LEAP Documentation --- Lag-based Expression Association for Pseudotime-series

Starting with pseudo time ordered data, LEAP calculates the Pearson's correlation of normalized mapped-read counts over temporal windows of a fixed size with different lags. The score recorded for a pair of genes is the maximum Pearson's correlation over all the values of lag that the method considers. The software includes a permutation test to estimate false discovery rates. Since the correlation computed is not symmetric, this method can output directed networks.

Advances in sequencing technology now allow researchers to capture the expression profiles of individual cells. Several algorithms have been developed to attempt to account for these effects by determining a cell's so-called "pseudo-time", or relative biological state of transition. By applying these algorithms to single-cell sequencing data, we can sort cells into their pseudo temporal ordering based on gene expression. LEAP then applies a time-series inspired lag-based correlation analysis to reveal linearly dependent genetic associations.

LEAP utilizes the estimated pseudo-time of the cells to find gene co-expression that involves time delay. Gene co-expression networks (GCNs) use nodes to represent genes and edges to represent co-expression (simultaneous expression/silence, or simultaneously high/low expression) of genes, and they can be used to predict gene functions. A popular way of constructing a GCN is called the "correlation-based" approach, which connects gene pairs whose expressions in different biological samples are highly correlated, measured by Pearson's correlation or other correlation coefficients. Biologically, if a gene enhances/inhibits another gene, then the latter gene will have delayed expression/silence. For such a pair, the co-expression of the two genes is strong if the delay in time is taken into account but can be weak if only simultaneous association is considered. Single-cell RNA-Sequencing is able to capture this time information. ScRNA-Seq measures the gene expression profile of each individual cell, and hundreds to thousands of cells in a single run. These cells are at different time points of their cell cycles, and these time points can be estimated based on the idea that expression profiles are similar in cells at similar time points. These estimated time points are called "pseudo-time".

LEAP computes gene co-expression and takes into account the possible lags in time. LEAP sorts cells according to the estimated pseudo-time and then computes the maximum correlation of all possible time lags. This maximum correlation is used as the statistic to replace the traditional Pearson's correlation coefficient for constructing the network, and the statistical significance of this statistic is measured by the false discovery rate calculated using permutations. LEAP works by calculating the correlation of normalized mapped-read counts over varying lag-based windows. LEAP discovers much more gene regulatory associations as it is able to take the time lag into account. LEAP is able to capture associations that were hidden by the time lags. The asymmetric associations detected by LEAP more likely reflect regulatory relationships as they describe which gene follows another gene in expression.

Note:

The pseudo-time was estimated using Monocle3, which works by first mapping the gene expressions to low-dimensional space and then finding the longest path along a minimum spanning tree of the cell's locations.

Input

A data matrix where rows are genes and columns are experiments, sorted by their pseudo-time. The dataset used here contains 100 genes across 200 single-cell sequencing experiments, sorted by pseudo-time using the package Monocle3.

Main function:

1. MAC_counter: Perform lag-based correlation analysis of single-cell sequencing data, sorted by pseudo-time.
2. MAC_perm: Perform a permutation analysis to determine a cutoff for significant MAC values.
3. MAC_lags: Internal function used by MAC_counter and MAC_perm, performs the lag-based correlation analysis.

Resulting datasets.

1. MAC_example: The resulting matrix of MACs from applying MAC_counter() to original dataset.
2. Lag_example: The resulting lag matrix from applying MAC_lags() to original dataset.
3. Perm_example: The resulting data output from applying MAC_perm() to original dataset.

References

https://academic.oup.com/bioinformatics/article/33/5/764/2557687