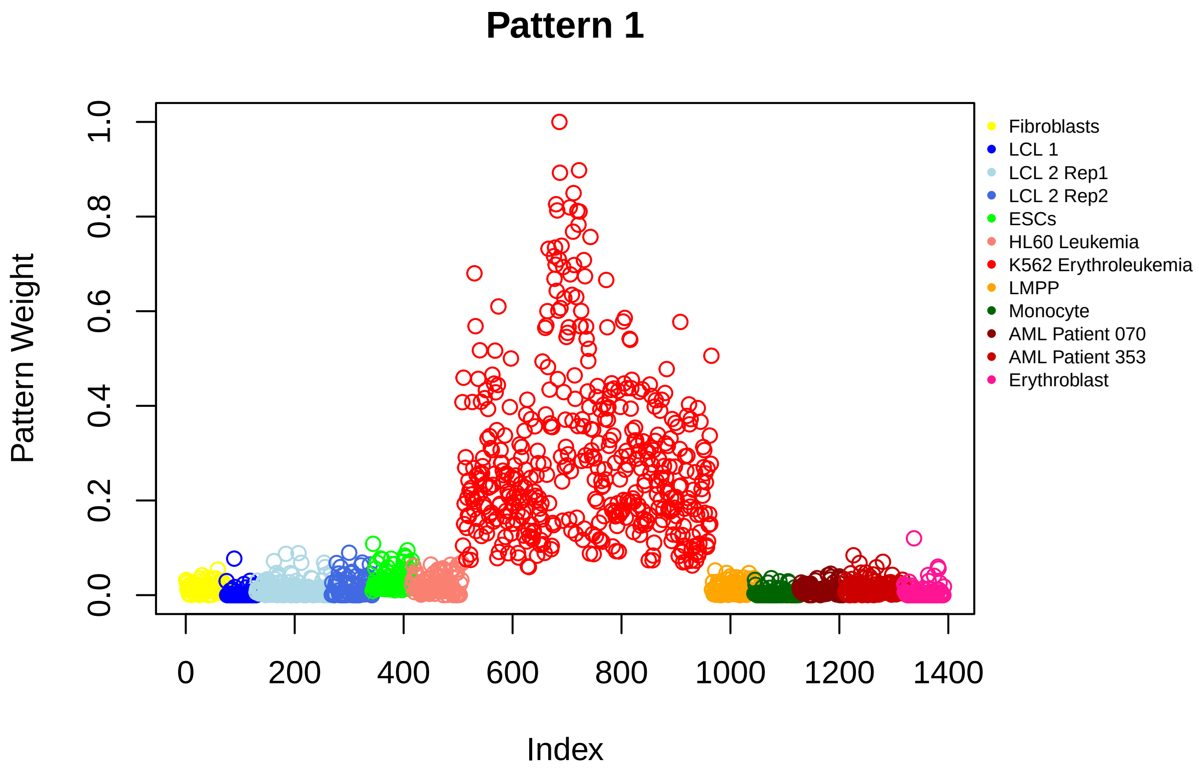
```r
data("schepCogapsResult")
```

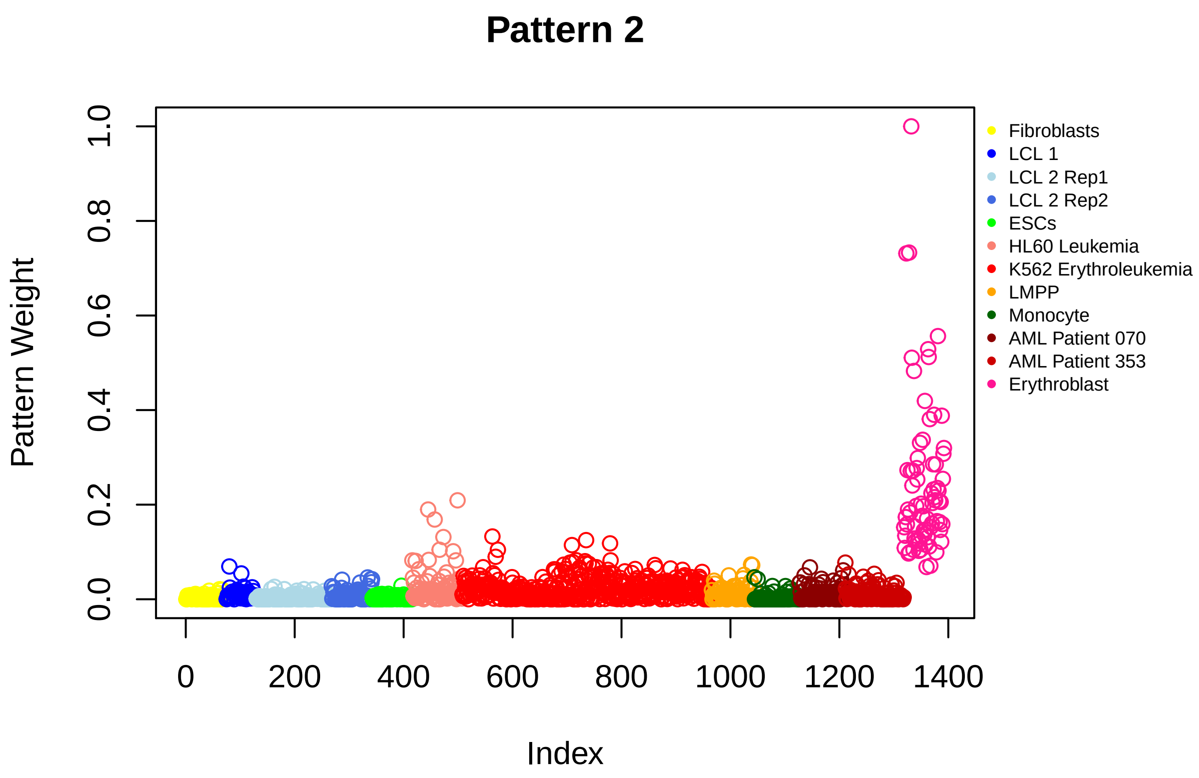
Pattern Matrix Visualization

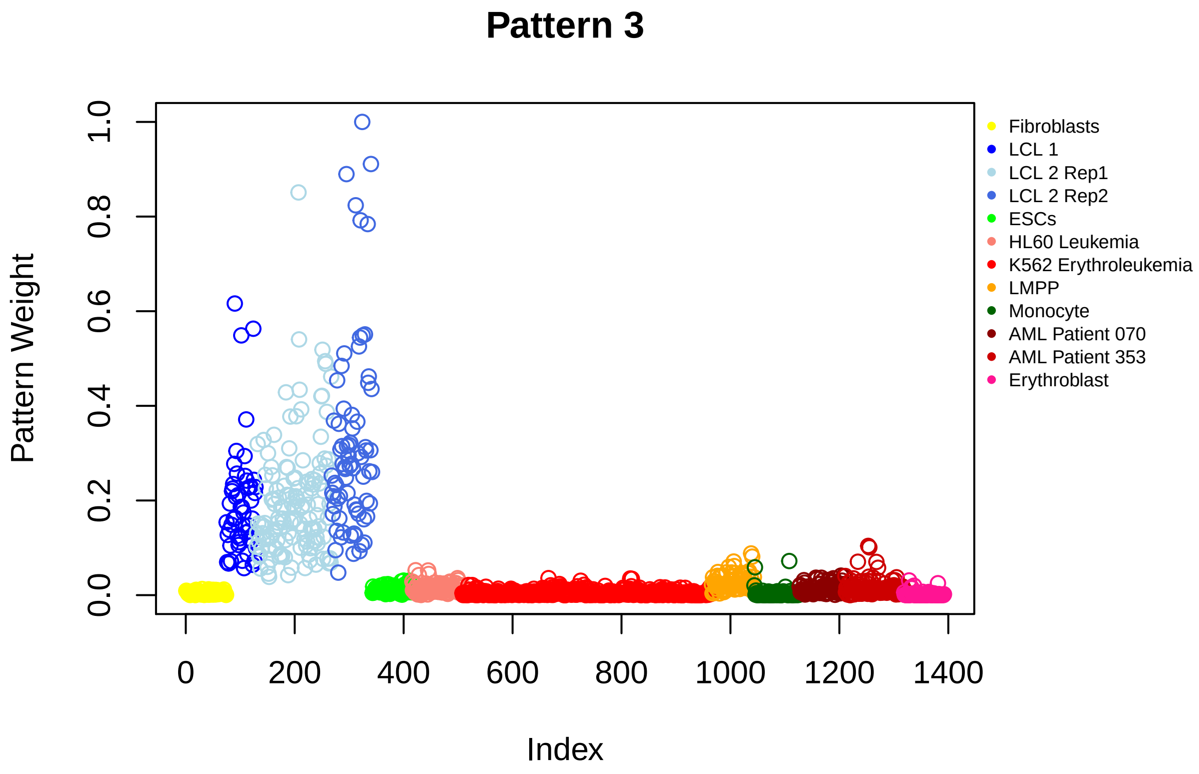
The first quick visualization of CoGAPS results is generally plotting the Pattern Matrix (the output matrix which is patterns x cells). These plots allow us to determine which patterns differentiate which cell types.

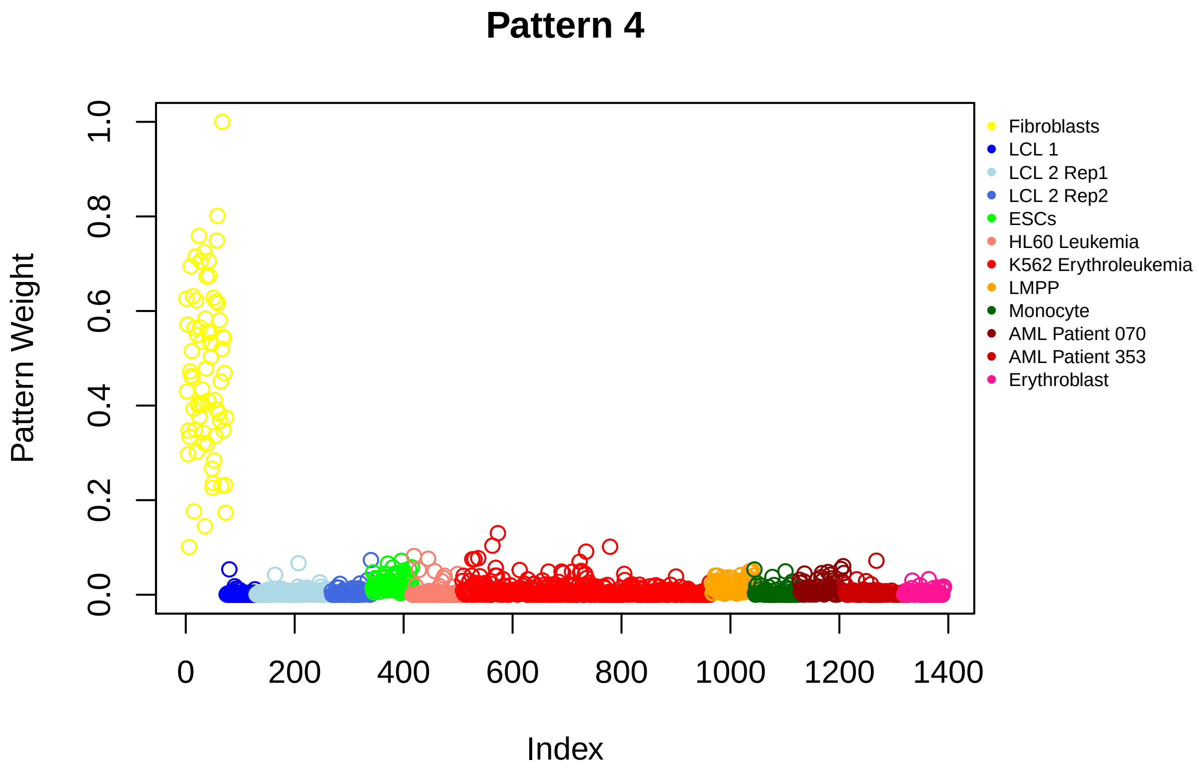
We can either plot each pattern indvidually

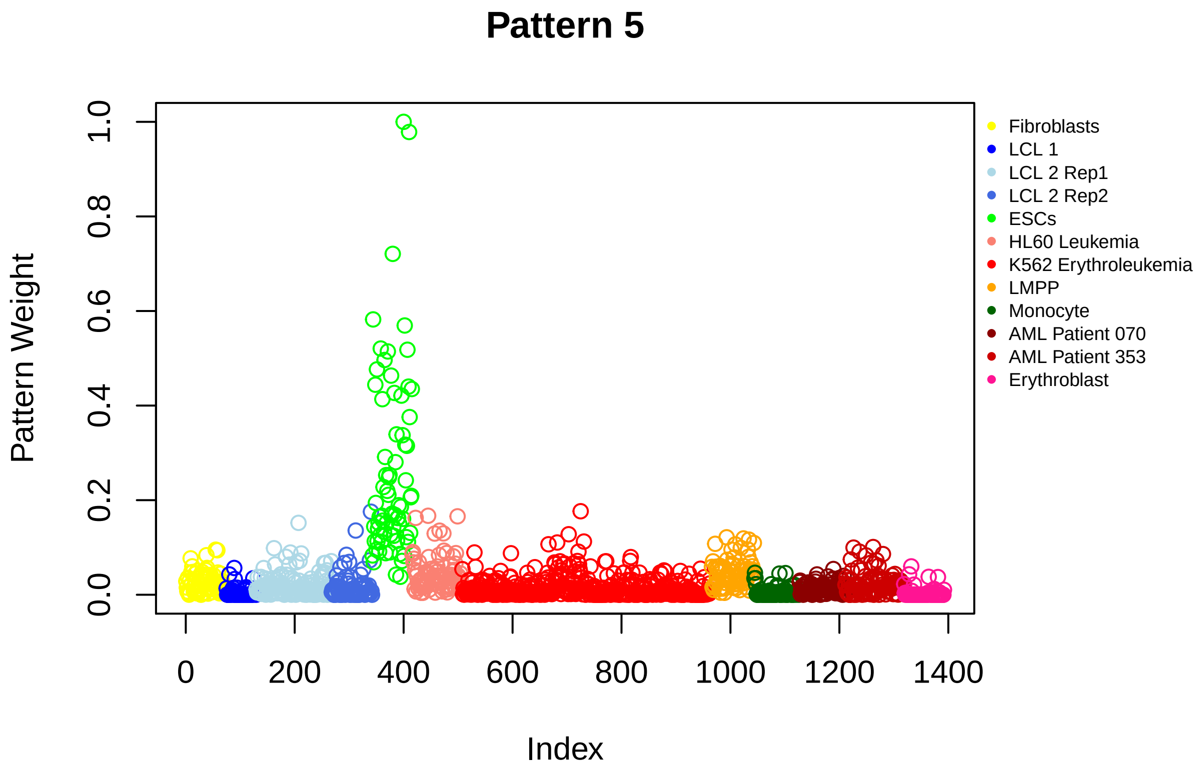
```r
#colors to plot by
col <- c('yellow', 'blue', 'lightblue', "royalblue", "green", "salmon", "red", "orange", "darkgreen", "


cgapsPlot(cgaps_result = schepCogapsResult, sample.classifier = schepCelltypes, cols = col, ylab = "Pat
```

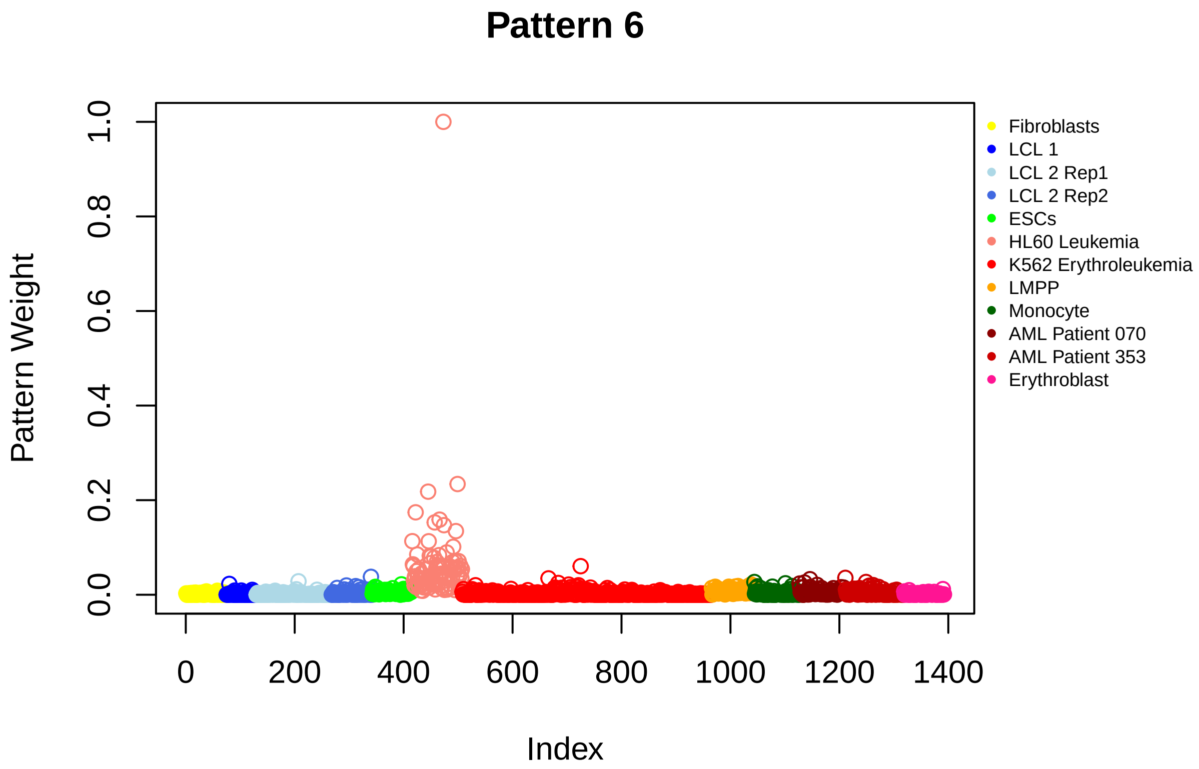

Pattern 1
Pattern Weight
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
1200
1400
Index
Fibroblasts
LCL 1
LCL 2 Rep1
LCL 2 Rep2
ESCs
HL60 Leukemia
K562 Erythroleukemia
LMPP
Monocyte
AML Patient 070
AML Patient 353
Erythroblast

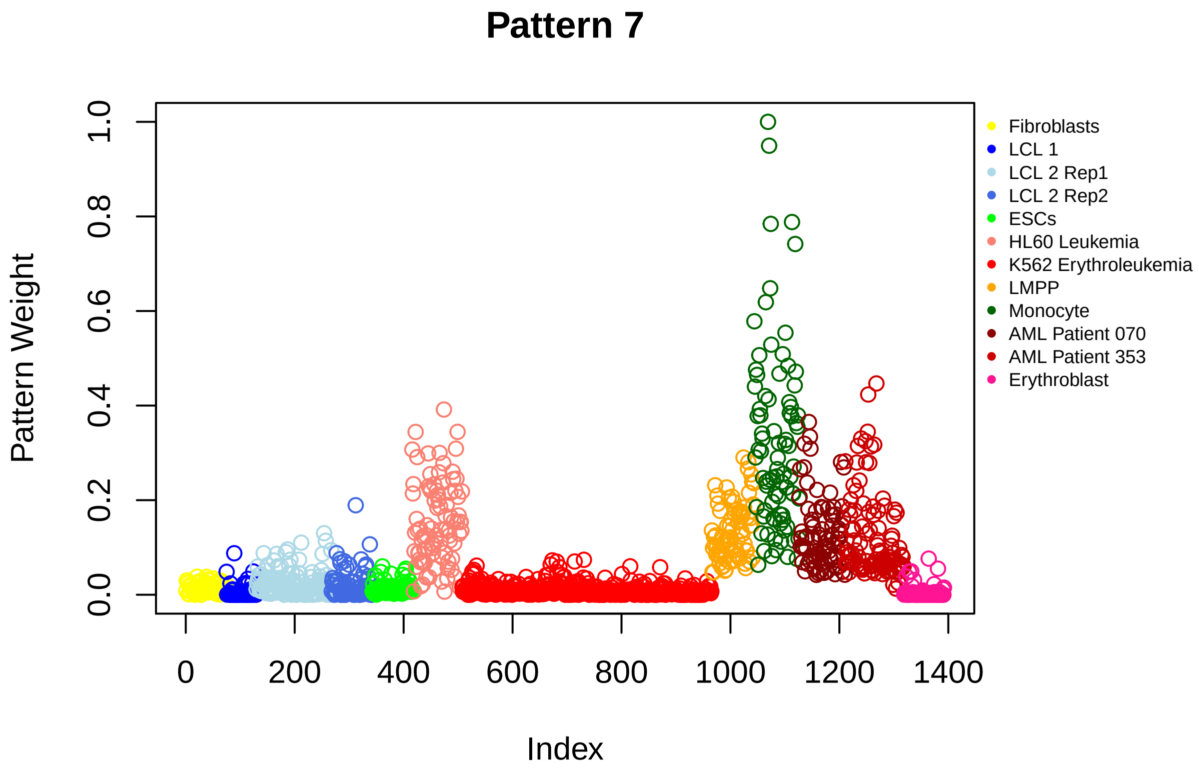

Pattern 2
Pattern Weight
0.0
0.2
0.4
0.6
0.8
1.0
Index
0
200
400
600
800
1000
1200
1400
Fibroblasts
LCL 1
LCL 2 Rep1
LCL 2 Rep2
ESCs
HL60 Leukemia
K562 Erythroleukemia
LMPP
Monocyte
AML Patient 070
AML Patient 353
Erythroblast


Pattern 3
Pattern Weight
0.0
0.2
0.4
0.6
0.8
1.0
Index
0
200
400
600
800
1000
1200
1400
Fibroblasts
LCL 1
LCL 2 Rep1
LCL 2 Rep2
ESCs
HL60 Leukemia
K562 Erythroleukemia
LMPP
Monocyte
AML Patient 070
AML Patient 353
Erythroblast


Pattern 4
Pattern Weight
0.0
0.2
0.4
0.6
0.8
1.0
Index
0
200
400
600
800
1000
1200
1400
Fibroblasts
LCL 1
LCL 2 Rep1
LCL 2 Rep2
ESCs
HL60 Leukemia
K562 Erythroleukemia
LMPP
Monocyte
AML Patient 070
AML Patient 353
Erythroblast

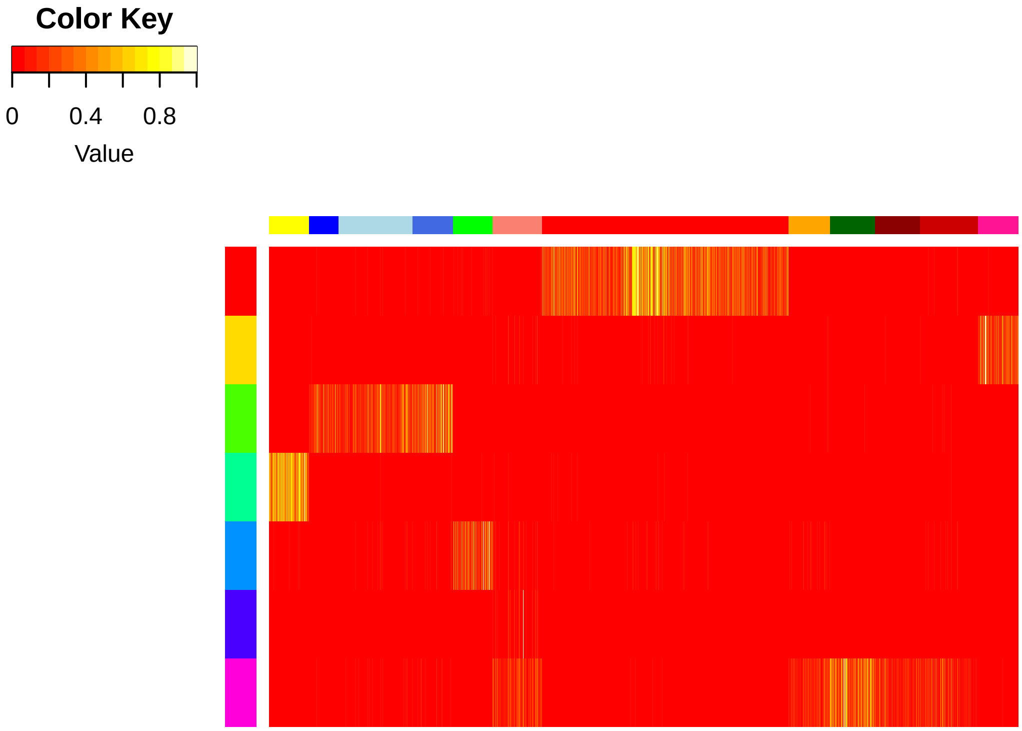

Pattern 5
Pattern Weight
0.0
0.2
0.4
0.6
0.8
1.0
Index
0
200
400
600
800
1000
1200
1400
Fibroblasts
LCL 1
LCL 2 Rep1
LCL 2 Rep2
ESCs
HL60 Leukemia
K562 Erythroleukemia
LMPP
Monocyte
AML Patient 070
AML Patient 353
Erythroblast


Pattern 6
Pattern Weight
0.0
0.2
0.4
0.6
0.8
1.0
Index
0
200
400
600
800
1000
1200
1400
Fibroblasts
LCL 1
LCL 2 Rep1
LCL 2 Rep2
ESCs
HL60 Leukemia
K562 Erythroleukemia
LMPP
Monocyte
AML Patient 070
AML Patient 353
Erythroblast



Or all together in a heatmap

```
heatmapPatternMatrix(cgaps_result = schepCogapsResult, sample.classifier = schepCelltypes, cellCols = c
```



We can note which patterns differentiate which cell types (for example that pattern 1 seems to be defining the K562 Erythroleukmia Cell Line). If any patterns are unclear, such as pattern 7, we can perform a Wilcoxon Rank Sum test to determine which cell types are most significantly associated with the pattern.

```
#get the pattern Matrix
patMatrix <- getSampleFactors(schepCogapsResult)
#perform a pairwise Wilcoxon test
pairwise.wilcox.test(patMatrix[,7], schepCelltypes, p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  patMatrix[, 7] and schepCelltypes
##
##                        Fibroblasts LCL 1   LCL 2 Rep1 LCL 2 Rep2 ESCs
## LCL 1                  1.6e-07     -       -          -          -
## LCL 2 Rep1             0.0254      1.5e-10 -          -          -
## LCL 2 Rep2             0.9310      1.9e-05 0.0677     -          -
## ESCs                   0.4005      6.8e-08 0.0088     0.9878     -
## HL60 Leukemia          < 2e-16     < 2e-16 < 2e-16    < 2e-16    < 2e-16
## K562 Erythroleukemia 2.0e-05     1.3e-05 4.4e-13    0.0108     0.0003
## LMPP                   < 2e-16     < 2e-16 < 2e-16    < 2e-16    < 2e-16
## Monocyte               < 2e-16     < 2e-16 < 2e-16    < 2e-16    < 2e-16
## AML Patient 070        < 2e-16     < 2e-16 < 2e-16    < 2e-16    < 2e-16
## AML Patient 353        < 2e-16     < 2e-16 < 2e-16    < 2e-16    < 2e-16
## Erythroblast           4.2e-12     0.5636  4.1e-16    1.3e-07    3.2e-13
##                      HL60 Leukemia K562 Erythroleukemia LMPP    Monocyte
## LCL 1                -             -                    -       -       
## LCL 2 Rep1           -             -                    -       -       
## LCL 2 Rep2           -             -                    -       -       
## ESCs                 -             -                    -       -       
## HL60 Leukemia        -             -                    -       -       
## K562 Erythroleukemia < 2e-16       -                    -       -       
## LMPP                 0.3480        < 2e-16              -       -       
## Monocyte             6.4e-12       < 2e-16              6.1e-14 -       
## AML Patient 070      0.1182        < 2e-16              0.3378  9.1e-15 
## AML Patient 353      0.0028        < 2e-16              0.0034  < 2e-16 
## Erythroblast         < 2e-16       1.6e-11              < 2e-16 < 2e-16 
##                      AML Patient 070 AML Patient 353
## LCL 1                -               -              
## LCL 2 Rep1           -               -              
## LCL 2 Rep2           -               -              
## ESCs                 -               -              
## HL60 Leukemia        -               -              
## K562 Erythroleukemia -               -              
## LMPP                 -               -              
## Monocyte             -               -              
## AML Patient 070      -               -              
## AML Patient 353      0.0729          -              
## Erythroblast         < 2e-16         < 2e-16        
## 
## P value adjustment method: BH
```

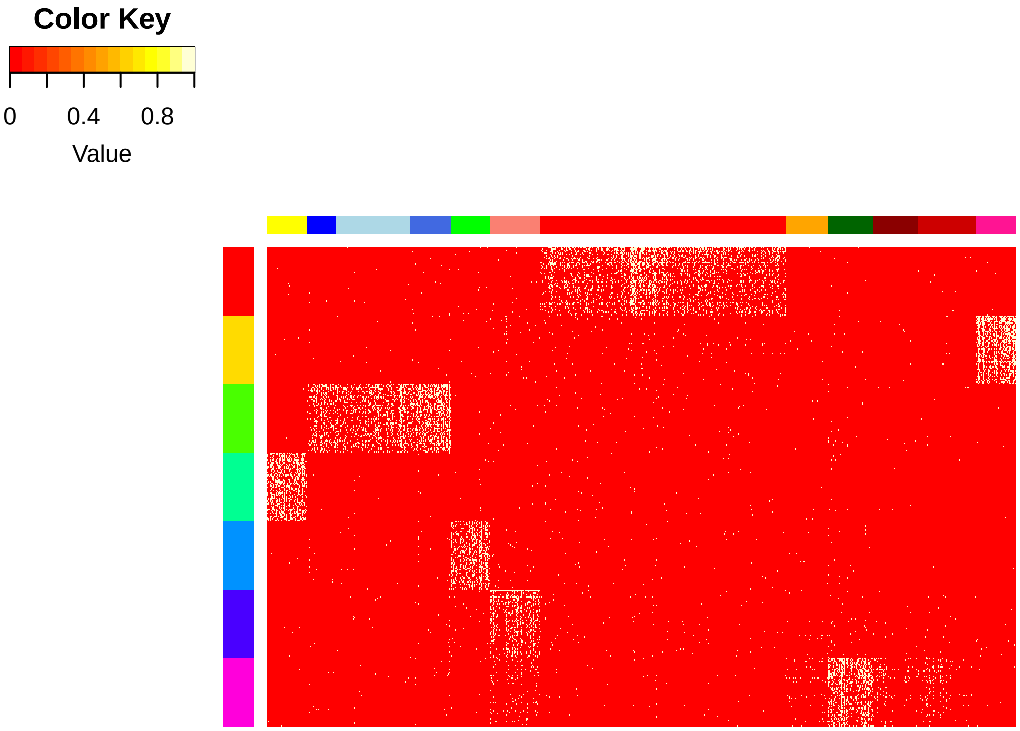
We see that pattern 7 is most strongly associated with the monocytes in the data.

Finding Regulatory Differences between Cell Types

Now that we know which patterns distinguish which cell types, we can look at those same patterns in the amplitude matrix (peaks by patterns) to determine which peaks are differentially accessible between the patterns and thus which peaks are differentially accessible between the cell types.

We can use the patternMarker Statistic (Stein-O'Brien et al, 2017) to find which peaks are most differentially accessible. To show the degree of differentiation, we can plot the 50 most pattern differentiating peaks for each pattern from the original data.

```
heatmapPatternMarkers(cgaps_result = schepCogapsResult, atac_data = schepFilteredData, celltypes = schep
```



The differentially accessible peaks we find distinguish the cell types we see in the pattern Matrix. In patterns 6 and 7 it seems to distinguish those cell types even better than the pattern Matrix does. This visualization allows us to see the biological differences between cell types CoGAPS is identifying.

Pathway Based Analysis

To make use of this differential accessibility data, one option is to try to find genes that fall within these peaks and determine whether the accessibility of certain groups of genes suggests differential pathway activation.

```
data("schepGranges")

#loading TxDb of human genes
library(Homo.sapiens)

#find genes known to fall witin the top 500 patternMarker peaks for each pattern
genes <- genePatternMatch(cogapsResult = schepCogapsResult, numregions = 500, generanges = schepGranges

#download hallmark pathways using msigdbr
library(dplyr)
```

```
pathways = msigdbr::msigdbr(species = "Homo sapiens", category =
                             "H") %>% dplyr::select(gs_name, gene_symbol) %>% as.data.frame()

#match these pattern Gene sets to hallmark pathways, using an adjusted p-value threshold of 0.001.
pathways <- pathwayMatch(gene_list = genes, pathways = pathways, p_threshold = 0.001)
```

```
pathways
```

```
## [[1]]
## [[1]]$gene_overlaps
## list()
##
## [[1]]$matched_pathways
## named list()
##
## [[1]]$pathway_names
## character(0)
##
##
## [[2]]
## [[2]]$gene_overlaps
## list()
##
## [[2]]$matched_pathways
## named list()
##
## [[2]]$pathway_names
## character(0)
##
##
## [[3]]
## [[3]]$gene_overlaps
## list()
##
## [[3]]$matched_pathways
## named list()
##
## [[3]]$pathway_names
## character(0)
##
##
## [[4]]
## [[4]]$gene_overlaps
## [[4]]$gene_overlaps[[1]]
## GeneOverlap object:
## listA size=748
## listB size=200
## Intersection size=35
## Overlapping p-value=3.4e-16
## Jaccard Index=0.0
##
## [[4]]$gene_overlaps[[2]]
## GeneOverlap object:
## listA size=748
## listB size=144
## Intersection size=16
## Overlapping p-value=1.9e-05
## Jaccard Index=0.0
##
```

```
##
## [[4]]$matched_pathways
## [[4]]$matched_pathways$HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION
##   [1] "ABI3BP"    "ACTA2"     "ADAM12"    "ANPEP"     "APLP1"
##   [6] "AREG"      "BASP1"     "BDNF"      "BGN"       "BMP1"
##  [11] "CADM1"     "CALD1"     "CALU"      "CAP2"      "CAPG"
##  [16] "CCN1"      "CCN2"      "CD44"      "CD59"      "CDH11"
##  [21] "CDH2"      "CDH6"      "COL11A1"   "COL12A1"   "COL16A1"
##  [26] "COL1A1"    "COL1A2"    "COL3A1"    "COL4A1"    "COL4A2"
##  [31] "COL5A1"    "COL5A2"    "COL5A3"    "COL6A2"    "COL6A3"
##  [36] "COL7A1"    "COL8A2"    "COLGALT1"  "COMP"      "COPA"
##  [41] "CRLF1"     "CTHRC1"    "CXCL1"     "CXCL12"    "CXCL6"
##  [46] "CXCL8"     "DAB2"      "DCN"       "DKK1"      "DPYSL3"
##  [51] "DST"       "ECM1"      "ECM2"      "EDIL3"     "EFEMP2"
##  [56] "ELN"       "EMP3"      "ENO2"      "FAP"       "FAS"
##  [61] "FBLN1"     "FBLN2"     "FBLN5"     "FBN1"      "FBN2"
##  [66] "FERMT2"    "FGF2"      "FLNA"      "FMOD"      "FN1"
##  [71] "FOXC2"     "FSTL1"     "FSTL3"     "FUCA1"     "FZD8"
##  [76] "GADD45A"   "GADD45B"   "GAS1"      "GEM"       "GJA1"
##  [81] "GLIPR1"    "GPC1"      "GPX7"      "GREM1"     "HTRA1"
##  [86] "ID2"       "IGFBP2"    "IGFBP3"    "IGFBP4"    "IL15"
##  [91] "IL32"      "IL6"       "INHBA"     "ITGA2"     "ITGA5"
##  [96] "ITGAV"     "ITGB1"     "ITGB3"     "ITGB5"     "JUN"
## [101] "LAMA1"     "LAMA2"     "LAMA3"     "LAMC1"     "LAMC2"
## [106] "LGALS1"    "LOX"       "LOXL1"     "LOXL2"     "LRP1"
## [111] "LRRC15"    "LUM"       "MAGEE1"    "MATN2"     "MATN3"
## [116] "MCM7"      "MEST"      "MFAP5"     "MGP"       "MMP1"
## [121] "MMP14"     "MMP2"      "MMP3"      "MSX1"      "MXRA5"
## [126] "MYL9"      "MYLK"      "NID2"      "NNMT"      "NOTCH2"
## [131] "NT5E"      "NTM"       "OXTR"      "P3H1"      "PCOLCE"
## [136] "PCOLCE2"   "PDGFRB"    "PDLIM4"    "PFN2"      "PLAUR"
## [141] "PLOD1"     "PLOD2"     "PLOD3"     "PMEPA1"    "PMP22"
## [146] "POSTN"     "PPIB"      "PRRX1"     "PRSS2"     "PTHLH"
## [151] "PTX3"      "PVR"       "QSOX1"     "RGS4"      "RHOB"
## [156] "SAT1"      "SCG2"      "SDC1"      "SDC4"      "SERPINE1"
## [161] "SERPINE2"  "SERPINH1"  "SFRP1"     "SFRP4"     "SGCB"
## [166] "SGCD"      "SGCG"      "SLC6A8"    "SLIT2"     "SLIT3"
## [171] "SNAI2"     "SNTB1"     "SPARC"     "SPOCK1"    "SPP1"
## [176] "TAGLN"     "TFPI2"     "TGFB1"     "TGFBI"     "TGFBR3"
## [181] "TGM2"      "THBS1"     "THBS2"     "THY1"      "TIMP1"
## [186] "TIMP3"     "TNC"       "TNFAIP3"   "TNFRSF11B" "TNFRSF12A"
## [191] "TPM1"      "TPM2"      "TPM4"      "VCAM1"     "VCAN"
## [196] "VEGFA"     "VEGFC"     "VIM"       "WIPF1"     "WNT5A"
##
## [[4]]$matched_pathways$HALLMARK_UV_RESPONSE_DN
##   [1] "ABCC1"      "ACVR2A"     "ADD3"       "ADGRL2"     "ADORA2B"
##   [6] "AGGF1"      "AKT3"       "AL162171.1" "AMPH"       "ANXA2"
##  [11] "ANXA4"      "APBB2"      "ARHGEF9"    "ATP2B1"     "ATP2B4"
##  [16] "ATP2C1"     "ATRN"       "ATRX"       "ATXN1"      "BCKDHB"
##  [21] "BDNF"       "BHLHE40"    "BMPR1A"     "CACNA1A"    "CAP2"
##  [26] "CAV1"       "CCN1"       "CDC42BPA"   "CDK13"      "CDKN1B"
##  [31] "CDON"       "CELF2"      "CITED2"     "COL11A1"    "COL1A1"
##  [36] "COL1A2"     "COL3A1"     "COL5A2"     "DAB2"       "DBP"
##  [41] "DDAH1"      "DLC1"       "DLG1"       "DMAC2L"     "DUSP1"
##  [46] "DYRK1A"     "EFEMP1"     "ERBB2"      "F3"         "FBLN5"
##  [51] "FHL2"       "FYN"        "FZD2"       "GCNT1"      "GJA1"
##  [56] "GRK5"       "HAS2"       "ICA1"       "ID1"        "IGF1R"
##  [61] "IGFBP5"     "INPP4B"     "INSIG1"     "IRS1"       "ITGB3"
##  [66] "KALRN"      "KCNMA1"     "KIT"        "LAMC1"      "LDLR"
##  [71] "LPAR1"      "LTBP1"      "MAGI2"      "MAP1B"      "MAP2K5"
##  [76] "MAPK14"     "MET"        "MGLL"       "MGMT"       "MIOS"
##  [81] "MMP16"      "MRPS31"     "MT1E"       "MTA1"       "MYC"
##  [86] "NEK7"       "NFIB"       "NFKB1"      "NIPBL"      "NOTCH2"
##  [91] "NR1D2"      "NR3C1"      "NRP1"       "PDGFRB"     "PDLIM5"
##  [96] "PEX14"      "PHF3"       "PIAS3"      "PIK3CD"     "PIK3R3"
## [101] "PLCB4"      "PLPP3"      "PMP22"      "PPARG"      "PRDM2"
## [106] "PRKAR2B"    "PRKCA"      "PRKCE"      "PTEN"       "PTGFR"
## [111] "PTPRM"      "RASA2"      "RBPMS"      "RGS4"       "RND3"
## [116] "RUNX1"      "RXRA"       "SCAF8"      "SCHIP1"     "SCN8A"
## [121] "SDC2"       "SERPINE1"   "SFMBT1"     "SIPA1L1"    "SLC22A18"
## [126] "SLC7A1"     "SMAD3"      "SMAD7"      "SNAI2"      "SPOP"
## [131] "SRI"        "SYNE1"      "SYNJ2"      "TENT4A"     "TFPI"
## [136] "TGFBR2"     "TGFBR3"     "TJP1"       "TOGARAM1"   "VAV2"
## [141] "VLDLR"      "WDR37"      "YTHDC1"     "ZMIZ1"
##
##
## [[4]]$pathway_names
## [1] "HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION"
## [2] "HALLMARK_UV_RESPONSE_DN"
##
##
## [[5]]
## [[5]]$gene_overlaps
## [[5]]$gene_overlaps[[1]]
## GeneOverlap object:
## listA size=828
## listB size=200
## Intersection size=21
## Overlapping p-value=1.2e-05
## Jaccard Index=0.0
##
##
## [[5]]$matched_pathways
## [[5]]$matched_pathways$HALLMARK_ESTROGEN_RESPONSE_EARLY
##   [1] "ABAT"     "ABCA3"    "ABHD2"    "ABLIM1"   "ADCY1"    "ADCY9"
##   [7] "ADD3"     "AFF1"     "AKAP1"    "ALDH3B1"  "AMFR"     "ANXA9"
##  [13] "AQP3"     "AR"       "AREG"     "ARL3"     "ASB13"    "B4GALT1"
##  [19] "BAG1"     "BCL11B"   "BCL2"     "BHLHE40"  "BLVRB"    "CA12"
##  [25] "CALB2"    "CALCR"    "CANT1"    "CBFA2T3"  "CCN5"     "CCND1"
##  [31] "CD44"     "CELSR1"   "CELSR2"   "CHPT1"    "CISH"     "CLDN7"
##  [37] "CLIC3"    "CXCL12"   "CYP26B1"  "DEPTOR"   "DHCR7"    "DHRS2"
##  [43] "DHRS3"    "DLC1"     "DYNLT3"   "EGR3"     "ELF1"     "ELF3"
##  [49] "ELOVL2"   "ELOVL5"   "ENDOD1"   "ESRP2"    "FAM102A"  "FARP1"
##  [55] "FASN"     "FCMR"     "FDFT1"    "FHL2"     "FKBP4"    "FKBP5"
##  [61] "FLNB"     "FOS"      "FOXC1"    "FRK"      "GAB2"     "GFRA1"
##  [67] "GJA1"     "GLA"      "GREB1"    "HES1"     "HR"       "HSPB8"
##  [73] "IGF1R"    "IGFBP4"   "IL17RB"   "IL6ST"    "INHBB"    "INPP5F"
##  [79] "ISG20L2"  "ITPK1"    "JAK2"     "KAZN"     "KCNK15"   "KCNK5"
##  [85] "KDM4B"    "KLF10"    "KLF4"     "KLK10"    "KRT13"    "KRT15"
##  [91] "KRT18"    "KRT19"    "KRT8"     "LAD1"     "LRIG1"    "MAPT"
##  [97] "MAST4"    "MED13L"   "MED24"    "MICB"     "MINDY1"   "MLPH"
## [103] "MPPED2"   "MREG"     "MSMB"     "MUC1"     "MYB"      "MYBBP1A"
## [109] "MYBL1"    "MYC"      "MYOF"     "NADSYN1"  "NAV2"     "NBL1"
## [115] "NCOR2"    "NPY1R"    "NRIP1"    "NXT1"     "OLFM1"    "OLFML3"
## [121] "OPN3"     "OVOL2"    "P2RY2"    "PAPSS2"   "PDLIM3"   "PDZK1"
## [127] "PEX11A"   "PGR"      "PLAAT3"   "PMAIP1"   "PODXL"    "PPIF"
## [133] "PRSS23"   "PTGES"    "RAB17"    "RAB31"    "RAPGEFL1" "RARA"
## [139] "RASGRP1"  "RBBP8"    "REEP1"    "RET"      "RETREG1"  "RHOBTB3"
## [145] "RHOD"     "RPS6KA2"  "RRP12"    "SCARB1"   "SCNN1A"   "SEC14L2"
## [151] "SEMA3B"   "SFN"      "SH3BP5"   "SIAH2"    "SLC16A1"  "SLC19A2"
## [157] "SLC1A1"   "SLC1A4"   "SLC22A5"  "SLC24A3"  "SLC26A2"  "SLC27A2"
## [163] "SLC2A1"   "SLC37A1"  "SLC39A6"  "SLC7A2"   "SLC7A5"   "SLC9A3R1"
## [169] "SNX24"    "SOX3"     "STC2"     "SULT2B1"  "SVIL"     "SYBU"
## [175] "SYNGR1"   "SYT12"    "TBC1D30"  "TFAP2C"   "TFF1"     "TFF3"
## [181] "TGIF2"    "TGM2"     "THSD4"    "TIAM1"    "TIPARP"   "TJP3"
## [187] "TMEM164"  "TMPRSS3"  "TOB1"     "TPBG"     "TPD52L1"  "TSKU"
## [193] "TTC39A"   "TUBB2B"   "UGCG"     "UNC119"   "WFS1"     "WWC1"
## [199] "XBP1"     "ZNF185"
##
##
## [[5]]$pathway_names
## [1] "HALLMARK_ESTROGEN_RESPONSE_EARLY"
##
##
## [[6]]
## [[6]]$gene_overlaps
## list()
##
## [[6]]$matched_pathways
## named list()
##
## [[6]]$pathway_names
## character(0)
##
##
## [[7]]
## [[7]]$gene_overlaps
## [[7]]$gene_overlaps[[1]]
## GeneOverlap object:
## listA size=760
## listB size=200
## Intersection size=24
## Overlapping p-value=5.3e-08
## Jaccard Index=0.0
##
## [[7]]$gene_overlaps[[2]]
## GeneOverlap object:
## listA size=760
## listB size=200
## Intersection size=21
## Overlapping p-value=3.2e-06
## Jaccard Index=0.0
```

```
##
##
## [[7]]$matched_pathways
## [[7]]$matched_pathways$HALLMARK_INFLAMMATORY_RESPONSE
##   [1] "ABCA1"    "ABI1"     "ACVR1B"   "ACVR2A"   "ADGRE1"   "ADM"
##   [7] "ADORA2B"  "ADRM1"    "AHR"      "APLNR"    "AQP9"     "ATP2A2"
##  [13] "ATP2B1"   "ATP2C1"   "AXL"      "BDKRB1"   "BEST1"    "BST2"
##  [19] "BTG2"     "C3AR1"    "C5AR1"    "CALCRL"   "CCL17"    "CCL2"
##  [25] "CCL20"    "CCL22"    "CCL24"    "CCL5"     "CCL7"     "CCR7"
##  [31] "CCRL2"    "CD14"     "CD40"     "CD48"     "CD55"     "CD69"
##  [37] "CD70"     "CD82"     "CDKN1A"   "CHST2"    "CLEC5A"   "CMKLR1"
##  [43] "CSF1"     "CSF3"     "CSF3R"    "CX3CL1"   "CXCL10"   "CXCL11"
##  [49] "CXCL6"    "CXCL8"    "CXCL9"    "CXCR6"    "CYBB"     "DCBLD2"
##  [55] "EBI3"     "EDN1"     "EIF2AK2"  "EMP3"     "EREG"     "F3"
##  [61] "FFAR2"    "FPR1"     "FZD5"     "GABBR1"   "GCH1"     "GNA15"
##  [67] "GNAI3"    "GP1BA"    "GPC3"     "GPR132"   "GPR183"   "HAS2"
##  [73] "HBEGF"    "HIF1A"    "HPN"      "HRH1"     "ICAM1"    "ICAM4"
##  [79] "ICOSLG"   "IFITM1"   "IFNAR1"   "IFNGR2"   "IL10"     "IL10RA"
##  [85] "IL12B"    "IL15"     "IL15RA"   "IL18"     "IL18R1"   "IL18RAP"
##  [91] "IL1A"     "IL1B"     "IL1R1"    "IL2RB"    "IL4R"     "IL6"
##  [97] "IL7R"     "INHBA"    "IRAK2"    "IRF1"     "IRF7"     "ITGA5"
## [103] "ITGB3"    "ITGB8"    "KCNA3"    "KCNJ2"    "KCNMB2"   "KIF1B"
## [109] "KLF6"     "LAMP3"    "LCK"      "LCP2"     "LDLR"     "LIF"
## [115] "LPAR1"    "LTA"      "LY6E"     "LYN"      "MARCO"    "MEFV"
## [121] "MEP1A"    "MET"      "MMP14"    "MSR1"     "MXD1"     "MYC"
## [127] "NAMPT"    "NDP"      "NFKB1"    "NFKBIA"   "NLRP3"    "NMI"
## [133] "NMUR1"    "NOD2"     "NPFFR2"   "OLR1"     "OPRK1"    "OSM"
## [139] "OSMR"     "P2RX4"    "P2RX7"    "P2RY2"    "PCDH7"    "PDE4B"
## [145] "PDPN"     "PIK3R5"   "PLAUR"    "PROK2"    "PSEN1"    "PTAFR"
## [151] "PTGER2"   "PTGER4"   "PTGIR"    "PTPRE"    "PVR"      "RAF1"
## [157] "RASGRP1"  "RELA"     "RGS1"     "RGS16"    "RHOG"     "RIPK2"
## [163] "RNF144B"  "ROS1"     "RTP4"     "SCARF1"   "SCN1B"    "SELE"
## [169] "SELENOS"  "SELL"     "SEMA4D"   "SERPINE1" "SGMS2"    "SLAMF1"
## [175] "SLC11A2"  "SLC1A2"   "SLC28A2"  "SLC31A1"  "SLC31A2"  "SLC4A4"
## [181] "SLC7A1"   "SLC7A2"   "SPHK1"    "SRI"      "STAB1"    "TACR1"
## [187] "TACR3"    "TAPBP"    "TIMP1"    "TLR1"     "TLR2"     "TLR3"
## [193] "TNFAIP6"  "TNFRSF1B" "TNFRSF9"  "TNFSF10"  "TNFSF15"  "TNFSF9"
## [199] "TPBG"     "VIP"
##
## [[7]]$matched_pathways$HALLMARK_TNFA_SIGNALING_VIA_NFKB
##   [1] "ABCA1"      "AC129492.1" "ACKR3"      "AREG"       "ATF3"
##   [6] "ATP2B1"     "B4GALT1"    "B4GALT5"    "BCL2A1"     "BCL3"
##  [11] "BCL6"       "BHLHE40"    "BIRC2"      "BIRC3"      "BMP2"
##  [16] "BTG1"       "BTG2"       "BTG3"       "CCL2"       "CCL20"
##  [21] "CCL4"       "CCL5"       "CCN1"       "CCND1"      "CCNL1"
##  [26] "CCRL2"      "CD44"       "CD69"       "CD80"       "CD83"
##  [31] "CDKN1A"     "CEBPB"      "CEBPD"      "CFLAR"      "CLCF1"
##  [36] "CSF1"       "CSF2"       "CXCL1"      "CXCL10"     "CXCL11"
##  [41] "CXCL2"      "CXCL3"      "CXCL6"      "DDX58"      "DENND5A"
##  [46] "DNAJB4"     "DRAM1"      "DUSP1"      "DUSP2"      "DUSP4"
##  [51] "DUSP5"      "EDN1"       "EFNA1"      "EGR1"       "EGR2"
##  [56] "EGR3"       "EHD1"       "EIF1"       "ETS2"       "F2RL1"
##  [61] "F3"         "FJX1"       "FOS"        "FOSB"       "FOSL1"
##  [66] "FOSL2"      "FUT4"       "G0S2"       "GADD45A"    "GADD45B"
##  [71] "GCH1"       "GEM"        "GFPT2"      "GPR183"     "HBEGF"
##  [76] "HES1"       "ICAM1"      "ICOSLG"     "ID2"        "IER2"
##  [81] "IER3"       "IER5"       "IFIH1"      "IFIT2"      "IFNGR2"
##  [86] "IL12B"      "IL15RA"     "IL18"       "IL1A"       "IL1B"
##  [91] "IL23A"      "IL6"        "IL6ST"      "IL7R"       "INHBA"
##  [96] "IRF1"       "IRS2"       "JAG1"       "JUN"        "JUNB"
## [101] "KDM6B"      "KLF10"      "KLF2"       "KLF4"       "KLF6"
## [106] "KLF9"       "KYNU"       "LAMB3"      "LDLR"       "LIF"
## [111] "LITAF"      "MAFF"       "MAP2K3"     "MAP3K8"     "MARCKS"
## [116] "MCL1"       "MSC"        "MXD1"       "MYC"        "NAMPT"
## [121] "NFAT5"      "NFE2L2"     "NFIL3"      "NFKB1"      "NFKB2"
## [126] "NFKBIA"     "NFKBIE"     "NINJ1"      "NR4A1"      "NR4A2"
## [131] "NR4A3"      "OLR1"       "PANX1"      "PDE4B"      "PDLIM5"
## [136] "PFKFB3"     "PHLDA1"     "PHLDA2"     "PLAU"       "PLAUR"
## [141] "PLEK"       "PLK2"       "PLPP3"      "PMEPA1"     "PNRC1"
## [146] "PPP1R15A"   "PTGER4"     "PTGS2"      "PTPRE"      "PTX3"
## [151] "RCAN1"      "REL"        "RELA"       "RELB"       "RHOB"
## [156] "RIPK2"      "RNF19B"     "SAT1"       "SDC4"       "SERPINB2"
## [161] "SERPINB8"   "SERPINE1"   "SGK1"       "SIK1"       "SLC16A6"
## [166] "SLC2A3"     "SLC2A6"     "SMAD3"      "SNN"        "SOCS3"
## [171] "SOD2"       "SPHK1"      "SPSB1"      "SQSTM1"     "STAT5A"
## [176] "TANK"       "TAP1"       "TGIF1"      "TIPARP"     "TLR2"
## [181] "TNC"        "TNF"        "TNFAIP2"    "TNFAIP3"    "TNFAIP6"
## [186] "TNFAIP8"    "TNFRSF9"    "TNFSF9"     "TNIP1"      "TNIP2"
## [191] "TRAF1"      "TRIB1"      "TRIP10"     "TSC22D1"    "TUBB2A"
## [196] "VEGFA"      "YRDC"       "ZBTB10"     "ZC3H12A"    "ZFP36"
##
##
## [[7]]$pathway_names
## [1] "HALLMARK_INFLAMMATORY_RESPONSE"   "HALLMARK_TNFA_SIGNALING_VIA_NFKB"
```

Several patterns do not return Hallmark pathways at this level of significance, but those that do seem logical in the cell types those patterns differentiate.

Of particular note, we find the Epithelial Mesenchymal Transition pathway to be strongly associated with Fibroblasts, which is known to be the classical wound healing pathway in Fibroblasts. Additionally, monocytes are most strongly associated with the Hallmark Inflammatory Response, as we would expect for inflammatory cells.

Motif/Transcription Factor Based Analysis

The other way we can use differential peak information is to match to DNA motifs and known Transcription Factor binding at those motifs.

```
motifResults = simpleMotifTFMatch(cogapsResult = schepCogapsResult, numregions = 50, generanges = schepC
```

```
## Registered S3 method overwritten by 'R.oo':
##   method        from
##   throw.default R.methodsS3
```

We can get a summary of TF binding, generally having more confidence in those that have multiple motifs at which the same TF could bind.

```
motifResults$tfMatchSummary
```

```
## [[1]]
##        EGR1 GATA1::TAL1         SP2         SP1        ELF4         FOS
##           4           3           3           2           1           1
##       FOXB1       GATA2        IRF1        KLF5       MEF2D        NFE2
##           1           1           1           1           1           1
##       NFKB2        NFYA      POU3F4       RREB1      ZNF263
##           1           1           1           1           1
##
## [[2]]
## GATA1::TAL1       GATA2        IRF1      POU3F3       PROP1        DUX4
##           3           3           2           2           2           1
##       FOXB1       GATA3        HSF2        JDP2        JUND        MAFK
##           1           1           1           1           1           1
##        NFE2       NR2F1        RFX3       STAT1  TAL1::TCF3      ZNF740
##           1           1           1           1           1           1
##
## [[3]]
##       ESRRB       HNF1A       RREB1      ZNF263   BATF::JUN       GRHL1
##           2           2           2           2           1           1
##        HSF2        IRF1        IRF7         JUN  JUN(var.2)       MEF2C
##           1           1           1           1           1           1
##        NFYA      POU3F4      POU4F3       PROP1        RELA RORA(var.2)
##           1           1           1           1           1           1
##       SMAD3        SPI1         TEF
##           1           1           1
##
## [[4]]
##        FOSL2    BATF::JUN        CEBPA         EGR1          FOS
##            2            1            1            1            1
##        FOSL1        FOXA1        FOXF2        FOXP1       HOXC12
##            1            1            1            1            1
##         JUND         LEF1         NFE2  NR1H2::RXRA       POU2F1
##            1            1            1            1            1
##       POU6F2        PPARG        SMAD3 STAT1::STAT2       TCF7L2
##            1            1            1            1            1
##         TP53         ZEB1         ZIC4       ZNF263
##            1            1            1            1
##
## [[5]]
##    POU3F4    ZNF263      RFX2 BATF::JUN      E2F6      ESR2     HNF1A
##         3         3         2         1         1         1         1
##     HNF4G    HOXC13      HSF1       MSC     NR2C2      PAX5    POU1F1
##         1         1         1         1         1         1         1
##    POU2F2    POU3F2    POU3F3     RREB1       SP2     TBX20
##         1         1         1         1         1         1
##
## [[6]]
##       KLF5      RREB1       CDX2      CEBPA        DBP       ESR2
##          2          2          1          1          1          1
##      ESRRB      FOSL2      GRHL1     HOXD13        ID4       IRF1
##          1          1          1          1          1          1
##        MSC NFIC::TLX1       NFYA      NR2F1       RELA      RUNX3
##          1          1          1          1          1          1
##        SP2        SP8     SREBF2      STAT1     ZNF263
##          1          1          1          1          1
##
## [[7]]
##          IRF1        ZNF263         BATF3          CDX2          EGR1
##             3             2             1             1             1
##         FOXH1    GATA1::TAL1           JUN    JUN(var.2)         MEF2D
##             1             1             1             1             1
##          NRF1        POU3F2        POU3F4     RARA::RXRA          REST
##             1             1             1             1             1
##         RREB1           SP2          SPI1   STAT1::STAT2         TBX15
##             1             1             1             1             1
## TFAP2B(var.2)        ZBTB18
##             1             1
```

The entrez gene summary is returned for all TFs found in matching, allowing us to easily check whether a TF seems like a plausible regulatory factor in a given cell type. For example, if we want to take a look at the function of EGR1 given the prevalence of potential binding sites for it found in pattern1:

```
motifResults$tfDescriptions[[1]][which(motifResults$tfDescriptions[[1]][,2]=="EGR1"), 1]
```

```
##
## 1: The protein encoded by this gene belongs to the EGR family of C2H2-type zinc-finger proteins. It i
```

This description gives us a sense that this TF may play some role in the oncogenesis of this cancer cell line.

Transfer Learning with ProjectR

To determine if the patterns we have identified with CoGAPS appear in other data sets we can apply transfer learning between ATAC datasets using projectR (Stein-O'Brein and Sharma, 2019). projectR allows us to project patterns learned on one data set into another.

This can be useful for validating the generality and biological relevance of patterns, determining if learned signatures appear in other datasets without needing to run CoGAPS again, or simply to learn more regulatory information by combining patterns learned on different data sets.

To demonstrate we will use a set of scATAC data published by Buenrostro et al, 2018 containing a number of hematopoietic lineage cells.

```
#getting count matrix - peaks x cells
repmis::source_data("https://github.com/FertigLab/ATACCoGAPS/blob/master/BuenrostroFinalSubsetData.Rdata
```

```
## [1] "BuenrostroFinalSubsetData"
```

```
#getting GRanges for peaks
repmis::source_data("https://github.com/FertigLab/ATACCoGAPS/blob/master/BuenrostroGRanges.Rdata?raw=tru
```

```
## [1] "BuenrostroGRanges"
```

```
#getting celltypes
repmis::source_data("https://github.com/FertigLab/ATACCoGAPS/blob/master/BuenrostroCellTypes.Rdata?raw=t
```

```
## [1] "BuenrostroCellTypes"
```

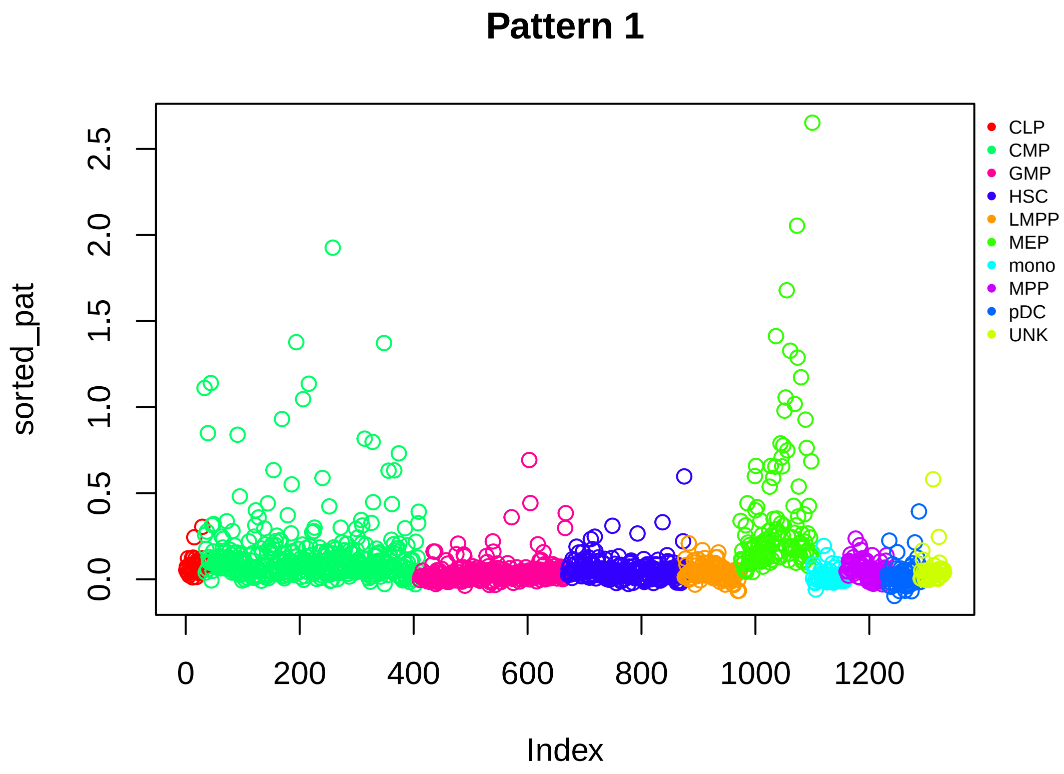
To transfer patterns between the two data sets, we have to find which peaks overlap between data sets because we can only project onto that overlapping subset. To do this we employ a wrapper function around projectR which automatically maps overlapping peaks together.

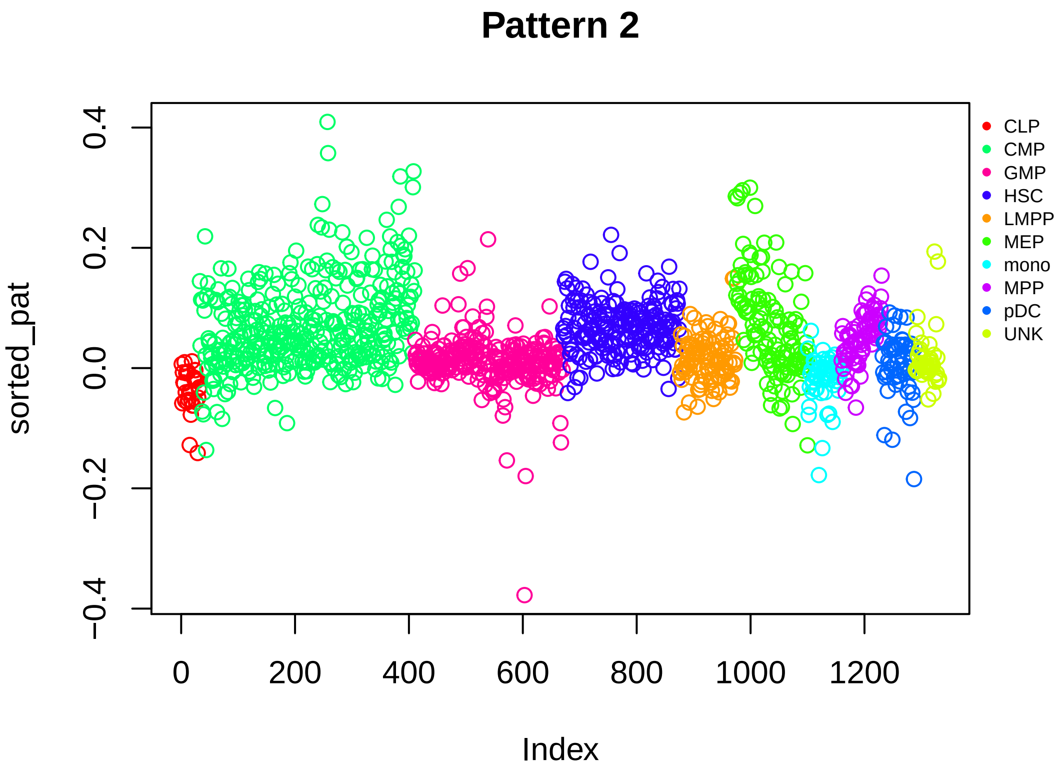
```
projectRResults <- ATACTransferLearning(newData = BuenrostroFinalSubsetData, CoGAPSResult = schepCogapsR
```

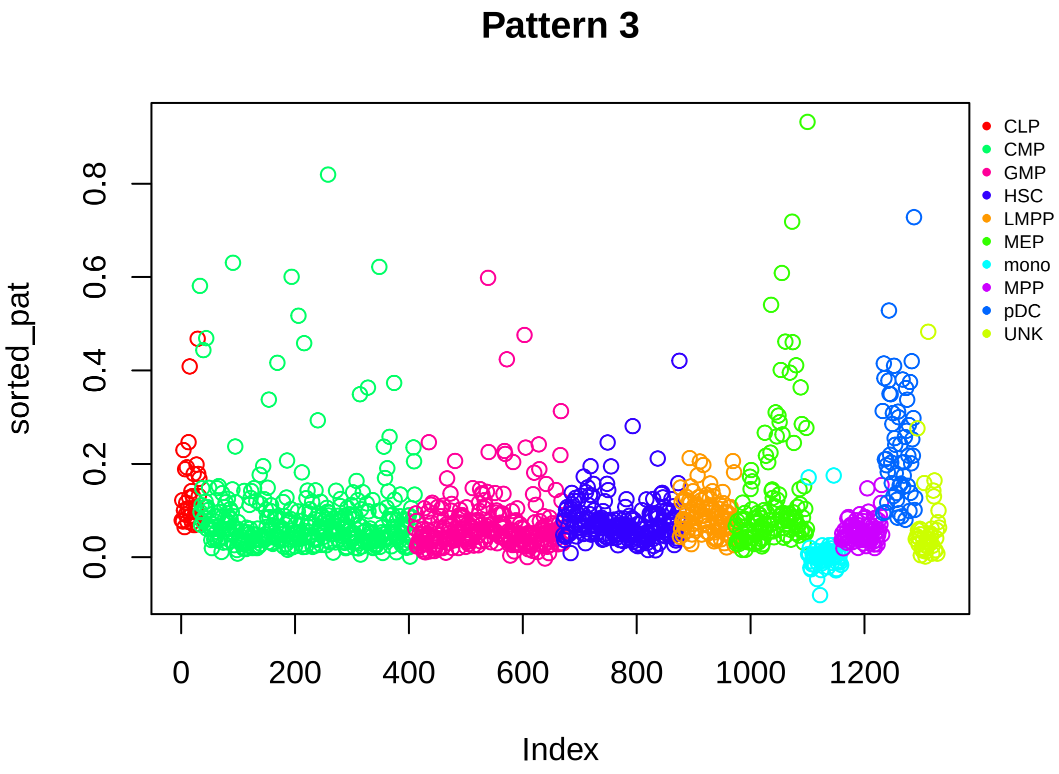
```
## [1] "62387 row names matched between data and loadings"
## [1] "Updated dimension of data: 62387 1331"
```

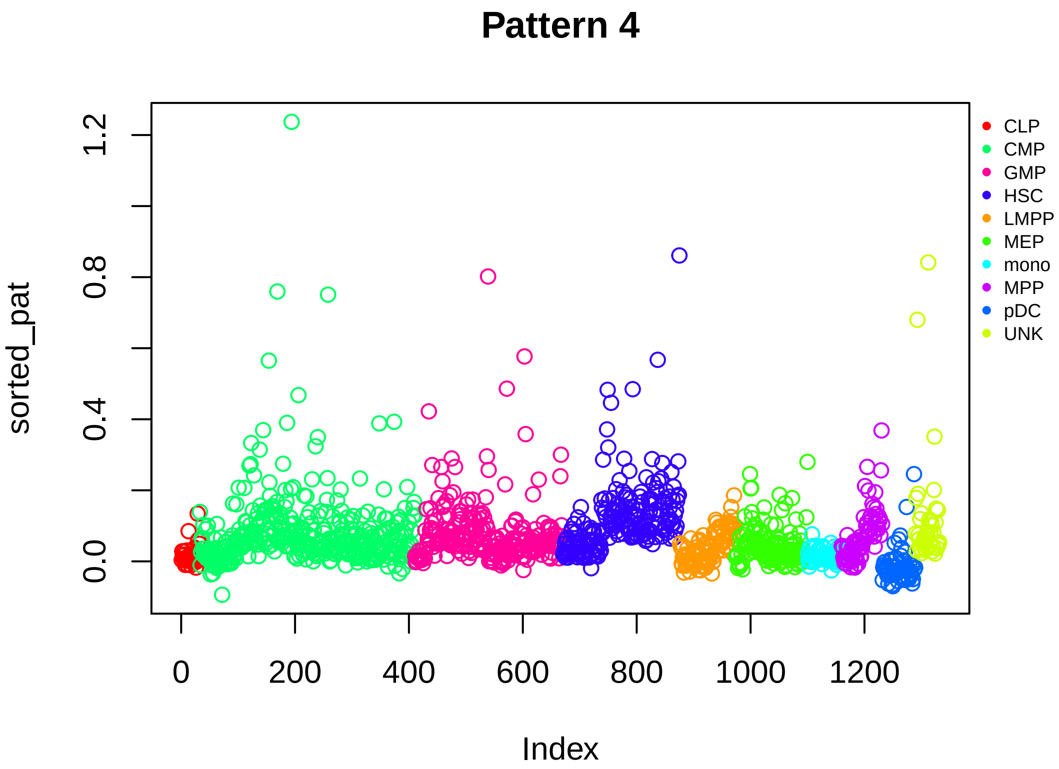
We can then plot the output patterns to see how well they transfer into the target data

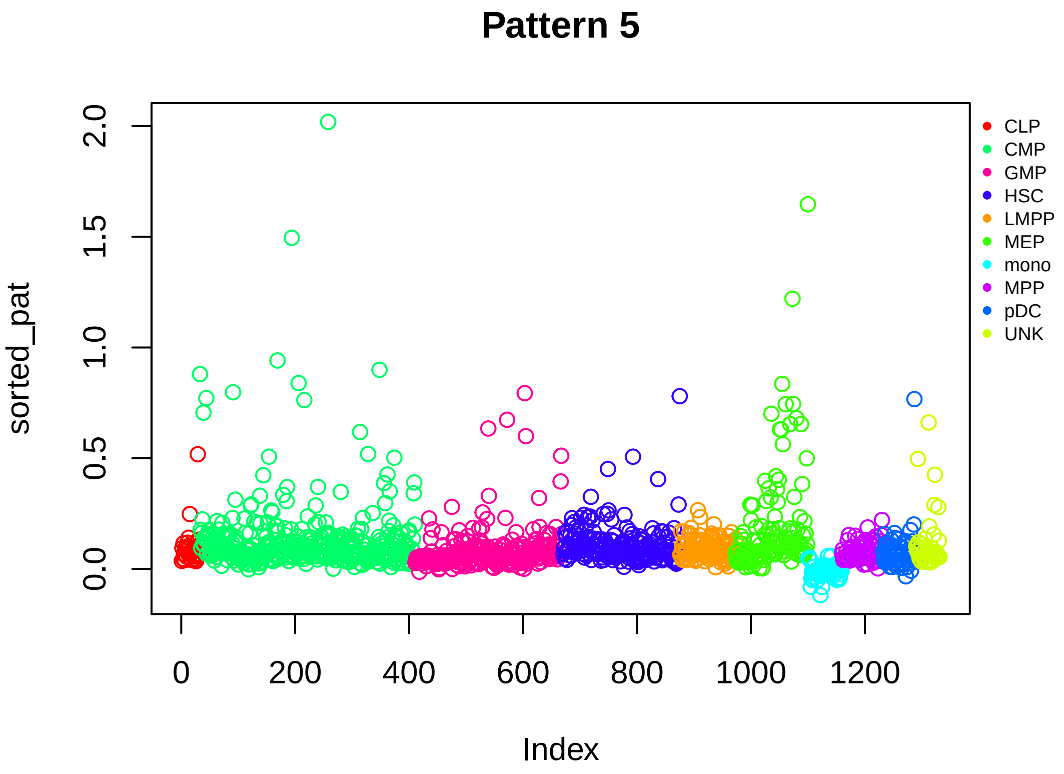
```
cgapsPlot(t(projectRResults$projection), as.factor(BuenrostroCellTypes), matrix = TRUE)
```

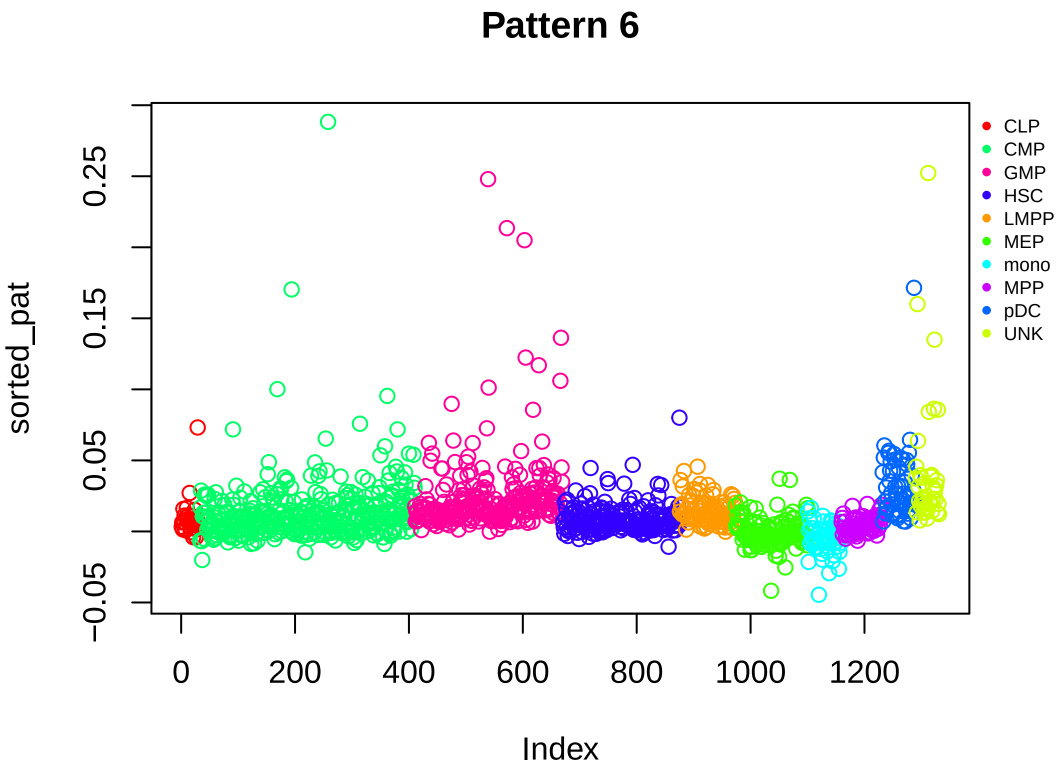


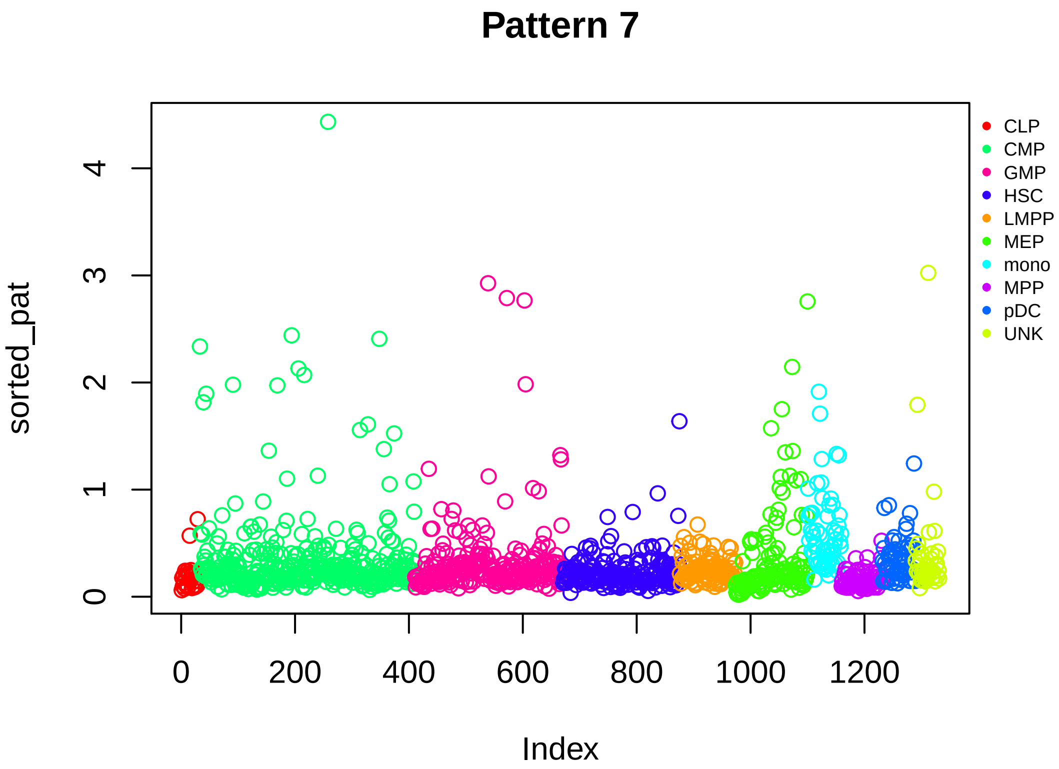

Pattern 2
sorted_pat
0.4
0.2
0.0
−0.2
−0.4
0
200
400
600
800
1000
1200
Index
CLP
CMP
GMP
HSC
LMPP
MEP
mono
MPP
pDC
UNK


Pattern 3
sorted_pat
0.0
0.2
0.4
0.6
0.8
Index
0
200
400
600
800
1000
1200
● CLP
● CMP
● GMP
● HSC
● LMPP
● MEP
● mono
● MPP
● pDC
● UNK


Pattern 4
sorted_pat
0.0
0.4
0.8
1.2
Index
0
200
400
600
800
1000
1200
CLP
CMP
GMP
HSC
LMPP
MEP
mono
MPP
pDC
UNK


Pattern 5
sorted_pat
0.0
0.5
1.0
1.5
2.0
0
200
400
600
800
1000
1200
Index
● CLP
● CMP
● GMP
● HSC
● LMPP
● MEP
● mono
● MPP
● pDC
● UNK


Pattern 6
sorted_pat
Index
0.25
0.15
0.05
−0.05
0
200
400
600
800
1000
1200
● CLP
● CMP
● GMP
● HSC
● LMPP
● MEP
● mono
● MPP
● pDC
● UNK



We see in pattern1 that there is correspondence between the Erythroleukemia pattern and Megakaryocyte-Erythrocyte Progenitors, which makes some sense as we would expect there to be some similiarities between Erythrocyte progneitors and Erythroleukemia.

In the B-cell derived LCL pattern (pattern3) we see strong activation of Common Lymphoid progenitors and Dendritic cells.

In pattern 7 (monocyte) we see strongest signal in the moncytes in the target data set. This may be difficult to determine visually, to confirm we can perform a Wilcoxon Rank Sum test.

```
pairwise.wilcox.test(projectRResults$projection[7,], BuenrostroCellTypes, p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  projectRResults$projection[7, ] and BuenrostroCellTypes
##
##      CLP     CMP     GMP     HSC     LMPP    MEP     mono    MPP
## CMP  0.00019 -       -       -       -       -       -       -
## GMP  5.1e-05 0.79073 -       -       -       -       -       -
## HSC  0.02140 0.00032 0.00017 -       -       -       -       -
## LMPP 5.3e-05 0.79073 0.94658 0.00100 -       -       -       -
## MEP  0.12105 0.01257 0.00959 0.79073 0.02485 -       -       -
## mono 1.3e-11 7.3e-15 3.8e-15 < 2e-16 7.8e-15 6.6e-11 -       -
## MPP  0.39297 1.7e-12 2.4e-13 2.2e-07 1.4e-11 0.00084 < 2e-16 -
## pDC  1.4e-06 0.00408 0.00356 6.2e-07 0.00356 0.00063 7.2e-06 1.8e-12
## UNK  6.8e-05 0.23074 0.26183 0.00110 0.31518 0.01539 6.2e-07 1.3e-08
##      pDC
## CMP  -
## GMP  -
## HSC  -
## LMPP -
## MEP  -
## mono -
## MPP  -
## pDC  -
## UNK  0.23074
##
## P value adjustment method: BH
```

And we observe that monocytes in the target dataset are most siginificantly associated with the monocyte pattern.

Session Info

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] BSgenome.Hsapiens.UCSC.hg19_1.4.0
##  [2] BSgenome_1.52.0
##  [3] rtracklayer_1.44.4
##  [4] Biostrings_2.52.0
##  [5] XVector_0.24.0
##  [6] dplyr_0.8.3
##  [7] Homo.sapiens_1.3.1
##  [8] TxDb.Hsapiens.UCSC.hg19.knownGene_3.2.2
##  [9] org.Hs.eg.db_3.8.2
## [10] GO.db_3.8.2
## [11] OrganismDbi_1.26.0
## [12] GenomicFeatures_1.36.4
## [13] GenomicRanges_1.36.1
## [14] GenomeInfoDb_1.20.0
## [15] AnnotationDbi_1.46.1
## [16] IRanges_2.18.2
## [17] S4Vectors_0.22.1
## [18] Biobase_2.44.0
## [19] BiocGenerics_0.30.0
## [20] ATACCoGAPS_0.90.2
## [21] CoGAPS_3.5.13
##
## loaded via a namespace (and not attached):
##   [1] backports_1.1.5             chromVAR_1.6.0
##   [3] VGAM_1.1-1                  NMF_0.21.0
##   [5] plyr_1.8.4                  lazyeval_0.2.2
##   [7] splines_3.6.1               BiocParallel_1.18.1
##   [9] gridBase_0.4-7              ggplot2_3.2.1
##  [11] TFBSTools_1.22.0            digest_0.6.21
##  [13] foreach_1.4.7               htmltools_0.4.0
##  [15] gdata_2.18.0                magrittr_1.5
##  [17] memoise_1.1.0               JASPAR2016_1.12.0
##  [19] cluster_2.1.0               doParallel_1.0.15
##  [21] ROCR_1.0-7                  limma_3.40.6
##  [23] readr_1.3.1                 annotate_1.62.0
##  [25] matrixStats_0.55.0          GeneOverlap_1.20.0
##  [27] R.utils_2.9.0               prettyunits_1.0.2
##  [29] colorspace_1.4-1            blob_1.2.0
##  [31] xfun_0.10                   crayon_1.3.4
##  [33] RCurl_1.95-4.12             jsonlite_1.6
##  [35] graph_1.62.0                TFMPvalue_0.0.8
##  [37] zeallot_0.1.0               iterators_1.0.12
##  [39] glue_1.3.1                  registry_0.5-1
##  [41] gtable_0.3.0                zlibbioc_1.30.0
##  [43] DelayedArray_0.10.0         R.cache_0.13.0
##  [45] Rhdf5lib_1.6.2              SingleCellExperiment_1.6.0
##  [47] scales_1.0.0                msigdbr_7.0.1
##  [49] rngtools_1.4                DBI_1.0.0
##  [51] bibtex_0.4.2                miniUI_0.1.1.1
##  [53] Rcpp_1.0.2                  viridisLite_0.3.0
##  [55] xtable_1.8-4                progress_1.2.2
##  [57] bit_1.1-14                  DT_0.9
##  [59] htmlwidgets_1.5.1           httr_1.4.1
##  [61] gplots_3.0.1.1              RColorBrewer_1.1-2
##  [63] pkgconfig_2.0.3             XML_3.98-1.20
##  [65] R.methodsS3_1.7.1           tidyselect_0.2.5
##  [67] rlang_0.4.0                 reshape2_1.4.3
##  [69] later_1.0.0                 munsell_0.5.0
##  [71] tools_3.6.1                 DirichletMultinomial_1.26.0
##  [73] RSQLite_2.1.2               evaluate_0.14
##  [75] stringr_1.4.0               projectR_1.0.0
##  [77] fastmap_1.0.1               yaml_2.2.0
##  [79] knitr_1.25                  bit64_0.9-7
##  [81] caTools_1.17.1.2            purrr_0.3.2
##  [83] KEGGREST_1.24.1             RBGL_1.60.0
##  [85] mime_0.7                    R.oo_1.22.0
##  [87] poweRlaw_0.70.2             biomaRt_2.40.5
##  [89] compiler_3.6.1              plotly_4.9.0
##  [91] curl_4.2                    png_0.1-7
##  [93] tibble_2.1.3                stringi_1.4.3
##  [95] lattice_0.20-38             CNEr_1.20.0
##  [97] Matrix_1.2-17               vctrs_0.2.0
##  [99] pillar_1.4.2                lifecycle_0.1.0
## [101] BiocManager_1.30.7          data.table_1.12.4
## [103] bitops_1.0-6                httpuv_1.5.2
## [105] R6_2.4.0                    promises_1.1.0
## [107] KernSmooth_2.23-15          codetools_0.2-16
## [109] gtools_3.8.1                assertthat_0.2.1
## [111] seqLogo_1.50.0              rhdf5_2.28.1
## [113] SummarizedExperiment_1.14.1 pkgmaker_0.27
## [115] withr_2.1.2                 GenomicAlignments_1.20.1
## [117] Rsamtools_2.0.3             GenomeInfoDbData_1.2.1
## [119] hms_0.5.1                   repmis_0.5
## [121] motifmatchr_1.6.0           grid_3.6.1
## [123] tidyr_1.0.0                 rmarkdown_1.16
## [125] shiny_1.4.0
```