TAD Map

Release 0.1.0

Rohit Singh

Oct 30, 2021

CONTENTS

Topologically associating domains (TADs) are contiguous segments of the genome where the genomic elements are in frequent contact with each other. Genes that cooccupy a TAD are often functionally related and more likely to be coexpressed than random gene pairs, The TAD Map (preprint to come) provides a consensus estimates of this layout in human and mouse, aggregated from multiple experimental datasets. A useful interpretation of the TAD Map is as a grouping of genes along the genome that share a similar chromatin neighborhood and may be co-regulated. Accordingly, this package also provides functionality to map any single-cell RNA-seq dataset to *TAD signatures*, where gene expression is grouped by TADs, and mapped to per-TAD activation probabilities in each cell.

Read the documentation. We encourage you to report issues at our Github page ; you can also create pull reports there to contribute your enhancements. If the TAD Map is useful in your research, please consider citing our preprint bioRxiv (2021).

CHAPTER

ONE

OVERVIEW

Topologically associating domains (TADs) are contiguous segments of the genome where the genomic elements are in frequent contact with each other. Genes that cooccupy a TAD are often functionally related and more likely to be coexpressed than random gene pairs, The TAD Map (bioRxiv (2021)) provides a consensus estimate, aggregated from multiple experimental datasets, of this layout in human and mouse. It also provides tools to map any single-cell RNA-seq dataset to *TAD signatures*, where gene expression is mapped to TAD activation probabilities in each cell.

1.1 Paper & Code

TAD Map and TAD signatures are described in the preprint *TBD* (http://doi.org/10.1101/TBD)

Source code available at: https://github.com/rs239/tadmap

CHAPTER

TWO

INSTALLATION

We recommend Python v3.6 or higher.

2.1 PyPI, Virtualenv, or Anaconda

You can use `pip` (or `pip3`):

```
pip install tadmap
```

2.2 Docker

TAD Map has been designed to be compatible with the popular and excellent single-cell Python package, Scanpy. We recommend installing the Docker image recommended by Scanpy maintainers and then using `pip`, as described above, to install Schema in it.

CHAPTER

THREE

API

tadmap.**compute_tad_signature**(*adata*, *sp_2_letter*)

Given an AnnData object and a species (*hs* or *mm*), compute a TAD activation profile for each cell

The activation profile is computed by fitting a 2-component Poisson mixture model using the Expectation Maximization (EM) algorithm. One component corresponds to TADs that are transcriptionally active (i.e., "ON"), while the other corresponds to "OFF" TADs. However, even "OFF" TADs can have genes with active expression (e.g. isolated expression of a single gene in a non-TAD-dependent fashion). For each cell, the EM algorithm computes for each TAD — there are approx 3000 of them in human and mouse — the probability that the TAD is in "ON" state.

Parameters

- **adata** (*AnnData object*) – AnnData object (*n* cells). The gene expression matrix can be sparse or dense and contain counts or log1p-transformed data— the method will try to adapt accordingly.
- **sp_2_letter** (*string*) – one of 'hs' or 'mm' Currently, TAD Maps are supported only for human ('hs') or mouse ('mm')

Returns

a pair of Pandas dataframes: the TAD signature and auxiliary information, respectively

The first dataframe is of dimensionality n x T where n is the number of cells and T is the number of TADs. The algorithm will filter out TADs which had no active genes in the data so T may vary a little across datasets. The column names correspond to TAD names which are in the following format: *<numeric_id>|<chromosome>|<start>|<end>*

The second dataframe contains one row per (TAD,gene) pair. Some genes may span two TADs and will have two rows. Each row contains the TADs score dispersion, an indication of its variability, similar to highly variable genes. Specifically, here are the column names of this dataframe:

- *tad_name*: see above
- *tad_gene_count*: number of protein-coding genes partially/fully contained in the TAD
- *lambda_ON*: lambda for the Poisson corresponding to "ON" TADs (same for all rows)
- *lambda_OFF*: lambda for the Poisson corresponding to "OFF" TADs (same for all rows)
- ***tad_activation_mean*: the average probability score of activation for this TAD** across *n* cells
- ***tad_activation_disp*: variance/mean of probability score of activation for this** TAD across *n* cells. Use this as the measure for identifying highly variable TADs

- ***gene*: Ensembl v102 name of a gene contained in the TAD.** There is one row for each (gene,TAD) pair

tadmap.**read_Ensembl_v102_refdata**(*sp*)

Returns a dataframe consisting of gene symbology, location etc. for Ensembl v102

Parameters **sp** – string, one of *hs* (human) or *mm* (mouse)

Returns A dataframe with gene name, chromosome, strad direction, TSS etc.,

tadmap.**read_TADMap_from_file_or_url**(*tad_file_or_url*)

Given a file that contains the TAD Map in the default format, parse it and return the results

The file format needs to match the ones currently hosted at *https://cb.csail.mit.edu/cb/tadmap/TADMap_geneset_{hs,mm}.csv*

In particular, it is a CSV format of the form:

<tad_name>,[semi-colon separated list of genes partially/fully contained in the TAD]

Gene may be specified by multiple pipe-separated identifiers, with first one being the canonical,

Parameters **tad_file_or_url** – string specifying file location or URL

Returns

the pair (*geneset*, *tad2genelist*)

- geneset: the set of all genes listed in the file. Each gene is specified as a 3-tuple: (canonical_name, Ensembl_name, MGI_or_HGNC_name)
- tad2genelist: a dictionary from tad_name -> list of genes. Only the canonical_name of the gene is provided in the list

tadmap.**retrieve_TADMap_by_species**(*sp_2_letter*)

Retrieve the pre-computed TAD MAp from the default remote location

The default location is *https://cb.csail.mit.edu/cb/tadmap/TADMap_geneset_{hs,mm}.csv*

Parameters **sp_2_letter** – string, specifying *hs* (human) or *mm* (mouse). These are the only species supported currently.

Returns the same output as *read_TADMap_from_file_or_url*, which is called by this function

tadmap.**set_loglevel**(*l*)

Set the loglevel to one of logging.{ERROR,WARNING,INFO,DEBUG}. Default = WARNING

tadmap.**standardize_adata_gene_names**(*adata_in*, *sp_2_letter*)

Process the gene names in an AnnData object to map them to Ensembl v102 names, removing those that do not match.

This is an preprocessing step to call before calling *compute_tad_signature*, since the rest of this module only works with Ensembl v102 canonical names.

Parameters

- **adata_in** – the input AnnData object, with gene names in *adata_in.var_names*
- **sp_2_letter** – string specifying the species: *hs* (human) or *mm* (mouse)

Returns AnnData object, with var_names mapped to Ensembl v102, duplicates and non-mappable names removed.

tadmap.**to_log_odds**(*tad_occupancy_df*)

Convert probability scores *p* to log-odds, *log(p/(1-p))*

This is a useful conversion to do before passing TAD signatures to a clustering or visualization process. It widens the range of values and makes them more compatible with the Euclidean distance metric, which underlies many clustering and visualization algorithms.

Parameters `tad_occupancy_df` – Pandas dataframe

This is the first dataframe item in the pair of dataframes returned by *compute_tad_signature*

Returns Pandas dataframe, same dimensions as the input

CHAPTER

FOUR

REFERENCES

Code: Github repo

Preprint: If you use the TAD Map or TAD signatures, please consider citing *Deciphering the species-level structure of topologically associating domains* (bioRxiv)

Project Website: http://tadmap.csail.mit.edu

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

t

INDEX