fcoex: using coexpression to explore cell type diversity in scRNA-seq data

Authors

- **Tiago Lubiana**
  [0000-0003-2473-2313](#) · [lubianat](#)
  Department of Clinical and Toxicological Analyses, School of Pharmaceutical Sciences, University of São Paulo, São Paulo, Brazil

- **Helder I Nakaya**
  [0000-0001-5297-9108](#)
  Department of Clinical and Toxicological Analyses, School of Pharmaceutical Sciences, University of São Paulo, São Paulo, Brazil

Abstract

Introduction

Single-cell RNA sequencing (scRNA-seq) data analysis is at the core of the current quest to describe all human cell types. The annotation of cell events in scRNA-seq is commonly done using some variatino of the following steps (Luecken and Theis 2019):

- Cluster cells
- Find clusters markers.
- Annotate clusters based on known markers.

The annotations are reported in legends alongside tSNE and UMAP plots, and in metadata files in columns that give each cell a single type label.

Two of the tasks that scientists are interested in are (1) mapping cells in a dataset to *known* cell types and (2) discover *new* groupings with biological relevance.

By biological relevance, we mean that similar cells might be found in other studies, and help building predictions about reality.

Here we focus on the second task: the discovery of new groups.

Final cluster annotations might point out rare, uniform populations, and have been used sucessfully to identify new types, like the airway ionocytes. (Plasschaert et al. 2018)(Montoro et al. 2018).

Another approach for proposing new classifications is to use hierarchical clustering, and thus provide a multilevel perspective on cell identity. An example is the description of the human middle temporal gyrus by Hodge et al ([1]), where a single-hierarchy ontology is provided both in a main figure and a supplementary file.

While such works are already groundbreaking, we identified a gap: current works seldom explore the multi-hierarchy clustering.

We are used to tree-like classifications, a natural side-effect of the macroevolutionary process of vertebrates. Cell type classifications, though, are not tree-like and many cell types have more than one direct parent. [2]

```
Figure not a tree ?
```

Towards that goal, we build *fcoex*, an R package that builds coexpression networks as an scaffold for hypothesis generation about cell types, and describe its application to some datasets.

Results

The fcoex method

The The *fcoex* tool was built bottom up from first principles, with a goal to avoid complicated algorithms and improve understandability.

Our first goal was to come up with a smaller set of genes that globally captured the diversity of the dataset.

Instead of using simply the highly variable genes, we decided to explore symmetrical uncertainty, a correlation metric from information theory that is used in the FCBF algorithm, a popular feature selection algorithm for machine learning with little previous use in biomedical sciences.

To calculate classical entropies we need categorized data, so we implemented in R a set of heuristics do binarized gene expressions (https://bioconductor.org/packages/release/bioc/html/FCBF.html).

Additionally, mutual information is a supervised method, meaning that before using it, we need to have labels for cells. These labels can be obtained after a standard clustering pipeline, and help to convey information about the distribution of cells in the gene expression manifold.

Our pipeline, then, uses a binarized gene expression matrix and preliminary labels to select a set of genes that *globally* separates cells from each other.

These global markers are not necessarily specific to any cluster; they might be specific to multiple clusters, but still provide information to tell them apart.

The selected features, however, share a degree of redundancy. Some pairs of genes have virtually identical expression patterns.

FCBF feature selection is good at finding and removing that redundancy, what is good for later classification tasks.

We decided to take an advantage of this efficiency, and hacked the method: instead of removing genes with redundant patterns, we inverted the process so to identify gene coexpression modules.

The gene coexpression modules yielded by the default pipeline are small by design (10s of genes per module), so to facilitate manual exploration of the coexpression landscape. Moreover, each module has one "header" gene, which expression pattern is most representative of the genes in the module.

The ultimate goatl of the *fcoex* pipeline, though, is not necessarily the modules, but use them to find biologically relevant populations. As modules contain correlated and anti-correlated genes, instead of

Re-clustering of cells. fcoex treats each module as a gene set. For each gene set, it re-classifies the cells using only their expression. Modules capture different biological functions, and provide complementary views on cell identities (Figure 1).
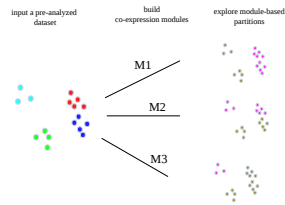
Figure 1: Overview

fcoex recovers multi-hierarchy of blood types

To validate the fcoex pipeline, we selected the well-known pbmc3k dataset from SeuratData, which contains around 2700 peripheral blood mononuclear cells (PBMC) with author-defined cluster labels (Figure 2A). The standard fcoex pipeline detected nine modules which, reassureingly, captured transcriptional pathways known to be active in different blood cells (Table A, SUPP FIGURE PBMC 3K). For example, module M8 contained cytotoxicity genes, as PRF1 and GZMA. he reclustering based on M8 split the dataset into cytotoxic (NK + CD8) and non-cytotoxic cells (Figure 2 B-C). M7 (CD79A) cluster corresponded to B cells, while M5 (HLA-DRB1) grouped monocytes, B cells, and dendritic cells, all known antigen-presenting cells (APC) (https://www.ebi.ac.uk/ols/ontologies/cl/terms? obo_id=CL:0000145). Of note, the APC populations are apart in the UMAP (Uniform Manifold Approximation and Projection for dimensional reduction)(McInnes et al. 2018) representation (Figure 2D), even though the relevance of the class is known acknowledged similarity might have gone unnoticed in the single-cell analysis without fcoex. In general, fcoex clusters are combinations of similar cell types of the original division (Figure 2E), one of which corresponds to a known cell class and the other to the complementary set of cells in the tissue.

fcoex uncovers unknown populations in the zebrafish embryo

After the proof of concept, e explored gene-gene interactions in more depth in a gastrulation dataset of zebrafish cells (75% epiboly, Figure 3 A)(Farrell et al. 2018). The 10 clusters fed to fcoex, though not corresponding to classic types, fed the algorithm with structure from the gene expression space. Using that structure, fcoex identified 8 modules (using default parameters) and the list of modules is displayed in Table B.

The two top ranked modules offer gateways into exploring of zebrafish development biology.

The top-ranked module M1 (MSGN1) harbored the genes msgn1, tbx16, and tbx6, all related to mesoderm development, a core task of gastrulation. In mice, Msgn1 signals via Tbx16(Chalamalasetty et al. 2014) and tbx6l and tbx16 (homologs of Tbx16) play a role together in shaping mesoderm development in zebrafish (Morrow et al. 2017). The population described by this module might be of interest for understanding how mesoderm unfolds.
The second-best modules M2 presented a full pathway - a ligand, apela/Toddler, its receptors, aplnrb, and aplnra, and a putative downstream factor, mespab ((Pauli et al. 2014)(Deshwar et al. 2016). The module contains anticorrelated genes, and the ligand and its receptors are enriched in opposing clusters (TableD, Figure 3C-E).

Materials and methods

Data

The pbmc3k dataset version 3.0.0 was downloaded via the SeuratData package (https://github.com/satijalab/seurat-data). The zebrafish development dataset was downloaded from the Broad Single Cell portal (https://singlecell.broadinstitute.org/single_cell/study/SCP162). Supplementary datasets for mouse Cd11c+ enriched spleen cells and E.7 embryo cells were obtained from Gene Expression Omnibus (GSE54006 and GSE109071, respectively)

Preprocessing

1.0.1 pbmc3k dataset preprocessing pbmc3k data was loaded as a Seurat object from the data package. The expression matrix in the "data" slot and the labels in the "Idents" slotas input for creating the fcoex object.

1.0.2 Zebrafish and mouse datasets preprocessing The data downloaded as log2 transformed counts were loaded in a Seurat object and preprocessed as in Seurat's 3.0 default tutorial. The resolution of the "FindClusters" function was arbitrarily set to yield 10-20 clusters. The normalized expression matrices, and the labels from the FindClusters function were used as input for the fcoex expression. Precise descriptions of the settings used are available in the source code for this paper on https://github.com/lubianat/fcoex_paper.

Gene expression discretization As the original Fast Correlation-Based Filter algorithm was constructed to deal with discrete data, we had to discretize gene counts. This was done with the fcoex package, version 1.0.0 (https://bioconductor.org/packages/fcoex/) We chose as a discretization metric a min-max-percent approach. For each gene, we took the lowest and the highest normalized value across the cells. We set a threshold at 25% of the max-min range. All the values below this threshold were considered "OFF," and all above was "ON". For considerations about the discretization, please see the Supplementary Note.

Identification of fcoex modules

1.0.3 Filtering genes by correlation to labels

After the discretization step, genes were ranked by their correlation to labels (previously assigned by Seurat's 3.1 FindClusters function). The correlation metric we used was the nonlinear Symmetrical Uncertainty, a variation of mutual information that maps the values between 0 (worst) and 1 (best), and accounts for differences in entropy ranges that arise when variables have a different number of classes (number of labels and number of gene classes). All downstream steps were performed only with the previously filtered genes

1.0.4 Finding predominantly-correlated module seeds

Modules were built in a bottom-up approach, first selecting genes predominantly correlated to the labels - a higher symmetrical uncertainty score towards the labels than towards any other gene. These genes, that are the output of the Fast Correlation-Based Filter algorithm, are called the module seeds.

1.0.5 Building the coexpression modules/communities

Each module M is composed of one module seed (x) predominantly-correlated to the label (L) and all the genes (Yi) more correlated to the seed than to the label.

In other words, a gene Yi from all the genes in the Y universe of all genes in the dataset belongs to a module M headed by gene x if and only if it is more correlated to a gene x (from the set X of module seeds) than to the labels.

```
In practice, the algorithm builds an all x all correlation matrix, the
adjacency matrix of the co-expression network. This adjacency matrix is then
trimmed, and edges between nodes Yi and Yj are removed from the network iff
SU(Yi, Yj) < SU(Yi, L) or SU(Yi, Yj) < SU(Yj, L).
```

1.1 Over-representation analysis We performed an over-representation analysis on the human PBMC dataset by Reactome Pathway gene sets processed locally prior to data analysis ("Reactome - a Curated Knowledgebase of Biological Pathways," n.d.). Visualizations in the fcoex package were adapted from the CEMiTool R package (Russo et al. 2018)

Reclustering of cells

```
To recluster the cells based on each module, we use the "recluster" function
of the fcoex module. It uses the gene sets in each co-expression community
to subset the expression table given originally as input. This reduced table
contains the expression values regarding those genes for all the cells in
the dataset.
The distances between cells in this reduced matrix was calculated by the
manhattan distance, and hierarchical clustering was performed. The metric
used to calculate the linkage distance between groups was the "ward.D2"
metric as implemented in the hclust function of the stats package in R
3.6.1. Two groups of cells were retrieved from each  clustering (the k
parameter was set to 2) . The cluster with a higher expression of the module
seed was labeled as SP (Seed Positive) and the complementary cluster
received, then, the label SN (Seed Negative). Plots were generated via the
DimPlot function of the Seurat package, substituting labels of the Seurat
object for the new ones.
```

1.2 Code availability The fcoex package, which performs the coexpression analysis is available at http://bioconductor.org/packages/fcoex/. The discretization and feature selection algorithms are available in a second package, FCBF (http://bioconductor.org/packages/FCBF/). All the analyses performed for this work are available at https://github.com/lubianat/fcoex paper.

References

1. **Conserved cell types with divergent features in human versus mouse cortex**
   Rebecca D. Hodge, Rebecca D. Hodge, Trygve E. Bakken, Jeremy A. Miller, Kimberly A. Smith, Eliza R. Barkan, Lucas T. Gray, Jennie L. Close, Brian R. Long, Nelson Johansen, … Ed S. Lein
   Nature (2019-01-01) https://www.wikidata.org/wiki/Q71306466
   DOI: 10.1038/s41586-019-1506-7

2. **An ontology for cell types**
   Jonathan Bard, Sue Rhee, Michael Ashburner
   Genome Biology (2005-01-01) https://www.wikidata.org/wiki/Q21184168
   DOI: 10.1186/gb-2005-6-2-r21