Statistical Theory and Related Fields

ISSN: 2475-4269 (Print) 2475-4277 (Online) Journal homepage: www.tandfonline.com/journals/tstf20

scTenifoldNet and scTenifoldKnk: A package suite for single-cell gene regulatory network construction, comparison, and perturbation analysis

Yan Zhong, Daniel Osorio, Guanxun Li, Qian Xu, Yongjian Yang, Jianhua Z. Huang & James J. Cai

To cite this article: Yan Zhong, Daniel Osorio, Guanxun Li, Qian Xu, Yongjian Yang, Jianhua Z. Huang & James J. Cai (29 Sep 2025): scTenifoldNet and scTenifoldKnk: A package suite for single-cell gene regulatory network construction, comparison, and perturbation analysis, Statistical Theory and Related Fields, DOI: 10.1080/24754269.2025.2557719

To link to this article: https://doi.org/10.1080/24754269.2025.2557719

Published online: 29 Sep 2025.

STATISTICAL THEORY AND RELATED FIELDS
https://doi.org/10.1080/24754269.2025.2557719

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

OPEN ACCESS Check for updates

scTenifoldNet and scTenifoldKnk: A package suite for single-cell gene regulatory network construction, comparison, and perturbation analysis

Yan Zhong[a], Daniel Osorio[b], Guanxun Li[c], Qian Xu[d], Yongjian Yang[e], Jianhua Z. Huang[f] and James J. Cai[d,e]

[a]KLATASDS-MOE, School of Statistics, East China Normal University, Shanghai, People's Republic of China; [b]QIAGEN Digital Insights, USA; [c]Department of Statistics, Beijing Normal University at Zhuhai, Zhuhai, People's Republic of China; [d]Department of Veterinary Integrative Biosciences, Texas A&M University, College Station, TX, USA; [e]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA; [f]School of Data Science, Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), Shenzhen, People's Republic of China

ABSTRACT

The comparative analysis of gene regulatory networks (GRNs) across various biological conditions reveals crucial shifts in regulatory mechanisms, shedding light on how genetic and environmental signals influence gene function. Recent advances in highresolution technologies have provided support and made it possible to profile gene expression at the single-cell level, thereby enabling more precise studies of transcriptional regulation. We have developed the `scTenifoldNet` and `scTenifoldKnk` R packages, which offer streamlined workflows for constructing single-cell gene regulatory networks (scGRNs) and facilitating comparisons across different samples or between pre- and post-gene perturbation states. Both packages employ a "tensor decomposition + manifold alignment" approach to achieve robust and effective comparisons.

ARTICLE HISTORY
Received 10 March 2025
Accepted 22 August 2025

KEYWORDS
Gene regulatory network; scRNA-seq data; network comparsion; virtual gene knockout; R package

1. scTenifoldNet package

The `scTenifoldNet` package implements the scGRN construction and comparison method introduced in Osorio et al. (2020). The workflow of `scTenifoldNet` starts with two datasets, each containing the raw single-cell RNA sequencing (scRNA-seq) count matrix for a condition. The two datasets should share the same genes, and the order of the genes in the data matrices should also be consistent. The `scTenifoldNet` package constructs the scGRN for each sample and ranks genes according to their differential regulation levels through the following steps.

(1) Preprocessing. Use the `scQC()` function to perform quality control on each scRNA-seq dataset, followed by normalization using the `cpmNormalization()` function.

CONTACT Yan Zhong yzhong@fem.ecnu.edu.cn East China Normal University, No. 3663, North Zhongshan Road, Shanghai, 200062, China

(2) Constructing scGRNs for subsamples. For each sample, use the `makeNetworks()` function to generate multiple subsamples of the data and to construct an initial scGRN for each subsample via principal component regression (Jolliffe, 1982). A $p \times p \times b$ tensor, denoted as $\mathbf{T}$, is used to store the adjacency matrices of these networks, where p represents the number of genes and b represents the number of subsamples.
(3) Denoising via tensor decomposition. For each sample, use the `tensorDecomposition()` function to apply CANDECOMP/PARAFAC (CP) decomposition (Faber et al., 2003) to $\mathbf{T}$, resulting in a denoised, averaged adjacency matrix for the scGRN.
(4) Aligning genes of two samples into a manifold. Use the `manifoldAlignment()` function to perform non-linear manifold alignment on the denoised adjacency matrices of the two samples.
(5) Ranking genes. Use the `dRegulation()` function to calculate the distance between a gene's two projections from the two samples and rank genes in descending order of this distance.

The `scTenifoldNet()` function combines these five steps, offering a more straightforward implementation.

2. An example for scTenifoldNet package

To demonstrate the application of the `scTenifoldNet` package, we use the publicly available ifnb dataset from SeuratData, which includes scRNA-seq data from cells under two conditions, interferon-beta (IFN-β) stimulation and untreated control.

```
library(Seurat)
library(SeuratData)
library(scTenifoldNet)
library(igraph)
library(ggplot2)
library(ggrepel)

# Install dataset from the SeuratData package.
# ifnb dataset contains 6,548 controled cells
# and 7,451 IFNB-stimulated cells, with 14,053 genes
InstallData("ifnb")
ifnb <- LoadData("ifnb")
ifnb <- UpdateSeuratObject(ifnb)

# To ensure cellular homogeneity, select CD14 Mono cells
    for analysis
ifnb_CD14 <- subset(ifnb, subset = seurat_annotations ==
    "CD14 Mono")

# Split the RNA measurements into two layers
ifnb_CD14 <- SplitObject(ifnb_CD14, split.by = "stim")
X_seurat <- ifnb_CD14$CTRL
Y_seurat <- ifnb_CD14$STIM
```

```
# Identify the top 3,000 highly variable genes (HVGs) for
    each state
X_seurat <- NormalizeData
              (X_seurat, normalization.method =
               "LogNormalize", scale.factor = 10000)
X_seurat <- FindVariableFeatures(X_seurat, nfeatures = 3000)
X_HVGs <- VariableFeatures(X_seurat)

Y_seurat <- NormalizeData
              (Y_seurat, normalization.method =
               "LogNormalize", scale.factor = 10000)
Y_seurat <- FindVariableFeatures(Y_seurat, nfeatures = 3000)
Y_HVGs <- VariableFeatures(Y_seurat)

# Use their union for downstream analysis.
HVGs_union <- union(X_HVGs, Y_HVGs)

X <- ifnb_CD14$CTRL[["RNA"]]$counts
Y <- ifnb_CD14$STIM[["RNA"]]$counts
X <- X[HVGs_union, ]
Y <- Y[HVGs_union, ]

# scTenifoldNet
Output <- scTenifoldNet(X = X, Y = Y,
                        nc_nNet = 10, nc_nCells = 500,
                        td_K = 3, qc_minLibSize = 30)
# Structure of the output
str(Output)

# Accessing the computed weight-averaged denoised gene
   regulatory networks
# Network for sample X
graph_from_adjacency_matrix(adjmatrix =
  Output$tensorNetworks$X, weighted = TRUE)

# Network for sample Y
graph_from_adjacency_matrix(adjmatrix =
  Output$tensorNetworks$Y, weighted = TRUE)

# Accessing the manifold alignment result
head(Output$manifoldAlignment)

# Differential Regulation
head(Output$diffRegulation,n = 10)

# Visualization
```

```
# Genes with FDR < 0.1 are labeled as red
set.seed(1)
qChisq <- rchisq(dim(Output$diffRegulation)[1],1)
sorted_qChisq <- sort(qChisq, decreasing = TRUE)
highlight_rows <- which(Output$diffRegulation$p.adj < 0.1)
highlight_genes <- Output$diffRegulation$gene[highlight_rows]
geneColor <- ifelse(Output$diffRegulation$p.adj < 0.1,
   "#9F0000", "gray50")

# Provide data for Figure 1.
plot_data <- data.frame(
  X = sorted_qChisq,
  Y = Output$diffRegulation$FC,
  Gene = Output$diffRegulation$gene,
  Color = geneColor,
  Highlight = Output$diffRegulation$gene %in% highlight_genes
)

# Code for Figure 1.
ggplot(plot_data, aes(x = X, y = Y)) +
  #qqline(qChisq)+
  geom_point(aes(color = Color), size = 2) +
  geom_text_repel(
    data = subset(plot_data, Highlight),
    aes(label = Gene),
    max.overlaps = 100,
    size = 4,
    color = "#2A67A3",
    box.padding = 0.3,
    segment.color = "gray80",
    segment.size = 0.2,
    min.segment.length = 0.1,
    nudge_x = 0.5,
    nudge_y = 5
  ) +
  scale_color_identity() +
  labs(
    x = expression(X^2~Quantiles),
    y = "FC",
    title = "H1"
  ) +
  coord_cartesian(xlim = c(0, max(sorted_qChisq)), ylim =
  c(0, max(Output$diffRegulation$FC)+5)) +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))+
  annotate("point",
           color = "#9F0000",
```

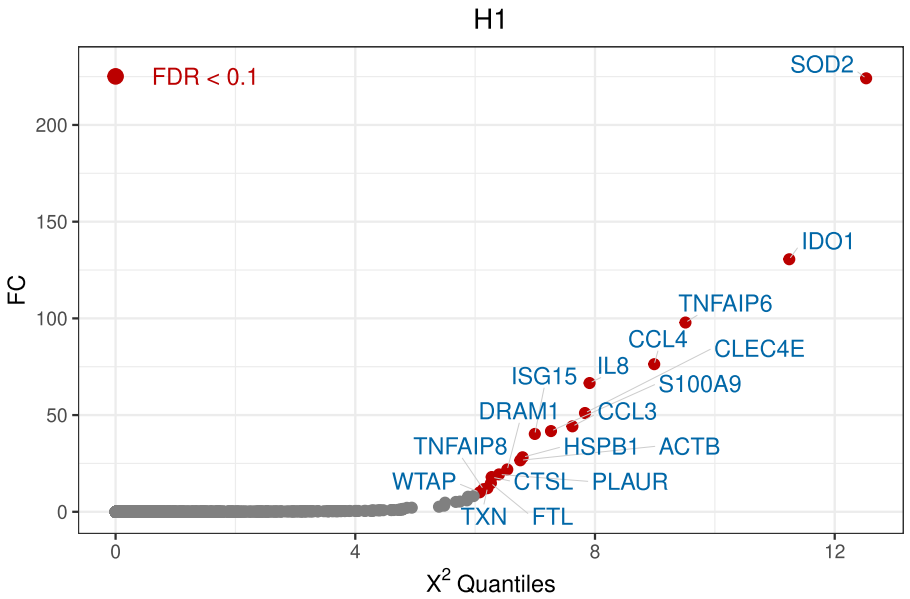

Figure 1. Differentially regulated genes identified by `scTenifoldNet`.

Table 1. Running time of `scTenifoldNet`.

State	nCells (after QC)	nGenes (after QC)	scGRN	Manifold Alignment	Total
CTRL	2167	1041	8.77 min	0.08 min	16.49 min
STIM	2120	1041	7.64 min		

```
            x = 0,
            y = max(Output$diffRegulation$FC)+1,
            size = 3)+
  annotate(
    "text",
    x = 1.5, y = max(Output$diffRegulation$FC)+1,
    label = "FDR < 0.1",
    color = "#9F0000",


    size = 4
  )
```

Figure 1 shows the differentially regulated genes identified by `scTenifoldNet`, reflecting the changes in the gene regulatory patterns between the two conditions. The red points indicate the differentially regulated genes identified by `scTenifoldNet` under an FDR control of 0.1.

The running time of `scTenifoldNet` for this example is summarized in Table 1.

Considering the scalability of `scTenifoldNet`, we note that its running time mainly depends on the time needed to construct scGRNs from subsampled expression matrices, which varies with the number of cells and genes in the input dataset. Table 2 summarizes the

Table 2. Running time to construct an scGRN with different cell and gene sizes.

nCells	nGenes	scGRN
300	1000	3.45 min
1000	1000	4.25 min
1000	5000	171.88 min (2 h 51.6 min)
2500	5000	175.29 min (2 h 55.3 min)
5000	5000	188.88 min (3 h 8.9 min)
5000	7500	189.51 min (3 h 9.5 min)
7500	5000	615.45 min (10 h 15.5 min)
7500	7500	616.12 min (10 h 16.1 min)

running time of `scTenifoldNet` for constructing an scGRN under different scenarios of cell and gene sizes.

3. scTenifoldKnk package

The `scTenifoldKnk` package implements the virtual knockout (KO) experiments proposed by Osorio et al. (2022), which construct and compare the scGRNs of a wildtype (WT) control sample before and after knocking out a target gene. It provides a data-driven alternative to traditional experimental setups by virtually removing the influence of a target gene within a gene regulatory network (GRN). The input of `scTenifoldKnk` is only a count matrix of the scRNAseq data of a WT control sample, and `scTenifoldKnk` provides a ranking for genes that reflects the level of effects made by knocking out the target gene. The `scTenifoldKnk` enables systematic evaluation of a gene's regulatory impact, revealing downstream pathways and affected biological processes without the need for expensive CRISPR libraries. It also supports multi-gene knockout (KO) analysis. Built on the foundation of `scTenifoldNet`, the `scTenifoldKnk` package includes a single function, `scTenifoldKnk()`, which performs the following.

- Constructing scGRNs for WT and virtual KO. Using the `makeNetworks()` and `tensorDecomposition()` functions from the `scTenifoldNet` package, `scTenifoldKnk()` constructs a denoised scGRN for the WT control sample. A pseudo-KO scGRN is then generated by virtually knocking out the target gene.
- Virtual KO analysis with manifold alignment. A quasi-manifold alignment method aligns the scGRNs of the WT and virtual KO samples, constructing two low-dimensional representations for each gene. The distance between these two representations is then calculated for each gene, and this distance is used to rank genes, identifying those that are differentially regulated (DR) in response to the KO gene.

4. An example for scTenifoldKnk package

To evaluate the performance of the `scTenifoldKnk` package, we apply it to the publicly available cbmc dataset from SeuratData, which includes scRNA-seq data of cord blood mononuclear cells.

```
library(Seurat)
library(SeuratData)
library(scTenifoldKnk)
```

```
library(ggplot2)
library(ggrepel)

# Install dataset from the SeuratData package.
# cbmc dataset contains 8,617 cells and 20,501 genes,
# with scRNA-seq and 13-antibody sequencing.
InstallData("cbmc")
cbmc <- LoadData("cbmc")
cbmc <- UpdateSeuratObject(cbmc)

# To ensure cellular homogeneity, select Memory CD4 T cells
   for analysis.
X_seurat <- subset(cbmc, subset = rna_annotations ==
   "Memory CD4 T")


# Identify the top 3,000 highly variable genes (HVGs)
X_seurat <- NormalizeData
            (X_seurat, normalization.method = "LogNormalize",
                scale.factor = 10000)
X_seurat <- FindVariableFeatures(X_seurat, nfeatures = 3000)
X_HVGs <- VariableFeatures(X_seurat)

# Use scRNA-seq of HVGs for downstream analysis.
X <- X_seurat[["RNA"]]$counts
X <- X[X_HVGs, ]

# scTenifoldKnk
Output <- scTenifoldKnk(countMatrix = X, gKO = "BCL2",
   qc_minLSize = 0)

# Accessing the manifold alignment result
head(Output$manifoldAlignment)

# Differential Regulation
head(Output$diffRegulation,n = 10)

# Visualization
set.seed(1)
qChisq <- rchisq(dim(Output$diffRegulation)[1],1)
sorted_qChisq <- sort(qChisq, decreasing = TRUE)
highlight_rows <- which(Output$diffRegulation$p.adj < 0.1)
highlight_genes <- Output$diffRegulation$gene[highlight_rows]
geneColor <- ifelse(Output$diffRegulation$p.adj < 0.1,
   "#9F0000", "gray50")

# Provide data for Figure 2.
```

```
plot_data <- data.frame(
  X = sorted_qChisq,
  Y = Output$diffRegulation$FC,
  Gene = Output$diffRegulation$gene,
  Color = geneColor,
  Highlight = Output$diffRegulation$gene %in% highlight_genes
)

# Code for Figure 2.
ggplot(plot_data, aes(x = X, y = Y)) +
  #qqline(qChisq)+
  geom_point(aes(color = Color), size = 2) +
  geom_text_repel(
    data = subset(plot_data, Highlight),
    aes(label = Gene),
    max.overlaps = 100,
    size = 4,
    color = "#2A67A3",
    box.padding = 0.3,
    segment.color = "gray80",
    segment.size = 0.2,
    min.segment.length = 0.1,
    nudge_x = 0.5,
    nudge_y = 5
  ) +
  scale_color_identity() +
  labs(
    x = expression(X^2~Quantiles),
    y = "FC",
    title = "H1"
  ) +
  coord_cartesian(xlim = c(0, max(sorted_qChisq)), ylim =
  c(0, max(Output$diffRegulation$FC)+5)) +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))+
  annotate("point",
           color = "#9F0000",
           x = 0,
           y = max(Output$diffRegulation$FC)+1,
           size = 3)+
  annotate(
    "text",
    x = 1.5, y = max(Output$diffRegulation$FC)+1,
    label = "FDR < 0.1",
    color = "#9F0000",
    size = 4
  )
```

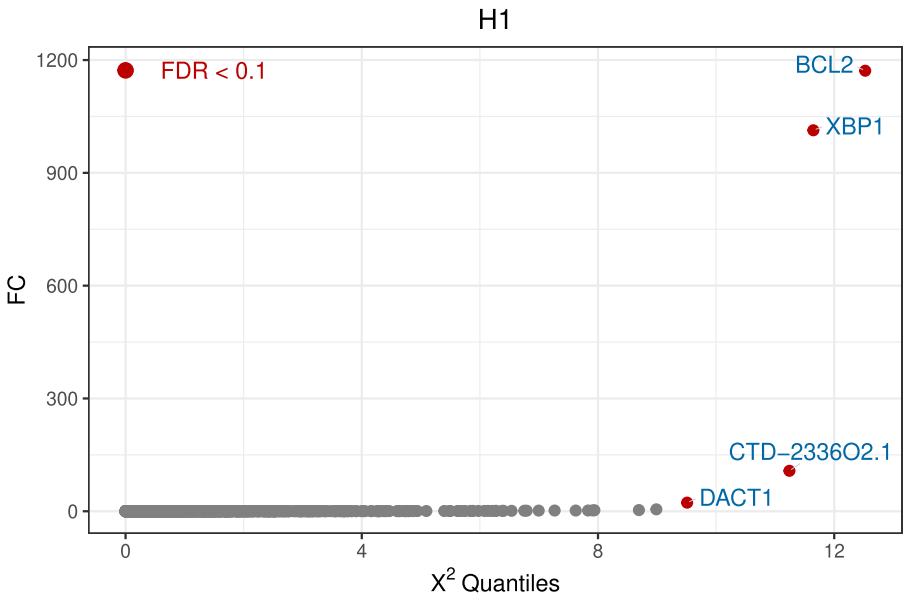

Figure 2. Differentially regulated genes identified by `scTenifoldKnk`.

Table 3. Running time of `scTenifoldKnk`.

nCells (after QC)	nGenes (after QC)	scGRN	Manifold Alignment	Total
1555	1229	13.01 min	0.05 min	13.06 min

With the virtual knockout of the BCL2 gene, the red points in Figure 2 show the associated genes identified by `scTenifoldKnk`, reflecting the potential functional role of BCL2.

The running time of `scTenifoldKnk` for this example is summarized in Table 3.

5. Conclusions

In summary, the `scTenifoldNet` and `scTenifoldKnk` packages offer well-structured and efficient workflows for the analysis of transcriptional regulation and gene-gene interactions. Each package provides unique insights into regulatory genes and pathways, advancing our understanding of cellular mechanisms across a variety of biological contexts. More examples can be found in `https://github.com/cailab-tamu/scTenifoldNet/tree/master` and `https://github.com/cailab-tamu/scTenifoldKnk/tree/master`.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Faber, N. K. M., Bro, R., & Hopke, P. K. (2003). Recent developments in CANDECOMP/PARAFAC algorithms: A critical review. *Chemometrics and Intelligent Laboratory Systems*, *65*(1), 119–137. https://doi.org/10.1016/S0169-7439(02)00089-8

Jolliffe, I. T. (1982). A note on the use of principal components in regression. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *31*(3), 300–303.

Osorio, D., Zhong, Y., Li, G., Huang, J. Z., & Cai, J. J. (2020). scTenifoldNet: A machine learning workflow for constructing and comparing transcriptome-wide gene regulatory networks from single-cell data. *Patterns*, *1*(9), Article 100139. https://doi.org/10.1016/j.patter.2020.100139

Osorio, D., Zhong, Y., Li, G., Xu, Q., Yang, Y., Tian, Y., Chapkin, R. S., Huang, J. Z., & Cai, J. J. (2022). scTenifoldKnk: An efficient virtual knockout tool for gene function predictions via single-cell gene regulatory network perturbation. *Patterns*, *3*(3), Article 100434. https://doi.org/10.1016/j.patter.2022.100434