
netZoomM Documentation

Release 0.2

netZoomM

Oct 23, 2020

Contents:

1	Home	1
2	Contents	3
2.1	Installation guide	3
2.1.1	Requirements	3
2.1.2	Install	3
2.1.3	Troubleshooting	3
2.2	Functions	3
2.2.1	panda_run	3
2.2.2	lioness_run	4
2.2.3	RunPUMA	5
2.2.4	UpdateDiagonal	6
2.2.5	Tfunction	6
2.2.6	SavePairs	7
2.2.7	Pairs2Mat	7
2.2.8	PANDA	8
2.2.9	PUMA	8
2.2.10	NormalizeNetwork	9
2.2.11	Coexpression	9
2.3	Tutorials	9
2.4	Changelog	10
2.4.1	0.5.1 (2020-06-27)	10
2.4.2	0.5.0 (2020-05-02)	10
2.4.3	0.4.3 (2020-03-06)	10
2.4.4	0.4.2 (2020-02-21)	10
2.4.5	0.4.1 (2020-01-18)	10
2.4.6	0.4.0 (2019-12-10)	10
2.4.7	0.3.0 (2019-11-17)	10
2.4.8	0.2.0 (2019-11-13)	11
2.4.9	0.1.0 (2019-5-28)	11
3	Indices and tables	13

CHAPTER 1

Home

netZoom is a catalog of methods for the reconstruction and analysis of network biology methods.

netZoom has started as a merge of the following repositories and forks:

- Cho-Yi Chen's [MATLAB PANDA](#) which was forked from Kimberly Glass's [PANDA](#).
- Marieke Kuijjer's [PUMA](#)

2.1 Installation guide

To install netZoom on your computer, please check the following requirements:

2.1.1 Requirements

- MATLAB (tested on 2013a -> 2019a)
- Octave (Some functions are cross-compatible with Octave please check the function description)

2.1.2 Install

- `git clone https://github.com/netZoo/netZoom.git`
- In MATLAB console, add netZoom to your path through `addpath(genpath('netZoom'))`

2.1.3 Troubleshooting

- To report any installation issue or function bug, please report through opening an [issue](#) on github.

2.2 Functions

2.2.1 panda_run

Description:

```
Using PANDA to infer gene regulatory network.
1. Reading in input data (expression data, motif prior, TF PPI data)
2. Computing coexpression network
3. Normalizing networks
4. Running PANDA algorithm
5. Writing out PANDA network (optional)
```

Inputs:

```
exp_file : path to file containing gene expression as a matrix of size (g,
↪g)
motif_file: path to file containing the prior TF-gene regulatory network_
↪based on TF motifs as a matrix of size (t,g)
ppi_file : path to file containing TF-TF interaction graph as a matrix of_
↪size (t,t)
panda_out : path to save output PANDA network
            '*.txt': the final network will be saved in .txt format
            '*.tsv': the final network will be saved in .tsv format
            '*.*' : the final network will be saved in .mat v6 format
            '' : the final network will not be saved
save_temp : path to save updated ppi, co-expression, and gene regulation_
↪network
            '': the networks will not be saved
alpha : learning parameter for the PANDA algorithm
save_pairs: (Optional) boolean parameter
            1: the final network will be saved .pairs format where each_
↪line has a TF-gene edge (Cytoscape compatible)
            0: the final network will not be saved in .pairs format
```

Outputs:

```
AgNet : Predicted TF-gene gene complete regulatory network using PANDA_
↪as a matrix of size (t,g).
```

Authors:

```
cychen, marieke, kglass
```

Notes:

```
Script adapted from Marieke's pLIONESSpcss.m, modified to run PANDA only.
```

Publications:

```
https://doi.org/10.1371/journal.pone.0064832
```

2.2.2 lioness_run

Description:

```
Using LIONESS to infer single-sample gene regulatory networks.
1. Reading in PANDA network and preprocessed middle data
2. Computing coexpression network
3. Normalizing coexpression network
4. Running PANDA algorithm
5. Writing out LIONESS networks
```


Inputs:

```

exp_file : path to file containing gene expression as a matrix of size (g,
↪g)
motif_file: path to file containing the prior TF-gene regulatory network_
↪based on TF motifs as a matrix of size (t,g)
ppi_file : path to file containing TF-TF interaction graph as a matrix of_
↪size (t,t)
panda_file: path to the PANDA generated gene regulatory network
save_dir : path to save directory. if It does not exist, it will be_
↪created.
START : index of first sample to generate predicted gene regulatory_
↪network.
END : index of last sample to generate predicted gene regulatory_
↪network. There will be END-START+1 single network samples generated.
-1: use the index of the final gene expression sample
alpha : learning parameter for the PANDA algorithm
ascii_out : 1 : save LIONESS networks in .txt format
0 : save LIONESS networks in .mat -v6 format
lib_path : path to library

```

Outputs:

```

PredNet : Predicted single sample network as a matrix of size (t,g)
This output is directly saved as a file and not
as workspace variable

```

Authors:

```

cychen, marieke, kglass

```

Publications:

```

https://doi.org/10.1016/j.isci.2019.03.021

```

2.2.3 RunPUMA

Description:

```

PUMA can reconstruct gene regulatory networks using both transcription_
↪factors and microRNAs as regulators of mRNA expression levels.

```

Inputs:

```

exp_file : path to file containing gene expression as a matrix of size (g,
↪g)
motif_file: path to file containing the prior TF-gene regulatory network_
↪based on TF motifs as a matrix of size (t,g)
ppi_file : path to file containing TF-TF interaction graph as a matrix of_
↪size (t,t)
outtag : path to save output PUMA network in .pairs format.
mir_file : path to file containing microRNA file
alpha : learning parameter for the PUMA algorithm

```

Outputs:

```
RegNet      : Predicted TF-gene gene complete regulatory network using  
↳PANDA as a matrix of size (t,g).
```

Authors:

```
Marieke Kuijjer
```

Publications:

```
https://www.ncbi.nlm.nih.gov/pubmed/28506242
```

2.2.4 UpdateDiagonal

Description:

```
Updates the diagonal of diagMat
```

Inputs:

```
diagMat: diagonal matrix  
num      : integer  
alpha    : learning rate of the PANDA algorithm  
step     : step number in the PANDA algorithm
```

Outputs:

```
diagMat: updated diagonal matrix
```

Authors:

```
Kimberley Glass
```

2.2.5 Tfunction

Description:

```
Updates Amat matrix using matrices X and Y
```

Inputs:

```
X      : adjacency matrix  
Y      : adjacency matrix
```

Outputs:

```
Amat: updated matrix from X and Y
```

Authors:

```
Kimberly Glass, cychen
```

Notes:

```

    bsxfun is more memory-efficient and faster than repmat implementation for
↪ large arrays.
    MATLAB uses BLAS routines to do matrix multiplication. MATLAB parser
↪ recognizes X*X' as
    a symmetric matrix multiply and will call the symmetric matrix multiply
↪ routine (only
    calculates about 1/2 the answer and then fills in the rest with copies,
↪ which is faster).

```

2.2.6 SavePairs

Description:

```

    A function to save a complete graph in matrix format to a pairs format where
↪ each line represents an edge in the network.
    The output file will have as much lines as edges in the network and will
↪ have the predicted edge weights as well as the
    binary edge weights from the prior motif data.

```

Inputs:

```

TFNames   : names of t TFs
GeneNames : names of g genes
AgNet     : predicted gene regulation network using PANDA of size (t,g)
RegNet    : prior gene regulation network obtained using TF motif scan of
↪ size (t,g)
outtag    : name of saved file

```

Authors:

```

Kimberley Glass, Marouen Ben Guebila

```

2.2.7 Pairs2Mat

Description:

```

Pairs2Mat transforms a TF-Gene network in .pairs network to a complete
matrix. It can also save the prior in matrix format.

```

Inputs:

```

networkPair: path to network in .pairs format with nGenes*nTFs
            edges
nGenes     : number of genes in network
prior      : 0: matNet is a matrix of the final network
            1: matNet is a matrix of the prior network

```

Outputs:

```

matNet     : network in nGenes by nTfs matrix format

```

Authors:

Marouen Ben Guebila 6/19

2.2.8 PANDA

Description:

PANDA infers a gene regulatory network **from gene** expression data, motif prior, **and** PPI between transcription factors

Inputs:

RegNet : motif prior of gene-TF regulatory network
GeneCoReg: gene-gene co-regulatory network
TFCoop : PPI binding between transcription factors

Outputs:

RegNet : inferred gene-TF regulatory network

Authors:

Kimberley Glass

Publications:

<https://doi.org/10.1371/journal.pone.0064832>

2.2.9 PUMA

Description:

PUMA can reconstruct gene regulatory networks using both transcription_↪factors **and** microRNAs **as** regulators of mRNA expression levels.

Inputs:

RegNet : motif prior of gene-TF regulatory network
GeneCoReg: gene-gene co-regulatory network
TFCoop : PPI binding between transcription factors
alpha : learning rate
s1, s2, t1, t2 are indices of miR interactions **in** TFCoop, the TF-TF PPI_↪network.

Outputs:

RegNet : inferred gene-TF regulatory network

Authors:

Marieke Kuijjer

Publications:

<https://www.ncbi.nlm.nih.gov/pubmed/28506242>

2.2.10 NormalizeNetwork

Description:

Normalize an `input` network (matrix) `X`

Inputs:

`X`: network adjacency matrix

Outputs:

`normMat`: nromalized adjacency matrix

Authors:

Kimberley Glass

2.2.11 Coexpression

Description:

Compute gene-gene coexpression network `for` a sample-by-gene matrix `X`
Note that each gene `is` a column `in` `X`

Inputs:

`X`: sample-by-gene matrix

Outputs:

`GeneCoReg`: gene-gene coexpression network

Authors:

Kimberley Glass

2.3 Tutorials

Tutorials in netZoom are the following case studies:

- Reconstruction of gene regulatory networks using PANDA
- Generating single-sample gene regulatory networks using LIONESS
- Differential targeting analysis using PANDA and LIONESS
- Using GRAND database's API for reproducible network reconstruction
- Accurate reconstruction of gene regulatory networks using optPANDA

- gpuPANDA and gpuLIONESS: Accelerated inference of gene regulatory networks using the Graphical Processing Unit

2.4 Changelog

2.4.1 0.5.1 (2020-06-27)

- Major fix for OTTER behavior across platforms.

2.4.2 0.5.0 (2020-05-02)

- new tool: OTTER
- (Cross-platform) Unit test for OTTER

2.4.3 0.4.3 (2020-03-06)

- gpuPANDA and gpuLIONESS
- gpuPANDA and gpuLIONESS tutorials

2.4.4 0.4.2 (2020-02-21)

- optPANDA
- optPANDA tutorial

2.4.5 0.4.1 (2020-01-18)

- Cross-platform harmonization of PANDA behavior.
- Data processing was set to 'Union' by default.

2.4.6 0.4.0 (2019-12-10)

- SPIDER

2.4.7 0.3.0 (2019-11-17)

- PUMA
- differential targeting tutorial
- GRAND database tutorial

2.4.8 0.2.0 (2019-11-13)

- LIONESS
- PANDA tutorial
- LIONESS tutorial

2.4.9 0.1.0 (2019-5-28)

- Changelog added to the doc
- PANDA's CY Chen implementation that is different from the original implementation on the following points
We reach a 20-25% reduction in computation time and memory usage in comparison to its early version. The following optimization has been implemented in this version:

Network Normalization:

Use MATLAB built-in zscore function instead of calculation from scratch. -> 30% speed-up.

T-function:

Use bsxfun instead of repmat, use symmetric matrix multiplication, and reuse summed-square vector. -> 25% speed-up.

PANDA Function:

Move out network normalization from PANDA function to main program to remove unnecessary repeated normalization in each following LIONESS iteration: both PPI network and motif network need to be normalized only once. -> 1-10 sec reduced for each LIONESS iteration depending on the network sizes.

Save W matrix ($R+A$) so that we don't have to compute it twice. -> ~0.5% overall speed-up.

Check hamming inside the while-loop to reduce the last unnecessary updates on TFcoop and GeneCoReg networks. -> ~2% overall speed-up.

I/O:

Use MATLAB binary files (mat-file) instead of text files for I/O which boost the performance (>10x faster).

Reusing the processed middle files:

Save the input expression matrix (transposed), normalized motif/PPI networks, and the aggregated PANDA network to binary files (mat-file) for later use in each LIONESS run: expression matrix can only be saved to v7.3 mat file (compressed Matlab HDF5 format) when its size is over 2GB; normalized motif/PPI networks and output PANDA network can be saved as v6 uncompressed mat files for best I/O efficiency.

Computing gene-gene coexpression matrix:

Move out this part as an independent function so that no need to transpose the input matrix each time in LIONESS iteration. Consequently, we save a copy of expression matrix in the memory and also one matrix transpose operation overhead for each single-sample LIONESS run.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`