Cottrazm: Construct Tumor Transition Zone Microenvironment

Zhenzhen Xun, Xinyu Ding, Youqiong Ye#

2022-1-23

1. Introduction

Cottrazm is an R package to construct tumor transition zone microenvironment of spatial transcriptomic (ST) data.

Conttrazm adjusted gene expression with morphological information,[1] and based on morphological adjusted gene expression, Cottrazm applied InferCNV[2] to recognize tumor spots. Then Cottrazm defined a cluster of spots spatially located close to tumor but with different morphology and gene expression pattern form tumor spots, this cluster of spots was defined as tumor boundary. Next based on corresponding single cell sequencing (scRNAseq) data of tumor, by dampened weighted least squares (DWLS), a kind of deconvolution algorithm,[3] Cottrazm calculated the possible proportion of each single cell types of each spatial spot, to further explore the character of tumor boundary as well as tumor and other spots. Then according to deconvolution result, scRNAseq expression data, and ST expression data, Cottrazm applied a novel algorithm to reconstruct the ST feature expression at single cell level, to deeper exploring the microenvironment of tumor transition zone, including specific cells enriched in tumor boundary and their differential expressed genes, potential cell-cell interactions between cells enriched in tumor boundary, and new possibly drug targets.

The full Cottrazm documention is avaliable in the package. To reach the user's guide, install the Cottrazm package and load it into an R session by library (Cottrazm). And they can get help by help (STPreProcess), help (STModiCluster), help (BoundaryDefine), help (SpatialDecon), and help (SpatialRecon) to see documentation of the main function. Also description of each function is including in docs/Cottrazm_0.1.1.pdf.

2. Installation and requirement

Before installing cottrazm, please properly create conda environment install all dependencies

```
# conda create -n TumorBoundary python=3.8
# conda activate TumorBoundary
# pip install -U stlearn
```

```
# library(Seurat)
# library(magrittr)
# library(dplyr)
# library(Matrix)
# library(ggplot2)
# library(stringr)
# library(RColorBrewer)
# library(patchwork)
# library(ggtree)
# library(BiocGenerics)
# library(readr)
# library(rtracklayer)
# library(infercnv)
# library(phylogram)
# library(utils)
# library(dendextend)
# library(assertthat)
# library(reticulate)
# library(openxlsx)
# library(scatterpie)
# library(cowplot)
# library(stats)
# library(quadprog)
# library(data.table)
# library(Rfast)
# library(ggrepel)
# library(tibble)
# library(clusterProfiler)
# library(utils)
# library(org.Hs.eg.db)
```

Installation of Cottrazm You can install Cottrazm.tar.gz from local path and directly from github (https://github.com/Yelab2020/Cottrazm).

```
#devtools::install_github('Yelab2020/Cottrazm')
#install.packages("Cottrazm.tar.gz", repos = NULL, type = "source")
#library(Cottrazm)
```

3. functional modules

3.1 Boundary define

3.1.1 Read, preprocessed and quality control of tumor ST data

We firstly read Spaceranger result, then we examined the quality control result with parameters: nCount_Spatial, nFeature_Spatial and percentage of mitochondria genes. The feature plot and .xlsx file of quality control were saved.

```
# print('STPreProcess')
# InDir = "Spaceranger/outs/"
# Sample = "YourSampleName"
# OutDir = "YourOutDir/"
# TumorST <-
#   STPreProcess(InDir = InDir,
#                OutDir = OutDir,
#                Sample = Sample)
```

3.1.2 Morphological adjusted cluster determination

We firstly use morphological information to adjust spatial feature expression, then the adjusted expression data was used to cluster spatial spots. A list of genes represent lymphocytes was used to score spatial spots to recognize normal spatial cluster, the cluster with the highest score of lymphocytes' features was defined as normal cluster. Then we used the barcode and the result of morphological adjusted cluster to generate cell annotation file for future use in InferCNV.

```
# print('STModiCluster')
# res = 1.5
# TumorST <-
#   STModiCluster(
#     InDir = InDir,
#     Sample = Sample,
#     OutDir = OutDir,
#     TumorST = TumorST,
#     res = res
#   )
```

3.1.3 Run InferCNV on ST data

We firstly use clusters with the highest score of lymphocytes features as reference, because the CNV copy number of lymphocytes were relatively stable. Then we run InferCNV on ST data. You can choose non-adjusted gene expression assay when tissue was properly permeabilized otherwise we recommended use morphological adjusted gene expression assay. At same time, you need to download the gene order file, a .txt file with chromosomal gene order. When running InferCNV, the parameter "analysis_mode" was set as random_trees, the observation spots will be divided to 8 clusters, it will take longer time, but tumor and normal spots will be better separated.

To avoid crashes, before running STCNV, we recommend set:

```
# ulimit -s 102400
```

Then run STCNV in the R script.

```
# print('STCNV')
# STInferCNV <-
#   STCNV(TumorST = TumorST,
#         OutDir  = OutDir,
#         assay = "Spatial")
```

Then we score ST data based on result of InferCNV, visualize CNV scores of each sub-clusters defended by random_tree. Next according to CNV scores, CNV sub-clusters and morphology information, you need to choose tumor sub-clusters.

```
# print('STCNVScore')
#
# infercnv.dend <-
#   read.tree(
#     file =
#       "YourPath/infercnv.17_HMM_predHMMi6.rand_trees.hmm_mode-subclusters.observation
#   )
#
# cnv_table <-
#   read.table(
#     file =
#       'YourPath/infercnv.17_HMM_predHMMi6.rand_trees.hmm_mode-subclusters.observation
#   )
#
# TumorST <-
#   STCNVScore(
#     infercnv.dend = infercnv.dend,
#     cnv_table = cnv_table,
#     TumorST = TumorST,
#     Sample = Sample,
#     OutDir = OutDir
#   )
```

3.1.4 Define of tumor boundary

According to plots above, CNV sub-cluster 1 and 2 had the highest CNV scores and share similar morphological state, so we recognize these sub-clusters as putative tumor cluster. Next, we start define the boundary of putative tumor cluster, the processed plots and data were saved to OutDir:

```
# TumorSTn <-
#   BoundaryDefine(
#     TumorST = TumorST,
#     TumorLabel = c(1,2),
#     OutDir = OutDir,
#     Sample = Sample
#   )
```

After defined the boundary of tumor, we need to project the tumor spots, boundary spots and other spots to spatial HE/IF staining plot. Finally, we generated and saved a boundary defined Seurat object for future analysis.

```
# TumorST <-
#   BoundaryPlot(
#     TumorSTn = TumorSTn,
#     TumorST = TumorST,
#     OutDir = OutDir,
#     Sample = Sample
#   )
```

3.2 Spot Deconvolution

To better understand of cell composition of tumor boundary as well as tumor and out tumor regions, we combined tumor ST data and tumor scRNAseq data and used deconvolution method to estimate the proportion of main cell types in scRNAseq data infiltrated in each spatial spot.

3.2.1 Preporcess of scRNAseq data

Firstly, we need to generate significant gene expression file (sig_exp) and a list of marker features of each scRNAseq cell type (clustermarkers_list)

```
#clustermarkers_list
# sc_obj <- readr::read_rds("YourPath/scRNAseqRef.rds.gz")
# clustermarkers <-
#   Seurat::FindAllMarkers(object = sc_obj,
#                          logfc.threshold = 0.25,
#                          only.pos = TRUE)
#
# clustermarkers_list <- split(clustermarkers, clustermarkers$cluster)
# clustermarkers_list <-
#   lapply(names(clustermarkers_list), function(cluster) {
#     sub_markers <- clustermarkers_list[[cluster]]$gene
#   })
# names(clustermarkers_list) <-
#   names(split(clustermarkers, clustermarkers$cluster))
#
# #sig_exp
# sig_exp <-
#   get_sig_exp(
#     se.obj = sc_obj,
#     DefineTypes = "MajorTypes",
#     sig_scran = unique(unlist(clustermarkers_list))
#   )
```

3.2.2 Enrichment analysis of ST data

Then, we preprocessed ST data and applied PAGE enrichment analysis on ST data to get the enrichment scores of main cell types in scRNAseq data of each spatial spot.

```
# #Processed ST data
# TumorST <- NormalizeData(TumorST, assay = "Spatial")
# TumorST@meta.data$Decon_topics <-
#   paste(TumorST@meta.data$Location,
#         TumorST@meta.data$seurat_clusters,
#         sep = "_")
# expr_values = as.matrix(TumorST@assays$Spatial@data) #ST expr log
# nolog_expr = 2 ^ (expr_values) - 1 #ST expr nolog
# meta_data <-
#   TumorST@meta.data[, c("nCount_Spatial", "Decon_topics", "Location")]
#
# #Signature score
# for (cluster in names(clustermarkers_list)) {
#   cluster_markers = clustermarkers_list[[cluster]][1:25]
#   cluster_score <-
#     apply(TumorST@assays$Spatial@data[rownames(TumorST@assays$Spatial@data) %in% clus
#   meta_data <- cbind(meta_data, cluster_score)
# }
# colnames(meta_data) <-
#   c("nCount_Spaital",
#     "Decon_topics",
#     "Location",
#     names(clustermarkers_list))
#
# #filter st and sc feature
# intersect_gene = intersect(rownames(sig_exp), rownames(nolog_expr))
# filter_sig = sig_exp[intersect_gene, ]
# filter_expr = nolog_expr[intersect_gene, ]
# filter_log_expr = expr_values[intersect_gene, ]
#
# #enrichment analysis
# enrich_matrix <-
#   get_enrich_matrix(filter_sig = filter_sig,
#                     clustermarkers_list = clustermarkers_list)
# enrich_result <-
#   enrich_analysis(filter_log_expr = filter_log_expr,
#                   enrich_matrix = enrich_matrix)
```

3.2.3 Spot deconvolution

Next, we applied deconvolution analysis of ST spots expression use dampened weighted least squares (DWLS) to explore the infiltration proportion of main scRNAseq cell type in each spot.

```
# DeconData <- SpatialDecon(enrich_matrix = enrich_matrix,
#                           enrich_result = enrich_result,
#                           filter_expr = filter_expr,
#                           filter_sig = filter_sig,
#                            clustermarkers_list = clustermarkers_list,
#                            meta_data = meta_data,
#                            malignant_cluster = "Malignant epithelial cells",
#                            tissue_cluster = "Epithelial cells",
#                            stromal_cluster = "Fibroblast cells")
#
# print(head(DeconData))
```

3.2.4 Visualization

After getting the deconvolution result, we need to visualize the result with pie plot and bar plot. Pie plot was used to project the main cell type infiltration of each spot to spatial. Bar plot was used to compare cell type infiltration of tumor, tumor boundary and other out tumor spots.

```
# plot_col <- colnames(DeconData)[2:ncol(DeconData)]
# img_path = "inst/extdata/outs/spatial/tissue_lowres_image.png"
# DeconPieplot(
#   DeconData = DeconData,
#   TumorST = TumorST,
#   plot_col = plot_col,
#   img_path = img_path,
#   pie_scale = 0.4,
#   scatterpie_alpha = 0.8,
#   border_color = "grey"
# )
#
# DeconBarplot(DeconData = DeconData,
#              TumorST = TumorST,
#              plot_col = plot_col)
```

3.3 Reconstruct of spatial tumor microenvironment

Based on tumor boundary define result and spatial deconvolution result, we eager to reconstruct of spatial tumor microenrironment at almost single cell level, which made up for the problem of insufficient resolution of ST data.

3.3.1 Reconstruction of spatial gene expression matrix for sub-spots

Finally, we constructed the ST feature expression at single cell level based on ST data, scRNAseq data and spot deconvolution result, which could be used for further analysis. In this vignette, we only reconstruct the TME of tumor boundary, which was the region most likely associated with tumor progression, drug resistance and tumor metastasis. If you want to reconstruct TME of whole ST data, you can change parameter Location = c("Tumor","Trans","OutSpots"), but it will take longer time.

```
# TumorSTRecon <-
#   SpatialRecon(
#     TumorST = TumorST,
#     sig_exp = sig_exp,
#     clustermarkers_list = clustermarkers_list,
#     DeconData = DeconData,
#     Location = c("Trans")
#   )
```

3.3.2 Differentially expressed genes of tumor boundary

To reconstruct of spatial tumor microenvironment, we firstly find differentially expressed genes (DEG) in boundary:

```
# DiffGenes <- FindDiffGenes(TumorST = TumorST, assay = "Spatial")
```

Then, we visualized tumor boundary DEG use volcano plot. If you want to visualize other types of ST data spots, you can change parameter Location to “Tumor” or “OutSpots”.

```
# DiffVolcanoplot(
#   DiffGenes = DiffGenes,
#   Location = "Trans",
#   cut_off_pvalue = 2,
#   cut_off_logFC = 0.25,
#   n = 15
# )
```

After find the DEG of tumor boundary, we explored the KEGG and GO pathways theses features enriched in and speculated the possibly function of tumor boundary

```
# boundary_enrich <-
#   FeatureEnrichment(
#     DiffGenes = DiffGenes,
#     cut_off_logFC = 0.25,
#     cut_off_pvalue = 0.05,
#     Location = "Trans"
#   )
```

4. Summary

Cottrazm contain three main function modules (Boundary Define, Spot Deconvolution, and Reconstruct Spatial TME) which allow users to easily analys ST tumor data. here may still be gaps in the toolkit, but we’ll work on more features later. If you have any suggestions or questions, please feel free to contact us (vera-morland@sjtu.edu.cn).

5. Session information

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/libopenblas-r0.3.3.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.1.0    pillar_1.7.0      prettyunits_1.1.1 remotes_2.4.2
##  [5] tools_4.1.0       testthat_3.1.3    digest_0.6.29     pkgbuild_1.3.1
##  [9] pkgload_1.2.4     memoise_2.0.1     evaluate_0.15     lifecycle_1.0.1
## [13] tibble_3.1.6      pkgconfig_2.0.3   rlang_1.0.2       cli_3.2.0
## [17] rstudioapi_0.13   yaml_2.3.5        xfun_0.30         fastmap_1.1.0
## [21] stringr_1.4.0     withr_2.5.0       knitr_1.38        desc_1.4.1
## [25] fs_1.5.2          vctrs_0.4.1       devtools_2.4.3    rprojroot_2.0.3
## [29] glue_1.6.2        R6_2.5.1          processx_3.5.3    fansi_1.0.3
## [33] rmarkdown_2.13    sessioninfo_1.2.2 callr_3.7.0       purrr_0.3.4
## [37] magrittr_2.0.3    ps_1.6.0          ellipsis_0.3.2    htmltools_0.5.2
## [41] usethis_2.1.5     utf8_1.2.2        stringi_1.7.6     cachem_1.0.6
## [45] crayon_1.5.1      brio_1.1.3
```

6. Reference

1. Pham, D. *et al.* stLearn: integrating spatial location, tissue morphology and gene expression to find cell types, cell-cell interactions and spatial trajectories within undissociated tissues. *undefined* (2020) doi:10.1101/2020.05.31.125658.

2. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science (New York, N.Y.)* **352**, 189–196 (2016).

3. Dong, R. & Yuan, G. C. SpatialDWLS: accurate deconvolution of spatial transcriptomic data. *Genome Biology* **22**, 1–10 (2021).