GPTCelltype: Reference-free and cost-effective automated cell type annotation with GPT-4 in single-cell RNA-seq analysis

Wenpin Hou[1,*], Zhicheng Ji[2,*]
[1]Department of Biostatistics, Columbia University Mailman School of Public Health Health
[2]Department of Biostatistics and Bioinformatics, Duke University School of Medicine
[*] corresponding authors

Introductions

Cell type annotation is an essential step in single-cell RNA-seq analysis. However, it is a time-consuming process that often requires expertise in collecting canonical marker genes and manually annotating cell types. Automated cell type annotation methods typically require the acquisition of high-quality reference datasets and the development of additional pipelines. We demonstrated that GPT-4, a highly potent large language model, can automatically and accurately annotate cell types by utilizing marker gene information generated from standard single-cell RNA-seq analysis pipelines in this manuscript. We developed this software, **GPTCelltype**, to provide reference-free, cost-effective automated cell type annotation using GPT-4 for single-cell RNA-seq analysis.

Dependencies

GPTCelltype depends on the R package openai. Please install and load `openai` before running `GPTCelltype`.

```
install.packages("openai")
library(openai)
```

OpenAI Key

GPTCelltype integrates the OpenAI API into the software. To connect to OpenAI API, a secret API key is required. You can generate your API key in your OpenAI account webpage: log in to OpenAI, click on "Personal" on the upper right corner, click on "View API keys" in the break-down list, and then click on "Create new secret key" which directs you API key page. Copy the key and paste it on a note for further use. Users need to pass their secret API key to GPTCelltype functions as one of the inputs.

Run GPTCelltype

We demonstrate how to run GPTCelltype as follows. The main function is **`gptcelltype()`**. It can annotate cell types by OpenAI GPT models in a Seurat pipeline or with a custom gene list. If **`gptcelltype()`** is used in a Seurat pipeline, Seurat **`FindAllMarkers()`** function needs to be run first and the differential gene table generated by Seurat will serve as the input. If the input is a custom list of genes, one cell type is identified for each element in the list.

Among the input arguments, **`input`** can either be the differential gene table returned by Seurat **`FindAllMarkers()`** function, or a list of genes. **`tissuename`** (optional) is a tissue name. **`openai_key`** is your OpenAI key obtained from API key page (see above section). **`model`** is a valid GPT-4 or GPT-3.5 model name listed on Models page. Default is 'gpt-4'. **`topgenenumber`** is the number of top differential genes to be used when the input is Seurat differential genes. The output is a vector of cell types.

For example, if we provide a list of two gene vectors: the first one contains *CD4* and *CD3D*, and the second one contains *CD14*, then we can call the function in this way:

```
gptcelltype(
  input = list(cluster1 = c('CD4, CD3D'), cluster2 = 'CD14'),
  tissuename = 'human PBMC',
  openai_key = yourkey, ## Note: Please use your OpenAI key.
  model = 'gpt-4'
)
```

Then we can obtain the output

```
"T helper cells"      "Monocytes"
```

Session Info

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/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
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.0.2  fastmap_1.1.1   cli_3.6.1       tools_4.0.2
##  [5] htmltools_0.5.5 rstudioapi_0.14 yaml_2.3.7      rmarkdown_2.21
##  [9] knitr_1.42      xfun_0.39       digest_0.6.31   rlang_1.1.1
## [13] evaluate_0.21
```