Single-Cell Signature Explorer Manual

Frédéric PONT

October 26, 2023

Contents

0.1 Softwares presentation

Single-Cell Signature Explorer

Single-Cell Signature Explorer is a package of four softwares dedicated to high throughput signature exploration in single-cell RNAseq analysis. Prior to their analysis by Single-Cell Signature Explorer, the data sets may optionally be transformed for noise reduction. We recommend sctransform as a noise reduction tool [1] available at `https://github.com/ChristophH/sctransform`

1. Single-Cell Signature Scorer computes for each cell a signature score.
2. Single-Cell Signature Merger collates the signature scores table with t-SNE coordinates.
3. Single-Cell Signature Viewer displays signatures scores on a t-SNE map.
4. Single-Cell Signature Combiner displays the combination of two signatures scores on a t-SNE map.

The softwares have been developed with usability and performances in mind. They require no complex command lines or computing skills, they can be used on a laptop but scale very well on powerful workstations for fast computations: depending on the size of the gene set, results are obtains in seconds or minutes. Files can be accessed at https://sites.google.com/site/fredsoftwares/products/single-cell-signature-explorer

0.2 Software installation and use

Single-Cell Signature Scorer and Single-Cell Signature Merger have been compiled for GNU Linux, MAC OS, and Microsoft©Windows™64 bits and require no complex installation procedure. A standalone executable file requiring no dependencies is provided.

Single-Cell Signature Viewer can be used in all operating systems supported by R language (https://www.r-project.org/), ie Windows, Linux and MacOS.

DataSet Noise Reduction

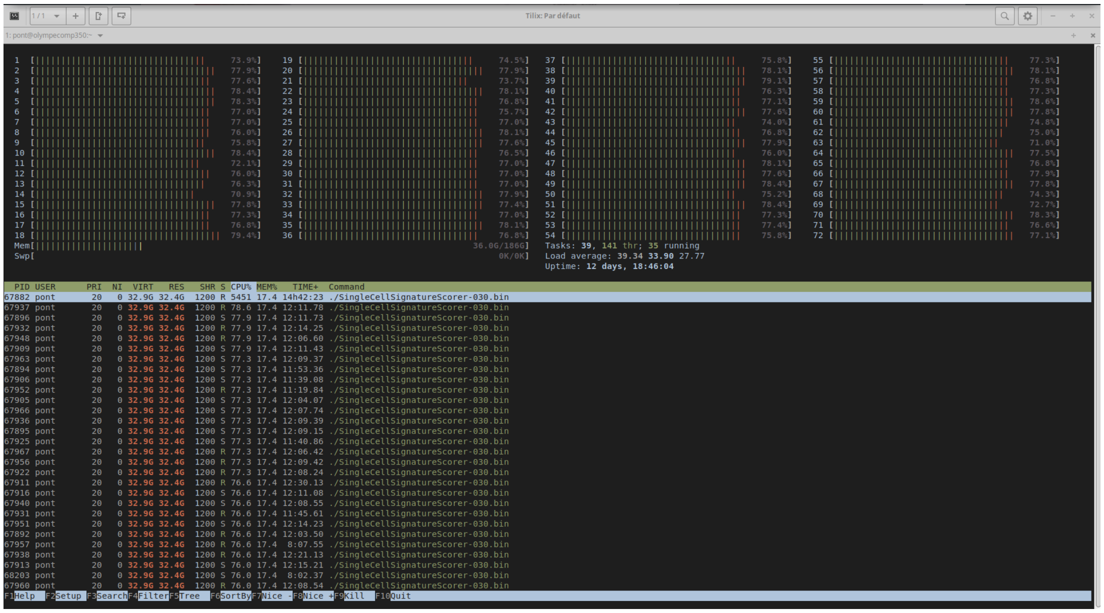
Single Cell transcriptomic technology is powerful to explore gene expression at the single cell level, but scRNAseq datasets may suffer of technical variability, referred below to as ”noise”. This noise includes both gene sampling noise and cell-to-cell variations in sequencing efficiency [2].
Single-Cell Signature Scorer does not perform noise reduction. The user have the choice to use any noise reduction method present or future during the data normalisation step. For data with high noise levels, we recommend sctransform as a noise reduction tool [1] available at `https://github.com/ChristophH/sctransform`

Single-Cell Signature Explorer indifferently processes scRNAseq datasets composed of either raw UMI data, noise-reduced transformed UMI data, or UMI data normalized by Seurat or other pipeline. This choice is entirely under the responsibility of the user, and preliminary to use of Single-Cell Signature Explorer.

Single-Cell Signature Scorer

This software is massively parallel and will take all available CPU cores of the computer (fig. 1). A speed increase $> \times 17$ was measured between a single physical core and 20 cores computation of the same dataset. Each thread will process the score of one cell for all the pathways in a given database. Hyper-threading gives no or little performance improvements.
Single-Cell Signature Scorer has been rewritten in August 2020 to reduce the memory footprint of the software (low memory scorer). Instead of storing the results in RAM, it now uses temporary files. In october 2023, temporary files have been replaced by a high speed database in the low memory scorer. At the expense of a little slowdown (around 20%), huge data-sets can be processed with a limited amount of RAM (typically 16 Go). Another difference between the scorer and the low memory scorer is that the scorer remove the pathways with $scores = 0$ everywhere, whereas the low memory scorer save those columns

full of zeros[1].
Single-Cell Signature Scorer can be configured to run on workstation in interactive mode and on server or batch mode with progress bars disabled. Prior to launching Single-Cell Signature Scorer, user should prepare its single-cell expression data. Typically, these are produced by Seurat R package using



```
  1  [|||||||||||||||||||||||||||||||     73.9%]   19 [|||||||||||||||||||||||||||||||     74.5%]   37 [|||||||||||||||||||||||||||||||     75.8%]   55 [|||||||||||||||||||||||||||||||     77.3%]
  2  [|||||||||||||||||||||||||||||||||   77.9%]   20 [|||||||||||||||||||||||||||||||||   77.9%]   38 [|||||||||||||||||||||||||||||||||   78.1%]   56 [|||||||||||||||||||||||||||||||||   78.1%]
  3  [||||||||||||||||||||||||||||||||    77.6%]   21 [||||||||||||||||||||||||||||||      73.7%]   39 [|||||||||||||||||||||||||||||||||   79.1%]   57 [||||||||||||||||||||||||||||||||    76.8%]
  4  [|||||||||||||||||||||||||||||||||   78.4%]   22 [|||||||||||||||||||||||||||||||||   78.1%]   40 [||||||||||||||||||||||||||||||||    76.3%]   58 [||||||||||||||||||||||||||||||||    77.3%]
  5  [|||||||||||||||||||||||||||||||||   78.3%]   23 [||||||||||||||||||||||||||||||||    76.8%]   41 [||||||||||||||||||||||||||||||||    77.1%]   59 [|||||||||||||||||||||||||||||||||   78.6%]
  6  [||||||||||||||||||||||||||||||||    77.0%]   24 [||||||||||||||||||||||||||||||||    75.7%]   42 [|||||||||||||||||||||||||||||||||   77.6%]   60 [|||||||||||||||||||||||||||||||||   77.8%]
  7  [||||||||||||||||||||||||||||||||    77.0%]   25 [||||||||||||||||||||||||||||||||    77.0%]   43 [|||||||||||||||||||||||||||||||     74.0%]   61 [||||||||||||||||||||||||||||||||    74.8%]
  8  [||||||||||||||||||||||||||||||||    76.0%]   26 [|||||||||||||||||||||||||||||||||   78.1%]   44 [||||||||||||||||||||||||||||||||    76.8%]   62 [|||||||||||||||||||||||||||||||     75.0%]
  9  [|||||||||||||||||||||||||||||||     75.8%]   27 [|||||||||||||||||||||||||||||||||   77.6%]   45 [|||||||||||||||||||||||||||||||||   77.9%]   63 [||||||||||||||||||||||||||||||      71.0%]
 10  [|||||||||||||||||||||||||||||||||   78.4%]   28 [||||||||||||||||||||||||||||||||    76.5%]   46 [|||||||||||||||||||||||||||||||     76.0%]   64 [|||||||||||||||||||||||||||||||||   77.5%]
 11  [||||||||||||||||||||||||||||||      72.1%]   29 [||||||||||||||||||||||||||||||||    77.0%]   47 [|||||||||||||||||||||||||||||||||   78.1%]   65 [||||||||||||||||||||||||||||||||    76.8%]
 12  [||||||||||||||||||||||||||||||||    76.0%]   30 [||||||||||||||||||||||||||||||||    77.0%]   48 [|||||||||||||||||||||||||||||||||   77.6%]   66 [||||||||||||||||||||||||||||||||    77.9%]
 13  [|||||||||||||||||||||||||||||||     76.3%]   31 [||||||||||||||||||||||||||||||||    77.0%]   49 [|||||||||||||||||||||||||||||||||   78.4%]   67 [|||||||||||||||||||||||||||||||||   77.8%]
 14  [|||||||||||||||||||||||||||||       70.9%]   32 [|||||||||||||||||||||||||||||||||   77.9%]   50 [|||||||||||||||||||||||||||||||     75.2%]   68 [|||||||||||||||||||||||||||||||     74.3%]
 15  [|||||||||||||||||||||||||||||||||   77.8%]   33 [|||||||||||||||||||||||||||||||||   77.4%]   51 [|||||||||||||||||||||||||||||||||   78.4%]   69 [||||||||||||||||||||||||||||||      72.7%]
 16  [||||||||||||||||||||||||||||||||    77.3%]   34 [||||||||||||||||||||||||||||||||    77.0%]   52 [||||||||||||||||||||||||||||||||    77.3%]   70 [|||||||||||||||||||||||||||||||||   78.3%]
 17  [||||||||||||||||||||||||||||||||    76.8%]   35 [|||||||||||||||||||||||||||||||||   78.1%]   53 [||||||||||||||||||||||||||||||||    77.4%]   71 [||||||||||||||||||||||||||||||||    76.6%]
 18  [||||||||||||||||||||||||||||||||||  79.4%]   36 [||||||||||||||||||||||||||||||||    76.8%]   54 [||||||||||||||||||||||||||||||||    75.8%]   72 [|||||||||||||||||||||||||||||||||   77.1%]
 Mem[||||||||||||||||||||||                                                     36.0G/186G]   Tasks: 39, 141 thr; 35 running
 Swp[                                                                                0K/0K]   Load average: 39.34 33.90 27.77
                                                                                              Uptime: 12 days, 18:46:04

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
67882 pont       20   0 32.9G 32.4G  1200 R 5451 17.4 14h42:23 ./SingleCellSignatureScorer-030.bin
67937 pont       20   0 32.9G 32.4G  1200 R 78.6 17.4 12:11.78 ./SingleCellSignatureScorer-030.bin
67896 pont       20   0 32.9G 32.4G  1200 S 77.9 17.4 12:11.73 ./SingleCellSignatureScorer-030.bin
67932 pont       20   0 32.9G 32.4G  1200 R 77.9 17.4 12:14.25 ./SingleCellSignatureScorer-030.bin
67948 pont       20   0 32.9G 32.4G  1200 R 77.9 17.4 12:06.60 ./SingleCellSignatureScorer-030.bin
67909 pont       20   0 32.9G 32.4G  1200 S 77.9 17.4 12:11.43 ./SingleCellSignatureScorer-030.bin
67963 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 12:09.37 ./SingleCellSignatureScorer-030.bin
67894 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 11:53.36 ./SingleCellSignatureScorer-030.bin
67906 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 11:39.08 ./SingleCellSignatureScorer-030.bin
67952 pont       20   0 32.9G 32.4G  1200 R 77.3 17.4 11:19.84 ./SingleCellSignatureScorer-030.bin
67905 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 12:04.07 ./SingleCellSignatureScorer-030.bin
67966 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 12:07.74 ./SingleCellSignatureScorer-030.bin
67936 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 12:09.39 ./SingleCellSignatureScorer-030.bin
67895 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 12:09.15 ./SingleCellSignatureScorer-030.bin
67925 pont       20   0 32.9G 32.4G  1200 S 77.3 17.4 11:40.86 ./SingleCellSignatureScorer-030.bin
67967 pont       20   0 32.9G 32.4G  1200 R 77.3 17.4 12:06.42 ./SingleCellSignatureScorer-030.bin
67956 pont       20   0 32.9G 32.4G  1200 R 77.3 17.4 12:09.42 ./SingleCellSignatureScorer-030.bin
67922 pont       20   0 32.9G 32.4G  1200 R 77.3 17.4 12:08.24 ./SingleCellSignatureScorer-030.bin
67911 pont       20   0 32.9G 32.4G  1200 R 76.6 17.4 12:30.13 ./SingleCellSignatureScorer-030.bin
67916 pont       20   0 32.9G 32.4G  1200 S 76.6 17.4 12:11.08 ./SingleCellSignatureScorer-030.bin
67940 pont       20   0 32.9G 32.4G  1200 S 76.6 17.4 12:08.55 ./SingleCellSignatureScorer-030.bin
67931 pont       20   0 32.9G 32.4G  1200 R 76.6 17.4 11:45.61 ./SingleCellSignatureScorer-030.bin
67951 pont       20   0 32.9G 32.4G  1200 S 76.6 17.4 12:14.23 ./SingleCellSignatureScorer-030.bin
67892 pont       20   0 32.9G 32.4G  1200 R 76.6 17.4 12:03.50 ./SingleCellSignatureScorer-030.bin
67957 pont       20   0 32.9G 32.4G  1200 R 76.6 17.4  8:07.55 ./SingleCellSignatureScorer-030.bin
67938 pont       20   0 32.9G 32.4G  1200 R 76.6 17.4 12:21.13 ./SingleCellSignatureScorer-030.bin
67913 pont       20   0 32.9G 32.4G  1200 S 76.0 17.4 12:15.21 ./SingleCellSignatureScorer-030.bin
68203 pont       20   0 32.9G 32.4G  1200 S 76.0 17.4  8:02.37 ./SingleCellSignatureScorer-030.bin
67960 pont       20   0 32.9G 32.4G  1200 R 76.0 17.4 12:08.54 ./SingleCellSignatureScorer-030.bin
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Figure 1: Screenshot of Single-Cell Signature Scorer high throughput parallel computations on one supercomputer Olympe node (`https://www.calmip.univ-toulouse.fr/`).

```
"write.table(t(as.matrix(MySeuratObject@data)), "FileName.tsv",
sep="\t", row.names = TRUE, col.names=NA)"
```

1. unzip the software
2. if needed, edit the conf.json file with a text editor.
   3 parameters for the scorer (”removeLog2”, ”server”, ”DBserver”) or 4 parameters for the low memory scorer (”removeLog2”, ”server”, ”DBserver”, ”CPU”) can be changed in the configuration file :

   (a) ”removeLog2”: 1 (1 : remove log2() transformation. 0 : do not remove log2() transformation. Depending on the data processing pipeline, the gene expression matrix can be log2 transformed, in that case, it is better to remove log2 transformation before calculating the

[1] To save RAM, the result table is constructed line by line, it is thus not possible to remove columns full of zeros.

scores. By default log2 removal is enabled.
Caution ! do not enable removeLog2 when the $UMI > 1000$, this will produce a ∞/∞ ratio (see troubleshooting).

(b) "server": 0 (1 : enable server configuration. No interaction are required by the user and progress bar are disabled to limit I/O when the scorer is used in batch mode on a cluster. Server mode is disabled by default)

(c) "DBserver": "2-4,17" (when server mode is enabled, the data base input is read from DBserver. For example "DBserver": "2-4,17" means that the database 2,3,4,17 will be used. Please note that since version 032, is also possible to enter the database list in command line using the -db flag. When the -db flag is used, the database list in the json file will be ignored.

(d) "CPU": 0 (0 : use all available CPUs with hyperthreading. $n > 1$ use n CPUs)

3. copy Single-Cell expression data tsv file to the data directory : the first row are gene names, the first column are cell barcodes and the next columns are the expression values, with separator TAB. Since cell names are used as file names for temporaty files, they should not contain special characters such as "/ \". This is an example of the table :

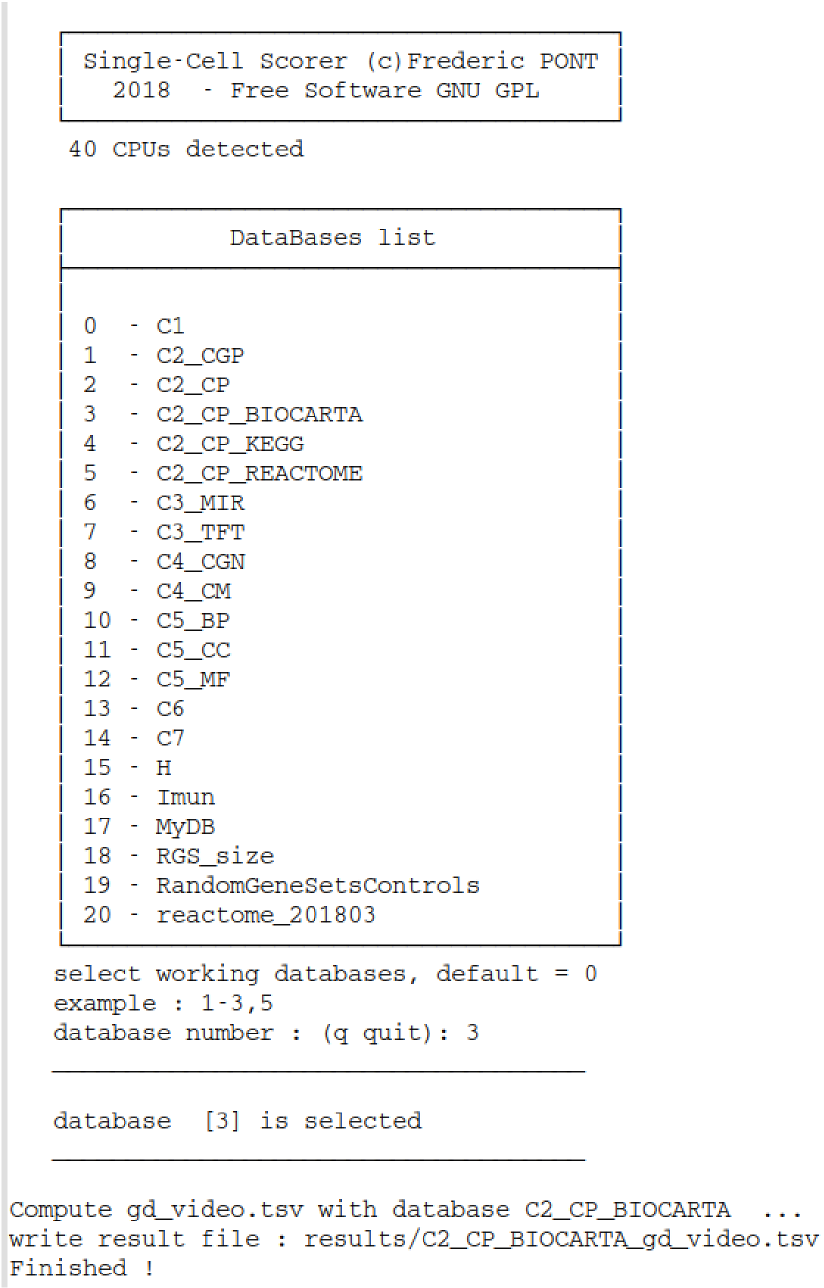
id	NOC2L	RP11-34P13.7
cel_AA...CTCA	2.5	0
cel_AA...TGGT	3.64	2.71
cel_AA...TTCT	0	0.195

4. copy your preferred databases of gene sets to the databases directory. Databases are text files, each file contains a list of genes and the title of the file is the name of the gene set. The files should be placed in a directory and the name of the directory is the name of the database. No sub-folders allowed.

5. execute the software by the command $< ./software - name >$ or double click on the .exe (Windows) or .bin (Linux) file

6. Enter the number of the signature collections to compute. Example : 1-3,5 will select collection 1,2,3,5.

7. Scores are in the results directory.



```
 ┌──────────────────────────────────────┐
 │ Single-Cell Scorer (c)Frederic PONT  │
 │   2018  - Free Software GNU GPL      │
 └──────────────────────────────────────┘
  40 CPUs detected

 ┌──────────────────────────────────────┐
 │            DataBases list            │
 ├──────────────────────────────────────┤
 │                                      │
 │ 0  - C1                              │
 │ 1  - C2_CGP                          │
 │ 2  - C2_CP                           │
 │ 3  - C2_CP_BIOCARTA                  │
 │ 4  - C2_CP_KEGG                      │
 │ 5  - C2_CP_REACTOME                  │
 │ 6  - C3_MIR                          │
 │ 7  - C3_TFT                          │
 │ 8  - C4_CGN                          │
 │ 9  - C4_CM                           │
 │ 10 - C5_BP                           │
 │ 11 - C5_CC                           │
 │ 12 - C5_MF                           │
 │ 13 - C6                              │
 │ 14 - C7                              │
 │ 15 - H                               │
 │ 16 - Imun                            │
 │ 17 - MyDB                            │
 │ 18 - RGS_size                        │
 │ 19 - RandomGeneSetsControls          │
 │ 20 - reactome_201803                 │
 └──────────────────────────────────────┘
   select working databases, default = 0
   example : 1-3,5
   database number : (q quit): 3
   ____________________________________

   database  [3] is selected
   ____________________________________

Compute gd_video.tsv with database C2_CP_BIOCARTA  ...
write result file : results/C2_CP_BIOCARTA_gd_video.tsv
Finished !
```

Figure 2: Screenshot of Single-Cell Signature Scorer

Database installation on Single-Cell Signature Scorer

In this section we describe

1. Structure of a database
2. How to install ready to use databases
3. How to build a custom databases

Structure of a database. In "SingleCellSignatureScorer", the folder containing databases is called "databases". The folder "databases" must contains at least one sub-folder containing genesets files. For example, the valid structure for the "KEGG" database is :
database/KEGG/[KEGG_ABC_TRANSPORTERS.txt KEGG_ACUTE_MYELOID_LEUKEMIA.txt KEGG_ADHERENS_JUNCTION.txt ...]
No more levels of subfolder are allowed. A geneset is a simple text file with each gene folowed by a newline. For example, the "KEGG_FOLATE_BIOSYNTHESIS.txt" file contains the folowing genes :

```
QDPR
ALPP
ALPPL2
DHFR
GGH
GCH1
FPGS
SPR
ALPL
PTS
ALPI
```

How to distinguish genes inhibitor or activator in a custom database ? It is possible to add a "-" sign in front of inhibitory genes, the scorer will subtract (instead of add) the signal of this gene. for example :

```
GENEACT1
-GENEINHIB
GENEACT2
...
```

Caution : if the inhibitor gene is more expressed that the sum of activator genes, the resulting score will be negative and difficult to interpret.
An alternative could be to score inhibitor and activator genes in two separated gene sets and then use the Single-Cell Signature Combiner to analyse their combination. One solution is to insert the activator and inhibitor scores as 2 supplementary columns in the data table and then view the results in the combiner (the viewer/combiner works on one data file at a time only). For example :

XY : choose UMAP1 & 2 or t-SNE1 & 2
signature 1 : activator genes for example
signature 2 : inhibitor genes for example
operator : combine with sign "+" = activator + inhibitor
operator : combine with sign "-" = activator - inhibitor

How to install ready to use databases. In the software webpage [2] a link [3] to our ready to use database is provided. To install a database, just unzip the database to the "database" directory and then remove the zip file. After a restart of Single-Cell Signature Scorer, the new databases will appear.

How to build a custom databases. In "Single Cell Signature Scorer", the folder containing databases is called "databases". To build a custom databases, in the folder "databases" create a new folder with the custom database name, for example "MyDB". In this folder add one text file per geneset. The structure of a geneset is explain in §0.2 Structure of a database. page 6

How to use Single-Cell Signature Scorer ($Version \geq 32$) on a cluster.

1. Edit the conf.json file with a text editor and enable the server mode : "server": 1,

2. you can use the "-db" flag in your sbatch file to start Single-Cell Signature Scorer on a cluster on a cluster node. The "-db" flag will overwrite the database setting of the JSON file, so it is not necessary to modify the database setting in this file. For example to run scScorer with databases 2,3,4,7 on a cluster node using 72 threads with a Slurm Workload Manager (Fig 1):

```
#!/bin/bash
#SBATCH -J scScorer
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --threads-per-core=1
#SBATCH --cpus-per-task=36
#SBATCH --time=01:00:00
workdir=/clusterDir/SingleCellSignatureScorer
cd ${workdir}
./SingleCellSignatureScorer_Linux_032.bin -db 2-4,7
```

[2] https://sites.google.com/site/fredsoftwares/products/single-cell-signature-explorer
[3] https://sites.google.com/site/fredsoftwares/products/databases

Single-Cell Signature Merger

This software merges one t-SNE coordinate table with n signature scores tables in parallel.

Typically, t-SNE coordinates could be exported from Seurat object using:

```
c<-cbind(MySeuratObject@cell.names, MySeuratObject@dr$tsne@cell.embeddings)
write.table(c,"My_tsne_coordinate.tsv", sep="\t", row.names = FALSE)
```

1. unzip the software

2. copy only one t-SNE coordinates table to the ”tsne” directory. : the first column are cell barcodes and the next columns are t-SNE (and/or UMAP, or other XY) coordinates, with separator TAB. Other columns are optional. This is an example of the table :

id	tSNE_1	tSNE_2
cel_AA...CTCA	16.16	3.97
cel_AA...TGGT	-14.51	12.97
cel_AA...TTCT	-3.33	4.38

3. copy signatures tsv files to the scores directory : the first column are cell barcodes and the next columns are the signature's scores, with separator TAB. This is an example of the table :

id	KEGG_PW1	KEGG_PW2
cel_AA...CTCA	0.072	0.398
cel_AA...TGGT	0.064	0.71
cel_AA...TTCT	0.049	0.195

4. optionally, it is possible to copy lists of column names (gene names, or lists of signatures) to the ”lists” directory. In that case the finale table will only contains the genes present in the lists. It is possible to include an unlimited number of lists, for each list a table will be produced. The list must be a text file with one item (gene name) per line. The first column is included by default.

5. execute the software by the command $< ./software-name >$ or double click on the .exe (Windows) or .bin (Linux) file. if lists are present in the ”lists” directory, one table per list will be produced. If no lists are present, the user will be asked if sorting the table is necessary. Sorting the table is use full to display gene expression in the viewer/combiner because it is easier to find a gene when they are ordered. Sorting the table is useless when signature scores have been calculated by Single-Cell Signature Scorer because they are already sorted. Merging without sorting is faster.

6. Merged tables are in the results directory. This is an example of the final table :

id	KEGG_PW1	KEGG_PW2	tSNE_1	tSNE_2
cel_AA...CTCA	0.072	0.398	16.16	3.97
cel_AA...TGGT	0.064	0.71	-14.51	12.97
cel_AA...TTCT	0.049	0.195	-3.33	4.38

Install R environment with conda/mamba for Single-Cell Signature Viewer/Combiner

To install easily the R packages needed for Single-Cell Signature Viewer/combiner, it is possible to use the conda/mamba as follow :

1. install conda or mamba `https://github.com/conda-forge/miniforge/releases/latest/download/Mambaforge-Linux-x86_64.sh`
2. conda deactivate
3. A file "Rbase.yml" is provided in the "conda_mamba_viewer directory". Use this file to create the "Rbase" environment.
   mamba env create −−file Rbase.yml or
   conda env create −−file Rbase.yml
4. If necessary, activate the environment : conda activate Rbase. "(Rbase)" should appear in your terminal at the beginning of the prompt line indicating that the environment is activated.
5. All the R packages needed for the Viewer/Combiner are installed and you can skip the packages installation step in the next §. This procedure has been tested on Linux, I did not try on windows.

Single-Cell Signature Viewer

Single-Cell Signature Viewer is a simple software to visualize gene scores, it can also display any type of numerical data. It is also possible to obtain beautiful images using Single-Cell Spatial Explorer `https://github.com/FredPont/spatial` with the help of Spatial Background Builder `https://github.com/FredPont/Spatial_Background_Builder`

1. install R programming language (https://www.r-project.org/)
2. Since version 2021-09-29 the R packages should be installed automatically at the first use of the viewer and the combiner. After the automatic install it can be necessary to restart the software and do the step 6 twice. If, for any reason the automatic install failed, you can install R packages manually : shiny, shinyWidgets, Cairo, ggplot2, svglite using the command :
   - install.packages("Cairo") (additional libraries [4] of your Linux distribution can be required to install this package. When possible, install Cairo directly from the package manager : r-cran-cairo in Ubuntu)
   - install.packages("ggplot2")
   - install.packages("shiny")

[4] https://stackoverflow.com/questions/9437246/unable-to-install-cairo-package-under-linux

- install.packages(”shinyWidgets”)
- install.packages(”svglite”)

For windows users the simplest install procedure is (see video) :

- Install R `https://cran.r-project.org/bin/windows/base/`
- it is generally necessary to install RTools : `https://cran.r-project.org/bin/windows/Rtools/`
- Install Rstudio `https://www.rstudio.com/products/rstudio/download/`
- In Rstudio, click on the Packages tab in the bottom-right section and then click on install.
- write the package name you want to install (Cairo, ggplot2, shiny) under the Packages field and then click install. Check the box ”dependencies”.

3. unzip the software

4. copy your score table merged with t-SNE coordinates to the ”data” directory. The table must have a header with the names of the columns. The first column of the table must be the cell barcodes. The next columns of the table are the signature scores. The table must contains two columns with the map (tSNE, UMAP...) coordinates. This is an example of the table :

id	KEGG_PW1	KEGG_PW2	tSNE_1	tSNE_2
cel_AA...CTCA	0.072	0.398	16.16	3.97
cel_AA...TGGT	0.064	0.71	-14.51	12.97
cel_AA...TTCT	0.049	0.195	-3.33	4.38

5. open a terminal and open the software directory using cd command

6. in the software directory, execute the software by the command : Rscript Run.R or start R and type < *source*(”*Run.R*”) >

7. a web address such as Listening on http://127.0.0.1:7918 will appear on the terminal

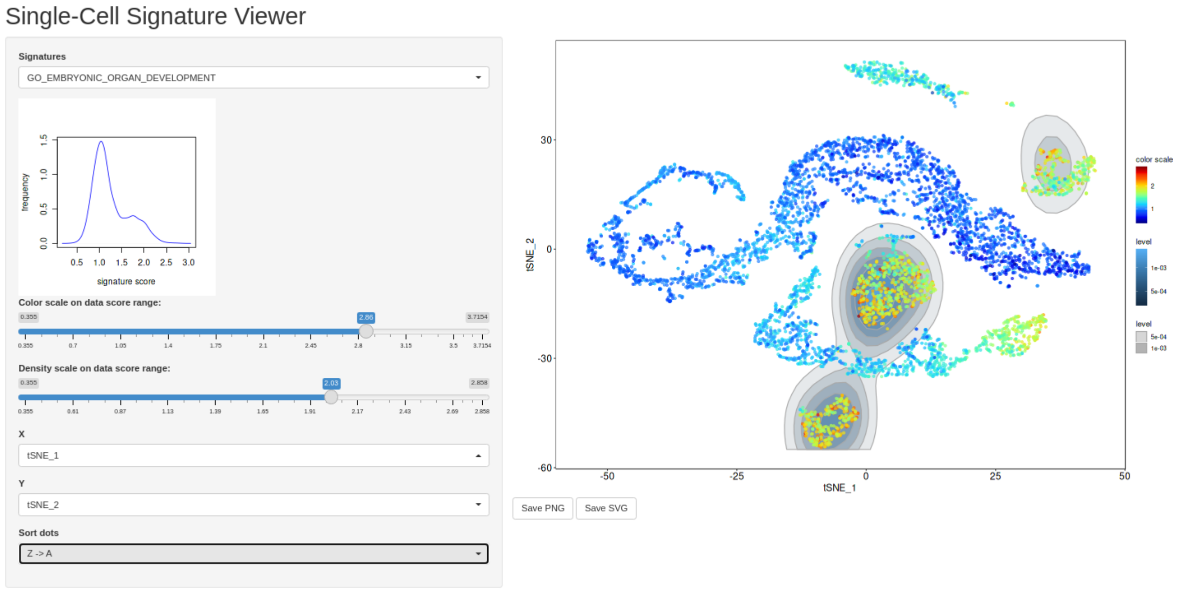
8. copy this address on your web browser

9. a web page will appear with an interactive plot.

   - Select the signature to plot on the t-SNE map using the signature list.
   - Select the XY coordinates in the X,Y lists.
   - To bring some dots in background/foreground use the dropdown list ”Sort dots”.

- The viewer draws a density distribution of scores and provides a color scale cursor to prune potential outliers.
- The buttons ”save PNG/SVG” saves the plot to the ”plot” directory.
- The button ”Map Dot size” change the size of the dots in the displayed image and also in exported images (png, svg).

10. Navigation tips :
    A keyboard navigation is possible to select the signatures. An arrow key open the drop-down list, the TAB key allows the signature selection and then the enter key load the selected signature.
    A quick access to one signature is also possible by typing the first letters of the signature in the drop-down list entry.



Single-Cell Signature Viewer

Signatures: GO_EMBRYONIC_ORGAN_DEVELOPMENT

Color scale on data score range: 0.355 – 2.86 (max 3.7154)

Density scale on data score range: 0.355 – 2.03 (max 2.858)

X: tSNE_1

Y: tSNE_2

Sort dots: Z -> A

Figure 3: Single-Cell Signature Viewer. Signatures can be selected with the drop-down menu on the left. Any [X,Y] coordinates can be selected with the drop-down menu on the left. The ”Sort dots” drop-down menu, sorts the dots by values, so higher values can be in foreground/background. The color scale of the map can be adjusted with the blue slider, such as to avoid the color-collapsing effect of potential outliers. The frequency of the corresponding signature scores across the whole data set is plotted against the signature score.

Single-Cell Signature Combiner

The requirements and the procedure is the same than the Single-Cell Signature Viewer.

1. Install R programming language (https://www.r-project.org/)
2. Since version 2021-09-29 the R packages should be installed automatically at the first use of the viewer and the combiner. After the automatic install it can be necessary to restart the software and do the step 6 twice. If, for any reason the automatic install failed, you can install R packages manually : shiny, shinyWidgets, Cairo, ggplot2, svglite using the command :
    - install.packages("Cairo") (additional libraries [5] of your Linux distribution can be required to install this package)
    - install.packages("ggplot2")
    - install.packages("shiny")
    - install.packages("shinyWidgets")
    - install.packages("svglite")
3. unzip the software
4. copy your score table merged with t-SNE coordinates to the "data" directory. The table must have a header with the names of the columns. The first column of the table must be the cell barcodes. The next columns of the table are the signature scores. The table must contains two columns with the map (tSNE, UMAP...) coordinates. This is an example of the table :

id	KEGG_PW1	KEGG_PW2	tSNE_1	tSNE_2
cel_AA...CTCA	0.072	0.398	16.16	3.97
cel_AA...TGGT	0.064	0.71	-14.51	12.97
cel_AA...TTCT	0.049	0.195	-3.33	4.38

5. open a terminal and open the software directory using cd command
6. in the software directory, execute the software by the command : Rscript Run.R or start R and type < *source("Run.R")* >
7. a web address such as

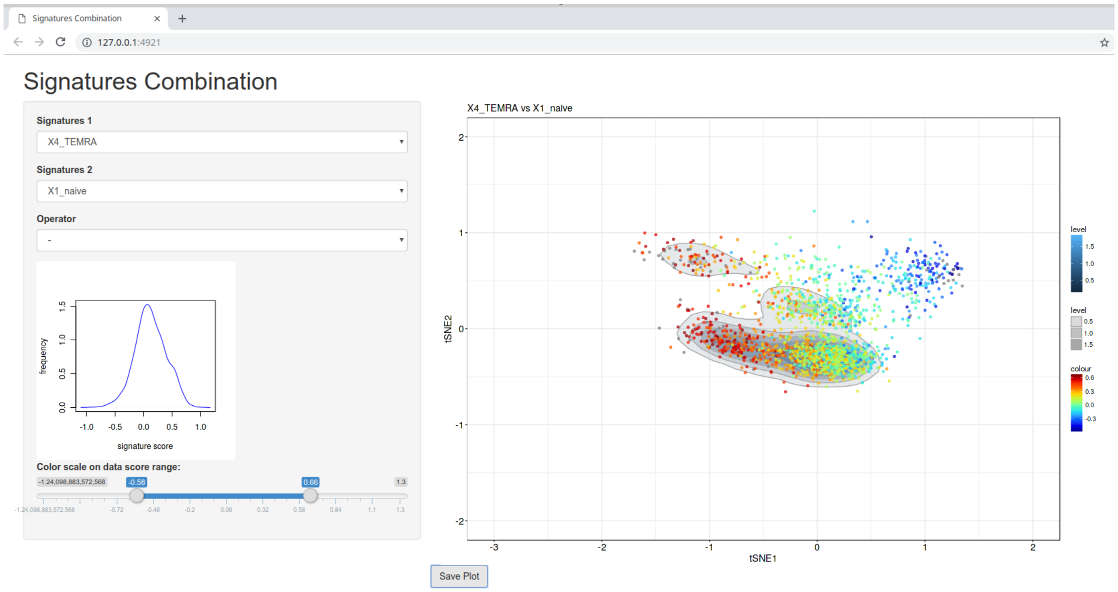
```
Listening on http://127.0.0.1:7918
```

[5] https://stackoverflow.com/questions/9437246/unable-to-install-cairo-package-under-linux

will appear on the R terminal

8. copy this address on your web browser

9. a web page will appear with an interactive plot. Select the signatures combination to plot on the t-SNE map using the two signature lists and the XY coordinates in the X,Y lists. Select the operator using the operators list. It is possible to compute the difference (-), the sum (+) or even to multiply (*) two signatures scores. Before combining the two score sets, they are normalized between[0-1] to be comparable. It is the author's experience that in some data sets, rare cells with very strong scores, referred below to as ”outliers”, artefactually collapse the color range of the whole t-SNE. This issue can be addressed as follows. The viewer draws a density distribution of scores and provides a color scale cursor allowing the user to prune potential outliers. The button ”save plot” saves the plot to the ”plot” directory.



127.0.0.1:4921

Signatures Combination

Signatures 1: X4_TEMRA

Signatures 2: X1_naive

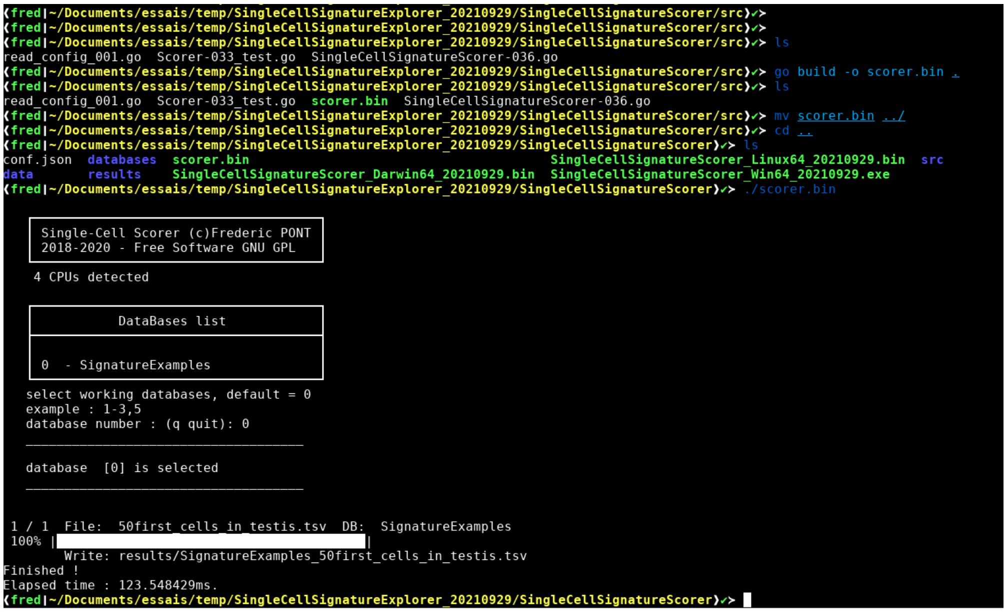
Operator: -

Color scale on data score range: -1.24,098,883,572,568 – -0.58 – 0.66 – 1.3

Figure 4: Screenshot of Single-Cell Signature Combiner. Signatures and operator can be selected with the drop-down menus on the left. The color scale of the map can be adjusted with the slider, according to the frequency histogram. The frequency of the combination scores is plotted against the signature score.

0.3 How to compile Go software for another platform ?

Single-Cell Signature Scorer/Merger are written in Go and standalone executable binaries are provided for Linux and windows 64 bits. Go is supported by many other OS. To compile a Go software use the following procedure. More details can be found in the Go website if necessary.



```
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓>
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓>
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓> ls
read_config_001.go  Scorer-033_test.go  SingleCellSignatureScorer-036.go
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓> go build -o scorer.bin .
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓> ls
read_config_001.go  Scorer-033_test.go  scorer.bin  SingleCellSignatureScorer-036.go
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓> mv scorer.bin ../
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer/src❭✓> cd ..
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer❭✓> ls
conf.json  databases  scorer.bin                                    SingleCellSignatureScorer_Linux64_20210929.bin  src
data       results    SingleCellSignatureScorer_Darwin64_20210929.bin  SingleCellSignatureScorer_Win64_20210929.exe
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer❭✓> ./scorer.bin


   ┌──────────────────────────────────────┐
   │ Single-Cell Scorer (c)Frederic PONT  │
   │ 2018-2020 - Free Software GNU GPL    │
   └──────────────────────────────────────┘
    4 CPUs detected

   ┌──────────────────────────────────────┐
   │           DataBases list             │
   ├──────────────────────────────────────┤
   │                                      │
   │ 0  - SignatureExamples               │
   └──────────────────────────────────────┘
   select working databases, default = 0
   example : 1-3,5
   database number : (q quit): 0
   ____________________________________

   database  [0] is selected
   ____________________________________


 1 / 1  File:  50first_cells_in_testis.tsv  DB:  SignatureExamples
 100% |████████████████████████████████████████|
       Write: results/SignatureExamples_50first_cells_in_testis.tsv
Finished !
Elapsed time : 123.548429ms.
❬fred|~/Documents/essais/temp/SingleCellSignatureExplorer_20210929/SingleCellSignatureScorer❭✓> █
```

Figure 5: Compilation of Single-Cell Signature Scorer.

1. download Go `https://golang.org/`
2. install Go `https://golang.org/doc/install`
3. install missing packages. The packages used by Single-Cell Signature Scorer/Merger are :
   ”bufio”
   ”bytes”
   ”encoding/csv”
   ”flag”
   ”fmt”
   ”github.com/schollz/progressbar”
   ”io”
   ”io/ioutil”
   ”log”

"math"
"os"
"path/filepath"
"runtime"
"sort"
"strconv"
"strings"
"time"
Most of them are already included in the Go distribution. To install a missing package use the command :
go get
for example to install the progress bar package :
go get "github.com/schollz/progressbar"

4. go into the src/ directory and compile the software with the command :
go build -o executable_name .
see figure 5

0.4 FAQ

1. Viewer/Combiner : meaning of the "level" legend ?

   The density is computed using the function :

   ```
   stat_density_2d(data=samples[samples[,input$z]>input$densitySelect,],

   aes(fill=..level..,alpha=..level..), geom='polygon', colour='darkgrey', n=50)
   ```

   see : `https://ggplot2.tidyverse.org/reference/geom_density_2d.html`

   - fill=..level.. maps the fill color of the polygons to the level of the density estimate. The level is a value between 0 and 1 that indicates the proportion of points inside each contour. For example, a level of 0.95 means that 95% of the points are inside the contour.
   - alpha=..level.. maps the transparency of the polygons to the level of the density estimate. The alpha value ranges from 0.1 to 1 by default, where 0.1 is the most transparent and 1 is the most opaque. You can change this range with scale_alpha_continuous().

   These aesthetics allow you to create a visual effect where the higher-density regions are more visible and colorful, while the lower-density regions are more faded and transparent. This can help you see the distribution and shape of the data better. It is possible to deactivate the density using the slider. Note that, depending on the dataset, the density is not always possible to compute, leading to an error. In that case it is necessary to disable the density using the slider.

0.5 Troubleshooting

1. In some versions of shiny R package (used by the viewer and the combiner) the following error can occur : Listening on http://127.0.0.1:5405 Error in utils::browseURL(appUrl) : 'browser' must be a non-empty character string.
   Solution : in R session specify the web browser name with the command options(browser="firefox")
   If the previous solution fails, then a non interactive session must be used, in a terminal go to the viewer/combiner directory using "cd" command, and then type :
   Rscript Run.R
   and then open the URL manually in the web browser.

2. Single Cell tables can be very large and slow the viewer/combiner. For small computers, it can be useful to split a large table or extract some interesting genes. This can be done using some tools in this website, for example "Select columns in table".

3. The maximum number of item that a drop down list can display in the Viewer/Combiner is limited. This limit is dependant on the browser, and browser version. For example in Firefox a large list can disappears. In this case, try to use google chrome or split the table as explained above. Note that Single-Cell Spatial Explorer `https://github.com/FredPont/spatial` does not have such limit and is able to display huge scRNAseq data on a small computer since the full data table is not loaded in RAM .

4. If the scorer crashes with a division by zero error : it means that the gene expression for one cell equal zero for all genes. This is normally not possible if the quality control have been done because these cells without signal should be removed. A division by zero error have also been reported by some users who copied some gene signature directly in the "database" folder without using a sub-folder with the database name as explained in the paragraph "Structure of a database".

5. If the scorer gives scores with unexpectedly high values, the "remove log2" option have been used on data that have not been normalized. Note that the "remove log2", for each UMI u, computes 2^u which give infinite values if $u > 1000$. So never use the "remove log2" option for large UMI values.

6. If the scorer gives scores with "NA", see previous remark or search for some strings in numerical columns of the data table.

Bibliography

[1] Hafemeister, C. and Satija, R. (2019) Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *bioRxiv,*.

[2] Grün, D., Kester, L., and Van Oudenaarden, A. (2014) Validation of noise models for single-cell transcriptomics. *Nature methods,* **11**(6), 637.