# Smmit: Multi-sample single-cell multi-omics integration

Changxin Wan, Program of Computational Biology and Bioinformatics, Duke University School of Medi
Zhicheng Ji, Department of Biostatistics and Bioinformatics, Duke University School of Medicine

## Introductions

Smmit performs integration both across samples and modalities to produce a single UMAP space. It first uses harmony to integrate across samples and then uses Seurat weighted nearest neighbor function to integrate across modalities.

## Load package

We first load the Smmit package. We also load the Seurat package for visualization.

```
library(Smmit)
library(Seurat)
```

## Example data that jointly profiles gene expression and protein abundances

The first example dataset is a CITE-seq dataset that jointly profiles gene expression and protein abundances. The dataset is a subset of a processed CITE-seq dataset downloaded from Gene Expression Omnibus GSE100866 and was from the original publication of Stoeckius et al., 2017, Nature Methods. The dataset contains a human peripheral blood mononuclear cells (PBMC) sample and a human cord blood mononuclear cells (CBMC) sample. The data have already been loaded in Seurat using the standard Seurat pipeline for CITE-seq data.

We first read in the two Seurat objects for the two samples:

```
cbmc <- readRDS(paste0(system.file('data',package = 'Smmit'),'/RNA_ADT/cbmc.rds'))
pbmc <- readRDS(paste0(system.file('data',package = 'Smmit'),'/RNA_ADT/pbmc.rds'))
```

We then combine the two objects into a list:

```
obj <- list(cbmc=cbmc,pbmc=pbmc)
```

We then run smmit using the RNA_ADT mode.

```
obj <- smmit(obj,mode='RNA_ADT')
```

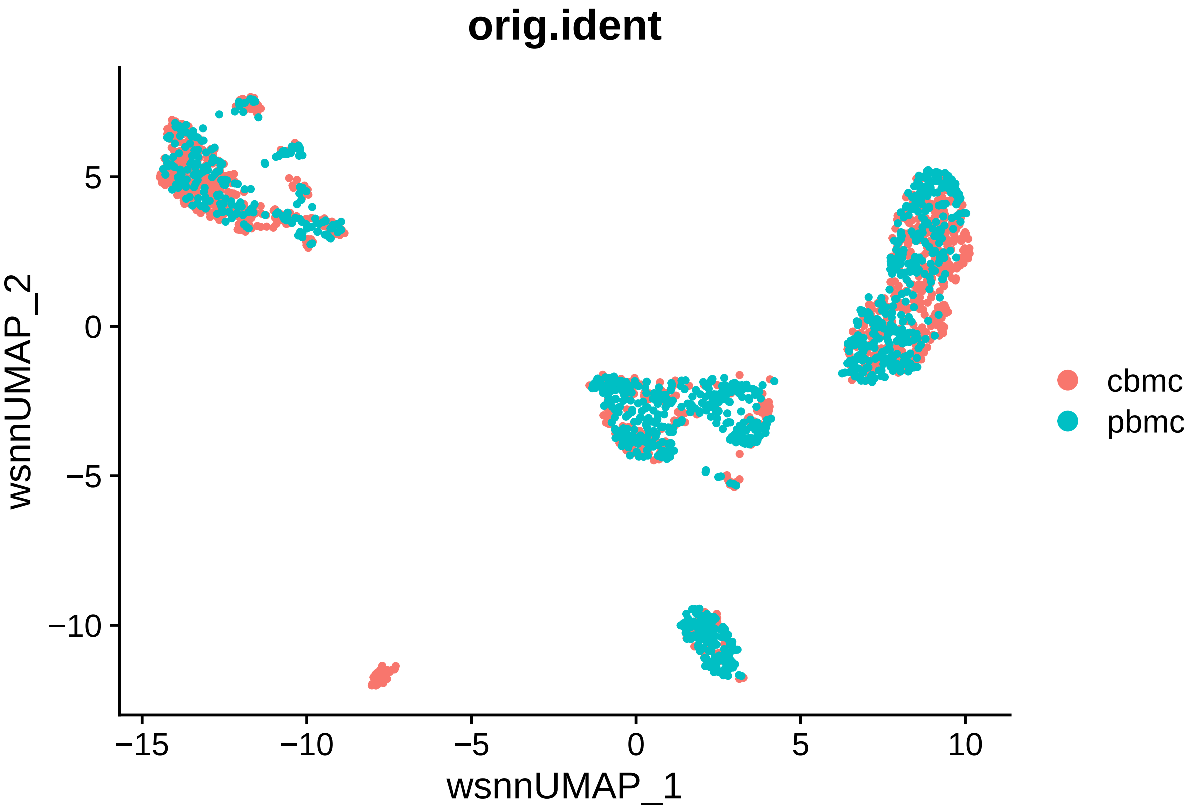
Smmit returns a single Seurat object with the UMAP space integrating both samples and modalities. The integrated UMAP space is stored in 'wsnnumap':

```
obj
```

```
## An object of class Seurat
## 21076 features across 2100 samples within 2 assays
## Active assay: ADT (8 features, 8 variable features)
##  1 other assay present: RNA
##  5 dimensional reductions calculated: pca, integrated_rna, apca, integrated_adt, wsnnumap
```

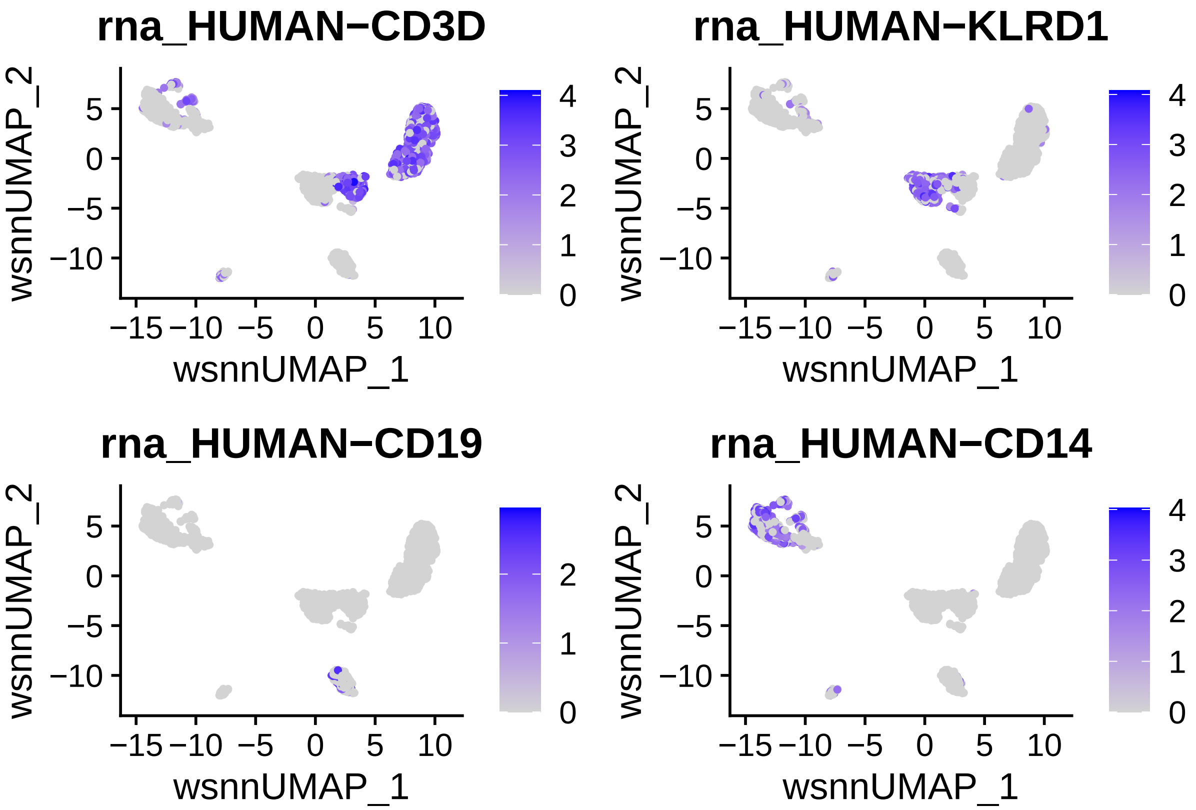
We can visualize the distribution of cells from the two samples. It seems that cells from two samples are mixed well:

```
DimPlot(obj,reduction = 'wsnnumap',group.by='orig.ident')
```



Finally we can visualize the expression of marker genes for major PBMC cell types. It seems that different cell types are separated in the UMAP space:

```
FeaturePlot(obj,reduction = 'wsnnumap',feature=c('HUMAN-CD3D','HUMAN-KLRD1','HUMAN-CD19','HUMAN-CD14'))
```



## Example data that jointly profiles gene expression and chromatin accessibility

The second example dataset is a single-cell multi-omics dataset that jointly profiles gene expression and chromatin accessibility. The dataset is a subset of a single-cell multi-omics dataset downloaded from the 10x website. The dataset contains a male PBMC sample and a female PBMC sample. The data have already been loaded in Seurat using the standard Signac pipeline for single-cell multi-omics data.

We first read in the two Seurat objects for the two samples:

```
male <- readRDS(paste0(system.file('data',package = 'Smmit'),'/RNA_ATAC/male.rds'))
female <- readRDS(paste0(system.file('data',package = 'Smmit'),'/RNA_ATAC/female.rds'))
```

We then combine the two objects into a list:

```
obj <- list(male=male,female=female)
```

We then run smmit using the RNA_ATAC mode.

```
obj <- smmit(obj,mode='RNA_ATAC')
```

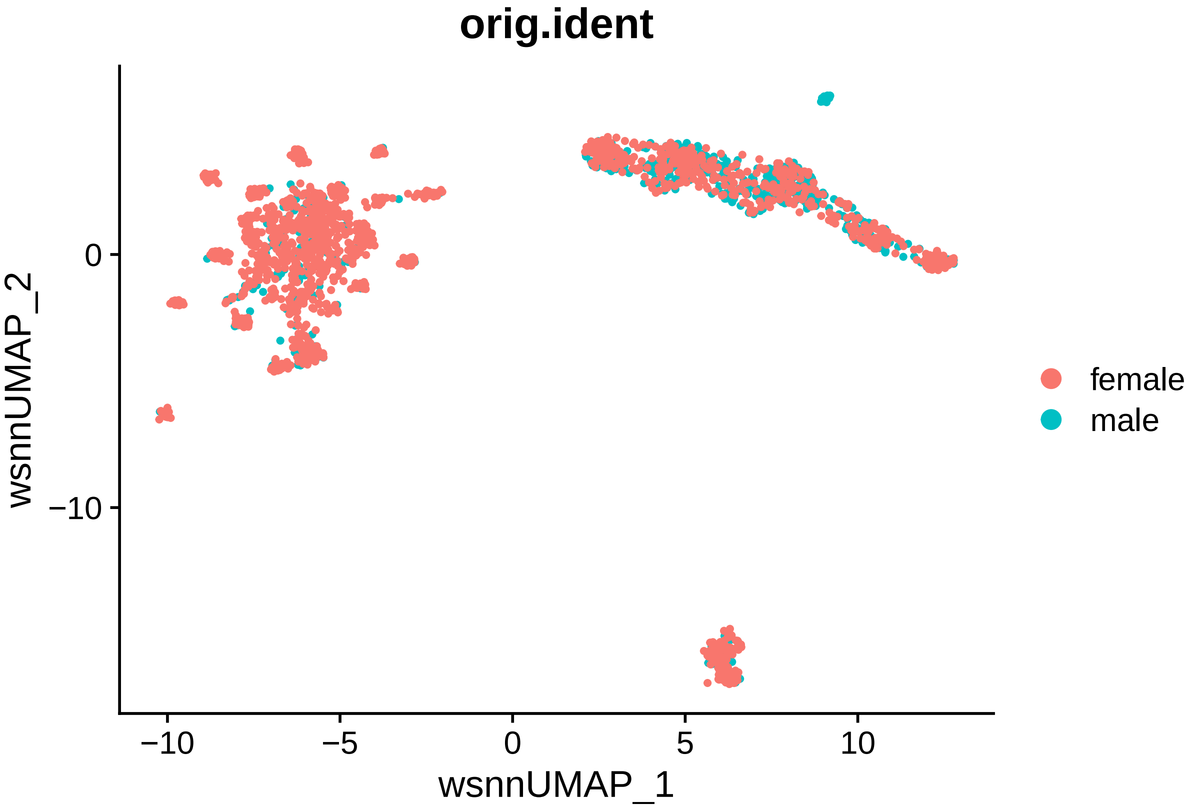
Smmit returns a single Seurat object with the UMAP space integrating both samples and modalities. The integrated UMAP space is stored in ‘wsnnumap’:

```
obj
```

```
## An object of class Seurat
## 150056 features across 2101 samples within 2 assays
## Active assay: ATAC (127933 features, 127933 variable features)
##  1 other assay present: RNA
##  5 dimensional reductions calculated: pca, integrated_rna, lsi, integrated_atac, wsnnumap
```

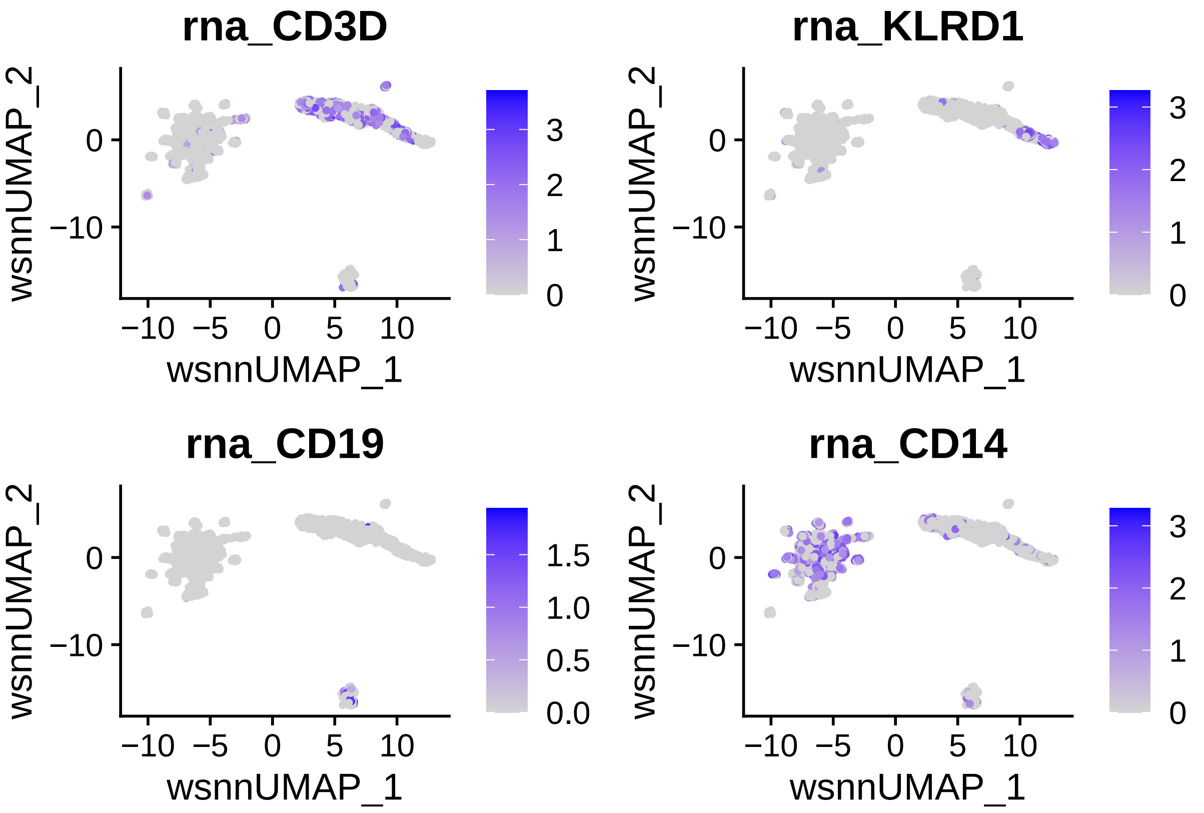
We can visualize the distribution of cells from the two samples. It seems that cells from two samples are mixed well:

```
DimPlot(obj,reduction = 'wsnnumap',group.by='orig.ident')
```



Finally we can visualize the expression of marker genes for major PBMC cell types. It seems that different cell types are separated in the UMAP space:

```
FeaturePlot(obj,reduction = 'wsnnumap',feature=c('CD3D','KLRD1','CD19','CD14'))
```



### Session Info

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] Signac_1.9.0       SeuratObject_4.1.3 Seurat_4.3.0       Smmit_1.0
##
## loaded via a namespace (and not attached):
##   [1] Rtsne_0.16             colorspace_2.0-3       deldir_1.0-6
##   [4] ellipsis_0.3.2         ggridges_0.5.4         XVector_0.36.0
##   [7] GenomicRanges_1.48.0   spatstat.data_3.0-1    farver_2.1.1
##  [10] leiden_0.4.3           listenv_0.9.0          ggrepel_0.9.1
##  [13] fansi_1.0.3            codetools_0.2-18       splines_4.2.1
##  [16] knitr_1.42             RcppRoll_0.3.0         polyclip_1.10-4
##  [19] jsonlite_1.8.3         Rsamtools_2.12.0       ica_1.0-3
##  [22] cluster_2.1.3          png_0.1-7              uwot_0.1.14
##  [25] shiny_1.7.4            sctransform_0.3.5      spatstat.sparse_3.0-1
##  [28] compiler_4.2.1         httr_1.4.4             assertthat_0.2.1
##  [31] Matrix_1.5-3           fastmap_1.1.0          lazyeval_0.2.2
##  [34] cli_3.4.1              later_1.3.0            htmltools_0.5.5
##  [37] tools_4.2.1            igraph_1.3.5           GenomeInfoDbData_1.2.8
##  [40] gtable_0.3.1           glue_1.6.2             RANN_2.6.1
##  [43] reshape2_1.4.4         dplyr_1.0.10           fastmatch_1.1-3
##  [46] Rcpp_1.0.9             scattermore_0.8        Biostrings_2.64.1
##  [49] vctrs_0.5.0            spatstat.explore_3.1-0 nlme_3.1-157
##  [52] progressr_0.13.0       lmtest_0.9-40          spatstat.random_3.1-4
##  [55] xfun_0.38              stringr_1.4.1          globals_0.16.2
##  [58] mime_0.12              miniUI_0.1.1.1         lifecycle_1.0.3
##  [61] irlba_2.3.5.1          goftest_1.2-3          future_1.32.0
##  [64] zlibbioc_1.42.0        MASS_7.3-57            zoo_1.8-11
##  [67] scales_1.2.1           promises_1.2.0.1       spatstat.utils_3.0-2
##  [70] parallel_4.2.1         RColorBrewer_1.1-3     yaml_2.3.6
##  [73] reticulate_1.26        pbapply_1.7-0          gridExtra_2.3
##  [76] ggplot2_3.3.6          stringi_1.7.8          highr_0.10
##  [79] S4Vectors_0.34.0       harmony_0.1.1          BiocGenerics_0.42.0
##  [82] BiocParallel_1.30.4    GenomeInfoDb_1.32.4    bitops_1.0-7
##  [85] rlang_1.0.6            pkgconfig_2.0.3        matrixStats_0.62.0
##  [88] evaluate_0.19          lattice_0.20-45        ROCR_1.0-11
##  [91] purrr_0.3.5            tensor_1.5             labeling_0.4.2
##  [94] patchwork_1.1.2        htmlwidgets_1.5.4      cowplot_1.1.1
##  [97] tidyselect_1.2.0       parallelly_1.35.0      RcppAnnoy_0.0.20
## [100] plyr_1.8.7             magrittr_2.0.3         R6_2.5.1
## [103] IRanges_2.30.1         generics_0.1.3         DBI_1.1.3
## [106] withr_2.5.0            pillar_1.8.1           fitdistrplus_1.1-8
## [109] RCurl_1.98-1.9         survival_3.3-1         abind_1.4-5
## [112] sp_1.6-0               tibble_3.1.8           future.apply_1.10.0
## [115] crayon_1.5.2           KernSmooth_2.23-20     utf8_1.2.2
## [118] spatstat.geom_3.1-0    plotly_4.10.0          rmarkdown_2.21
## [121] grid_4.2.1             data.table_1.14.4      digest_0.6.30
## [124] xtable_1.8-4           tidyr_1.2.1            httpuv_1.6.9
## [127] stats4_4.2.1           munsell_0.5.0          viridisLite_0.4.1
```