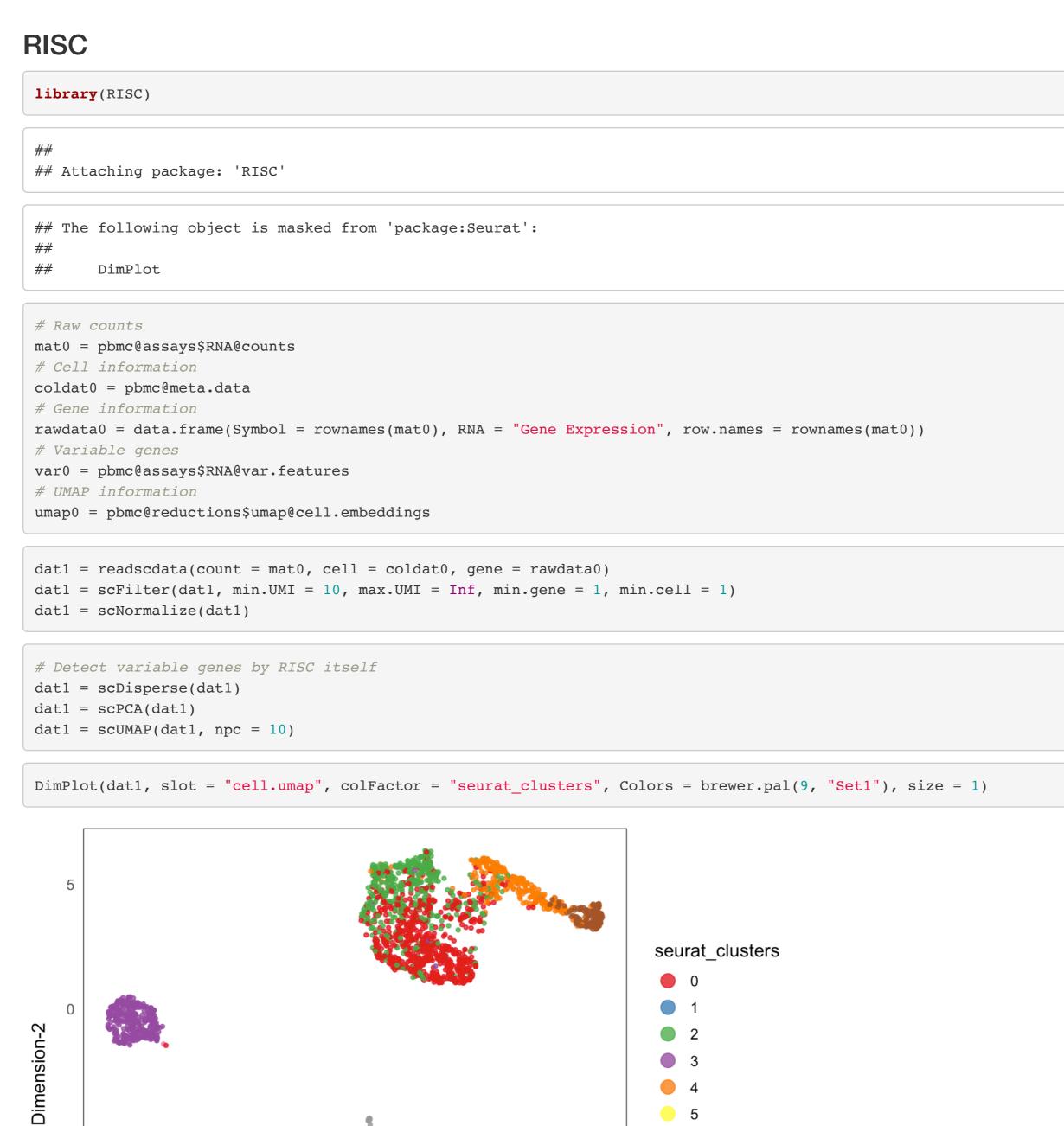
Seurat_to_RISC Yang Liu, Tao Wang, Deyou Zheng 6/27/2021 R Markdown We strongly suggest to clean the gene-cell matrix when using RISC, as shown in the vignettes. The RISC is based on the gene expression variance which could be severely distorted by abnormal genes with extremely large number of UMIs. Here we show how to transfer Seurat object to RISC object without filtering abnormal genes, we get the consistent result. The data can be download from Seurat website: https://satijalab.org/seurat/articles/pbmc3k_tutorial.html The analysis is completely based on Seurat's parameters and cell-population information, not optimized for RISC. Seurat v4 library(dplyr) ## Attaching package: 'dplyr' ## The following objects are masked from 'package:stats': filter, lag ## The following objects are masked from 'package:base': intersect, setdiff, setequal, union library(Seurat) ## Attaching SeuratObject library(patchwork) library(ggplot2) library(RColorBrewer) ## Warning: Feature names cannot have underscores ('_'), replacing with dashes ## ('-') pbmc[["percent.mt"]] = PercentageFeatureSet(pbmc, pattern = "^MT-") VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3) # plot1 = FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt") # plot2 = FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA") # plot1 + plot2 nCount_RNA nFeature_RNA percent.mt 15000 20 3000 15 10000 2000 10 5000 1000 Identity Identity Identity pbmc = NormalizeData(pbmc) pbmc = FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000) all.genes = rownames(pbmc) pbmc = ScaleData(pbmc, features = all.genes) ## Centering and scaling data matrix pbmc = RunPCA(pbmc, features = VariableFeatures(object = pbmc)) ## PC_ 1 ## Positive: CST3, TYROBP, LST1, AIF1, FTL, FTH1, LYZ, FCN1, S100A9, TYMP FCER1G, CFD, LGALS1, S100A8, CTSS, LGALS2, SERPINA1, IFITM3, SPI1, CFP PSAP, IFI30, SAT1, COTL1, S100A11, NPC2, GRN, LGALS3, GSTP1, PYCARD ## Negative: MALAT1, LTB, IL32, IL7R, CD2, B2M, ACAP1, CD27, STK17A, CTSW CD247, GIMAP5, AQP3, CCL5, SELL, TRAF3IP3, GZMA, MAL, CST7, ITM2A MYC, GIMAP7, HOPX, BEX2, LDLRAP1, GZMK, ETS1, ZAP70, TNFAIP8, RIC3 ## PC 2 ## Negative: NKG7, PRF1, CST7, GZMB, GZMA, FGFBP2, CTSW, GNLY, B2M, SPON2 CCL4, GZMH, FCGR3A, CCL5, CD247, XCL2, CLIC3, AKR1C3, SRGN, HOPX TTC38, APMAP, CTSC, S100A4, IGFBP7, ANXA1, ID2, IL32, XCL1, RHOC ## PC 3 PLAC8, BLNK, MALAT1, SMIM14, PLD4, LAT2, IGLL5, P2RX5, SWAP70, FCGR2B ## Negative: PPBP, PF4, SDPR, SPARC, GNG11, NRGN, GP9, RGS18, TUBB1, CLU HIST1H2AC, AP001189.4, ITGA2B, CD9, TMEM40, PTCRA, CA2, ACRBP, MMD, TREML1 ## PC 4 GP9, AP001189.4, CA2, PTCRA, CD9, NRGN, RGS18, GZMB, CLU, TUBB1 AQP3, CD2, CD14, FYB, LGALS2, GIMAP4, ANXA1, CD27, FCN1, RBP7 LYZ, S100A11, GIMAP5, MS4A6A, S100A12, FOLR3, TRABD2A, AIF1, IL8, IF16 ## PC 5 ## Positive: GZMB, NKG7, S100A8, FGFBP2, GNLY, CCL4, CST7, PRF1, GZMA, SPON2 GZMH, S100A9, LGALS2, CCL3, CTSW, XCL2, CD14, CLIC3, S100A12, CCL5 RBP7, MS4A6A, GSTP1, FOLR3, IGFBP7, TYROBP, TTC38, AKR1C3, XCL1, HOPX

pbmc.data = Read10X(data.dir = "/Users/yanliu/Desktop/PBMC_3k") # change directory path to your own pbmc = CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200) pbmc = subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)</pre> ## Positive: CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1, HLA-DRA, LINC00926, CD79B, HLA-DRB1, CD74 HLA-DMA, HLA-DPB1, HLA-DQA2, CD37, HLA-DRB5, HLA-DMB, HLA-DPA1, FCRLA, HVCN1, LTB BLNK, P2RX5, IGLL5, IRF8, SWAP70, ARHGAP24, FCGR2B, SMIM14, PPP1R14A, C16orf74 ## Positive: HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPB1, HLA-DPA1, CD74, MS4A1, HLA-DRB1, HLA-DRA HLA-DRB5, HLA-DQA2, TCL1A, LINC00926, HLA-DMB, HLA-DMA, CD37, HVCN1, FCRLA, IRF8 NGFRAP1, F13A1, SEPT5, RUFY1, TSC22D1, MPP1, CMTM5, RP11-367G6.3, MYL9, GP1BA ## Positive: HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1, CD74, HLA-DPB1, HIST1H2AC, PF4, TCL1A SDPR, HLA-DPA1, HLA-DRB1, HLA-DQA2, HLA-DRA, PPBP, LINC00926, GNG11, HLA-DRB5, SPARC ## Negative: VIM, IL7R, S100A6, IL32, S100A8, S100A4, GIMAP7, S100A10, S100A9, MAL ## Negative: LTB, IL7R, CKB, VIM, MS4A7, AQP3, CYTIP, RP11-290F20.3, SIGLEC10, HMOX1 PTGES3, LILRB2, MAL, CD27, HN1, CD2, GDI2, ANXA5, CORO1B, TUBA1B FAM110A, ATP1A1, TRADD, PPA1, CCDC109B, ABRACL, CTD-2006K23.1, WARS, VMO1, FYB VizDimLoadings(pbmc, dims = 1:2, reduction = "pca") DimPlot(pbmc, reduction = "pca") DimHeatmap(pbmc, dims = 1, cells = 500, balanced = TRUE) **TYROBP** HLA-DQA1 HLA-DQB1 HLA-DRA INC00926 CD79B LGALS3 GSTP1 MALAT1 PRF1 NKG7 -0.10 -0.05 0.00 0.05 0.10 PC_1 -0.1 0.0 0.1 PC_2 10 pbmc3k -10 10 0 PC_1 PC₁ MALAT1 LTB IL32 IL7R CD2 B2M ACAP1 CD27 STK17A **CTSW** CD247 GIMAP5 AQP3 CCL5 SELL CTSS S100A8 LGALS1 CFD FCER1G TYMP S100A9 FCN1 LYZ FTH1 FTL AIF1 LST1 **TYROBP** CST3 pbmc = FindNeighbors(pbmc, dims = 1:10) ## Computing nearest neighbor graph ## Computing SNN pbmc = FindClusters(pbmc, resolution = 0.5) ## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck ## Number of nodes: 2638 ## Number of edges: 95965 ## Running Louvain algorithm... ## Maximum modularity in 10 random starts: 0.8723 ## Number of communities: 9 ## Elapsed time: 0 seconds pbmc = RunUMAP(pbmc, dims = 1:10) ## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R-native UW OT using the cosine metric ## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation' ## This message will be shown once per session ## 06:52:45 UMAP embedding parameters a = 0.9922 b = 1.112 ## 06:52:45 Read 2638 rows and found 10 numeric columns ## 06:52:45 Using Annoy for neighbor search, n_neighbors = 30 ## 06:52:45 Building Annoy index with metric = cosine, n_trees = 50 ## 0% 10 20 30 40 50 60 70 80 90 100% ## [----|----|----| ## ************* ## 06:52:45 Writing NN index file to temp file /var/folders/lw/yxjg3xc97j5gw4p2dhkzr_1m0000gn/T//Rtmp6KA9tv/file7 87f581a4198 ## 06:52:45 Searching Annoy index using 1 thread, search_k = 3000 ## 06:52:46 Annoy recall = 100% ## 06:52:46 Commencing smooth kNN distance calibration using 1 thread ## 06:52:46 Initializing from normalized Laplacian + noise ## 06:52:47 Commencing optimization for 500 epochs, with 105124 positive edges ## 06:52:50 Optimization finished



5

DimPlot(dat1, slot = "cell.umap", colFactor = "seurat_clusters", Colors = brewer.pal(9, "Set1"), size = 1)

7

8

seurat_clusters

10

DimPlot(pbmc, reduction = "umap", cols = brewer.pal(9, "Set1"))

15 -

10 ·

0 .

-10

10

-10

dat1 = scDisperse(dat1)

dat1@vargene = var0 dat1 = scPCA(dat1)

Use Seurat's variable genes

dat1 = scUMAP(dat1, npc = 10)

-5

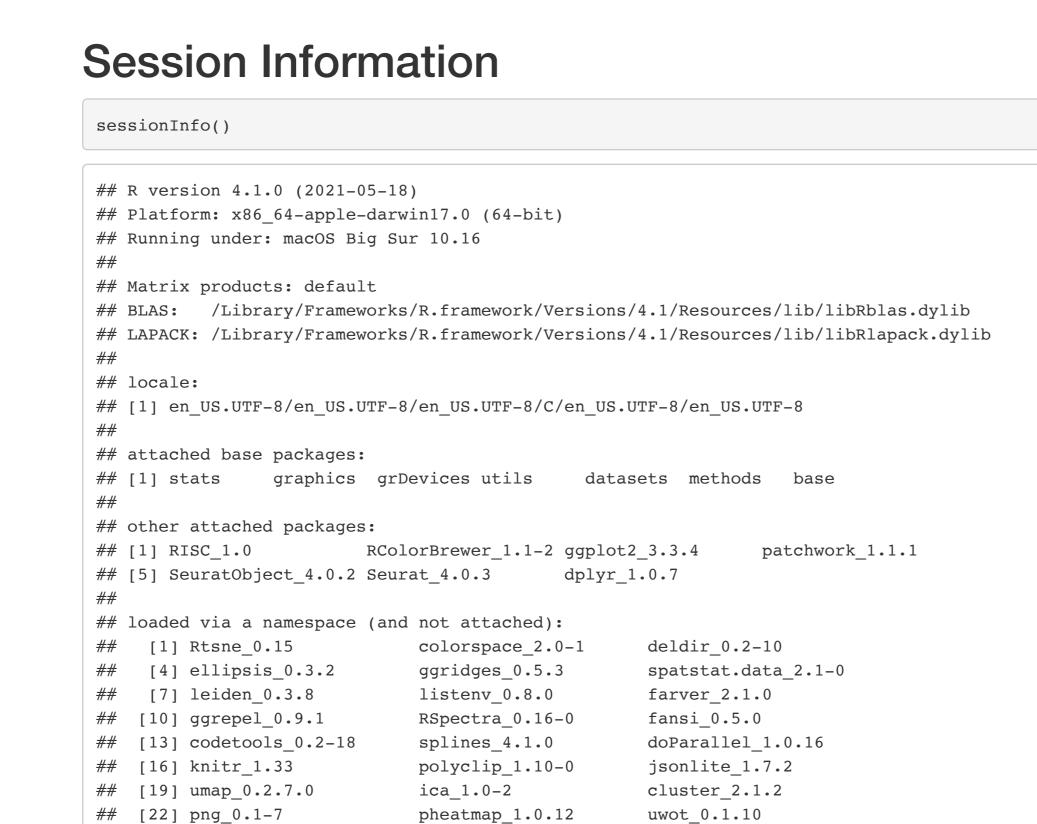
Dimension-1

-10

-5

UMAP_1





sctransform_0.3.2

htmltools_0.5.1.1

httr_1.4.2

fastmap_1.1.0

gtable_0.3.0

reshape2_1.4.4

stringr_1.4.0

miniUI_0.1.1.1

goftest_1.2-2

zoo_1.8-9

yaml_2.2.1

gridExtra_2.3

stringi_1.6.2

ROCR_1.0-11

plyr_1.8.6

withr_2.4.2

utf8_1.2.1

FNN_1.1.3

rmarkdown_2.9

xtable_1.8-4

openssl_1.4.4

bslib_0.2.5.1

jquerylib_0.1.4

iterators_1.0.13

promises_1.2.0.1

densityClust_0.3

matrixStats_0.59.0

htmlwidgets_1.5.3

tidyselect_1.1.1

generics_0.1.0

survival_3.2-11

future.apply_1.7.0

spatstat.sparse_2.0-0

assertthat_0.2.1

lazyeval_0.2.2

tools_4.1.0

glue_1.4.2

Rcpp_1.0.6

vctrs_0.3.8

lmtest_0.9-38

globals_0.14.0 lifecycle_1.0.0

future_1.21.0

scales_1.1.1

sass_0.4.0 highr 0.9

rlang_0.4.11

evaluate_0.14

labeling_0.4.2

magrittr_2.0.1

parallelly_1.26.0

spatstat.geom_2.2-0

purrr_0.3.4

DBI_1.1.1

mgcv_1.8-36

 $abind_1.4-5$

crayon_1.4.1

digest_0.6.27

munsell_0.5.0

tidyr_1.1.3

askpass_1.1

grid_4.1.0

reticulate_1.20

spatstat.utils_2.2-0

5

Dimension-1

10

-5

-10

[25] shiny_1.6.0

[28] compiler_4.1.0

[34] later_1.2.0

[40] RANN_2.6.1

[46] nlme_3.1-152

[49] xfun_0.24

[55] irlba_2.3.3

[58] MASS_7.3-54

[64] parallel_4.1.0

[67] pbapply_1.4-3

[70] rpart_4.1-15

[73] foreach_1.5.1

[82] tensor_1.5

[91] R6_2.5.0

[94] pillar_1.6.1

[100] tibble_3.1.2

[106] plotly_4.9.4.1

[76] pkgconfig_2.0.3

[79] lattice_0.20-44

[85] cowplot_1.1.1

[88] RcppAnnoy_0.0.18

[97] fitdistrplus_1.1-5

[103] KernSmooth_2.23-20

[109] data.table_1.14.0

[118] viridisLite_0.4.0

[112] pbmcapply_1.5.0

[115] httpuv_1.6.1

[52] mime_0.11

[31] Matrix_1.3-4

[37] igraph_1.2.6

[43] scattermore 0.7

[61] spatstat.core_2.2-0