Predict gene networks using Ordinary Differential Equation (ODE) based method

Wei Xiao, Yin Jin, Darong Lai, Xinyi Yang, Yuanhua Liu, Christine Nardini

1 Introduction

GOAL: The NTW package allows the computation of the interaction network of n genes (`A`, nxn) based on m independent experiments collecting gene expression profiles (`X`, nxm) with or without the associated transcriptional pertubation matrix (`P`, nxm, contains the direct targets of each of the m perturbation done on the system). The approach is based on ordinary differential equations (ODE) and 3 options for multiple regression.

The input data (`X`) is in tabular form (nxm) where rows represents different genes (n), columns represents perturbation or samples (m) and the content of the tables' cells is the abundance of the gene in the sample. Microarray experiments are the data of choice of this application, but the method can be applied to any data in the appropriate format (miRNA arrays, RNA-seq data, etc.). The results are two matrixes. The first one is `est.A` (nxn), where each cell represents the association computed among the corresponding genes. Because of the computation method, the elements of `A`, corresponding to gene-gene interactions will often be named regressors in the following. The second one is `est.P` (nxm), where each cell (P_{il}) represents the transcriptional perturbation of gene i in experiment l. $P_{il} = 1$ indicates that gene i is directly perturbed in perturbation l. If an option of basing on some prior estimation is chosen, an initial guess of matrix `A`, called `pred.net` (nxn) in the program, can be input set, the default value is NULL.

Like the other gene-network reconstruction methods described in [1, 2], NTW (Network of Transcripts Wirings) is an Ordinary Differential Equations (ODE) based model. This approach frames the reverse engineering problem with the flexible identification of a function that describes the variation of a gene's expression across m experiments, $\underline{x}$ (m dimensional vector), over time as:

$$\underline{x}' = f(\underline{x}, \underline{p}) \tag{1}$$

where f models how the transcriptional perturbations $\underline{p}$ lead to the new equilibria in $\underline{x}$. As stated in [1], we used an approximated version of this approach, leading to the

linearized matricial form,

$$AX = -P \tag{2}$$

where A represents the interaction network (adjacency matrix, $n \times n$), X the steady state expression values (expression matrix, $n \times m$), and P the transcriptional perturbations (also an expression matrix, $n \times m$). Differently from the original work in [1, 2], NTW can reconstruct A in the absence of a known matrix P, since transcriptional perturbations P can be an important unknown of the problem when reconstructing gene networks. Namely, NTW processes X to rank genes in each experiment by their absolute expression values, and selects $TopK$ genes with the highest values as potential target genes. It then produces a Boolean matrix IX where $ix_{ij} = 1$ if entry x_{ij} in matrix X is a potential transcriptional trigger, and $ix_{ij} = 0$ otherwise. Matrix A can then be computed row-wise by using multiple regression of each row of P on the corresponding row of X. To compute the i_{th} row a_i^T of A, NTW iteratively searches an optimal vector P_i as the i_{th} row of P. The searching space of P_i consists of a non-empty subset of the power set $S = 2^{k|IX_{ik} \neq 0, 1 \leq k \leq m}$. As a result, the cardinality of the searching space of P_i is $2^{|S|} - 1$, where $|S|$ represents the number of elements in set S. Given the upper value of $TopK$, the number of non-zero entries in each row of IX is small and so is the cardinality of S and the searching space of P_i. The identification of optimal P_i and thus a_i^T is then done through any of regression methods.

NTW offers different approaches for the inference of `A` using `X` (and `P`). In particular, to solve Equation 2, NTW can use the following methods: `geo` (error-in-variable model), `sse` (ordinary linear regression) and `ml` (maximum likelihood method). The result are the estimated gene interaction network (`A`) and, when unknown as input, transcriptional perturbation matrix (`P`). NTW also offers an option to input prior knowledge of the network `A`, when necessary or available (for example from literature). Basically, when some of the regressors of A are known, `NTW` can receive as input an extra matrix (`pred.net`) which contains the known regressors. The final network `A` is again estimated using the 3 alternative methods (`geo`, `sse` and `ml`) in `forward` (`pred.net` contrains less edges than necessary, typicaly information from literature, or from another algorithm) or `backward` (`pred.net` contrains more edges than necessary, typically when estimated by another algorithm).

Function `NTW` is the main function in this package. It can be used to predict the whole gene interaction network (`A`) and the associated transcriptional pertubation matrix (`P`). The major arguments include:

`cFlag` the type of method used, `geo`,`sse` and `ml`

`pred.net` the option for whether a prior estimation of the network is exploited (no prior information if `pred.net` is NULL), the final network is estimated based on it via `forward` or `backward` method (see `sup.drop`), if there is no prior network

estimation, the `backward` method is used as default (in this case it is assumed that the prior network is full)

`numP` the option to set the maximum number of perturbed times for each gene in all experiments

`restK` the vector numbers of maximum connections of each gene, a vector with higher numbers leads to longer running time. we suggest 20 to 30 percent of the number of genes based on the sparsity assumption of gene interaction matrix for reasonable computation time (however computation time can be improved with parallelization, see below)

`topD` the number of branches in each level of a tree in the grid algorithm: we use a grid algorithm to get the optimized solution and the process, basically while new regressors are tested the best TopD solutions are preserved to the next step (`forward` or `backward`) to add or delete the following regressor. This generates a tree of solutions. The choice of this value determines the effect as well as the speed of the optimization, we suggests 20 to 30 percent of the number of genes ;

`topK` the number of potential target genes in each perturbation: to reduce time for calculation, we suggest `topK` to be 10 to 20 percent of the number of genes

In this package, given a known `P`, `A` is estimated row by row with `A.estimation.Srow`. Otherwise, both `A` and `P` are estimated with `AP.estimation.Srow`. `AP.estimation.Srow` and `A.estimation.Srow` can be used independently so that estimation of each row can be performed in parallel, for improving computation time.

The functions' dependencies scheme of the NTW package is illustrated in Figure 1.

- `NTW`, the main function to estimate the gene interaction matrix `A` and the perturbation targets matrix `P`.
- `P.preestimation`, give a rough estimation of perturbation matrix, according to which a guess of non-zero element in each row of `P` is made.
- `AP.estimation.Srow`, estimation of a single row in gene interaction matrix `A` and perturbation targets matrix `P`.
- `A.estimation.Srow`, estimation of a single row in gene interaction matrix `A` with `P` known.
- `backward`, estimate the network using a `backward` mode to treat the prior information, less edges of the genes are considered.
- `forward`, estimate the network using a `forward` mode to treat the prior information, more edges of the genes are considered.

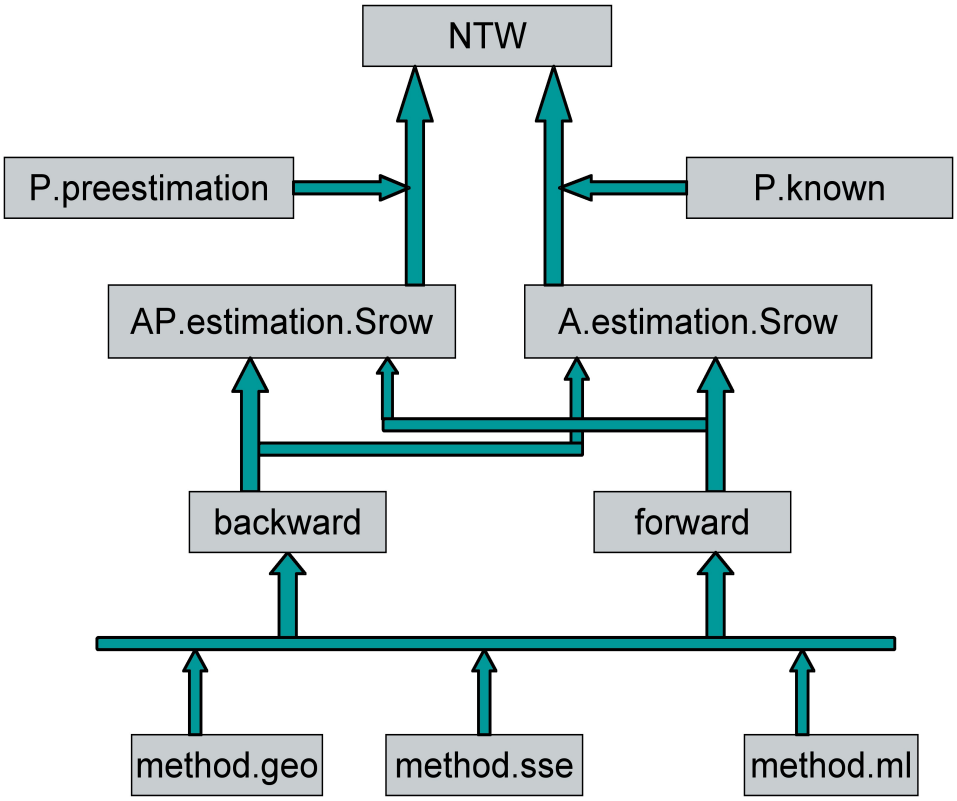

Figure 1: Scheme of the functions in NTW package

- `method.geo`, estimate the network using `geo` method (objective function: geometric mean) with fixed perturbations.
- `method.sse`, estimate the network using `sse` method (ordinary linear regression), with fixed perturbations.
- `method.ml`, estimate the network using `ml` (maximum likelihood method), with fixed perturbations).
- `com.matrix`, creates all combinations of regressorsto be tested for `A`, with the chosen regressor method. The most (`TopD`) successful compbinations are preserved in the following (`forward` or `backward`) step.

2 Beginning NTW

We use the RT-PCR data of 9 genes in SOS pathway of *Escherichia coli* [1] to introduce the usage of NTW package. The main function in this package is `NTW`,

used to estimate the whole gene interaction matrix `A` and the perturbation targets matrix `P`. Estimation of a single row of `A` and `P` independently is also available with the function `AP.estimation.Srow`. This is to supply a faster computation if large quantity of genes is involved, as individual row prediction can be distributed to several CPUs in parallel. In addition, the prediction of the single row of `A` is also possible when `P` is known. Details are stated below.

2.1 `NTW` to estimate the whole gene interaction matrix A and the perturbation targets matrix P

```
> library(NTW)
> library(mvtnorm)
```

Load the SOS pathway data.

```
> data(sos.data)
> X<-sos.data
> X<-as.matrix(X)
> X
```

```
      p.recA p.lexA  p.ssb p.recF p.dinI p.umuDC p.rpoD p.rpoH p.rpoS
recA   0.906 -0.132 -0.139  0.187  0.291   0.000 -0.077  0.000  0.000
lexA   0.212  0.383 -0.117  0.000  0.169  -0.087  0.000  0.125  0.000
ssb    0.000 -0.107 10.524  0.000  0.080   0.000  0.064  0.000  0.000
recF   0.104  0.000 -0.273  0.139  0.180   0.146  0.000  0.000  0.275
dinI   0.119  0.000  0.000  0.315  2.147   0.142 -0.068  0.135  0.000
umuDC  0.000 -0.189 -0.124  0.250  0.347   2.017 -0.067 -0.172  0.000
rpoD  -0.122  0.000 -0.102  0.000  0.000   0.000  3.068  0.365  0.217
rpoH   0.178 -0.183  0.000  0.000  0.000  -0.155  0.000 26.633  0.000
rpoS   0.000 -0.128  0.000  0.000  0.305   0.000  0.000  0.274  0.672
```

Set the parameters in NTW algorithm.

```
> restK=rep(ncol(X)-1, nrow(X))
> topD = round(0.6*nrow(X))
> topK = round(0.5*nrow(X))
> numP = round(0.25*nrow(X))
```

Input the gene association network `pred.net` from literature or some other method if possible. Here we randomly generate a network with `1` to indicate a connection between two genes, and `0` for no connection. `pred.net` must have the same dimensions as `A`.

```
> pred.net<-matrix(round(runif(nrow(X)*nrow(X), min=0, max=1)), nrow(X), nrow(X))
> pred.net

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
 [1,]    1    1    1    1    1    0    1    0    0
 [2,]    0    1    0    1    0    1    1    1    1
 [3,]    0    1    0    0    1    1    0    1    0
 [4,]    0    1    1    0    1    0    1    0    1
 [5,]    0    1    0    0    0    1    0    1    1
 [6,]    1    0    0    0    0    1    1    0    0
 [7,]    1    1    0    0    1    0    1    1    0
 [8,]    1    0    1    0    1    0    1    0    0
 [9,]    0    1    1    0    0    1    0    0    0
```

Estimate `A` and `P` without prior gene association information. Here the regression method is `sse`.

```
> result<-NTW(X, restK, topD, topK, P.known=NULL, cFlag="sse", pred.net = NULL, su
> result

$est.A
               recA        lexA           ssb         recF          dinI
recA   0.004977555 -0.02404257 -9.521820e-02  0.000000000  0.0048763857
lexA  -1.092623741 -0.44638945  0.000000e+00  0.733250538  0.1767204709
ssb    0.004977555 -0.02404257 -9.521820e-02  0.000000000  0.0048763857
recF   0.000000000 -0.01892404 -9.527017e-02 -0.002392418  0.0052141609
dinI   0.016062893 -0.36093297  0.000000e+00  0.474446606  0.1063749419
umuDC  0.004977555 -0.02404257 -9.521820e-02  0.000000000  0.0048763857
rpoD   0.008432960  0.00000000  8.875082e-05  0.000000000 -0.0008178425
rpoH   0.008432960  0.00000000  8.875082e-05  0.000000000 -0.0008178425
rpoS   0.071757103  0.05492974  1.953024e-03  0.000000000 -0.4799260016
              umuDC          rpoD        rpoH          rpoS
recA  -0.001514474  0.0022132822  0.00000000  0.0003041515
lexA  -0.084174579 -0.0221973113  0.00000000 -0.2416938635
ssb   -0.001514474  0.0022132822  0.00000000  0.0003041515
recF   0.000000000  0.0020923106  0.00000000  0.0000000000
dinI  -0.552254466 -0.0095769254  0.00000000 -0.2006185074
umuDC -0.001514474  0.0022132822  0.00000000  0.0003041515
rpoD  -0.002561085  0.0001450308 -0.03756027  0.0000000000
rpoH  -0.002561085  0.0001450308 -0.03756027  0.0000000000
rpoS   0.039802589 -0.0057605191  0.00000000  0.0243148293

$est.P
```

```
      p.recA p.lexA p.ssb p.recF p.dinI p.umuDC p.rpoD p.rpoH p.rpoS
recA       0      0     1      0      0       0      0      0      0
lexA       1      0     0      0      0       0      0      0      0
ssb        0      0     1      0      0       0      0      0      0
recF       0      0     1      0      0       0      0      0      0
dinI       0      0     0      0      0       1      0      0      0
umuDC      0      0     1      0      0       0      0      0      0
rpoD       0      0     0      0      0       0      0      1      0
rpoH       0      0     0      0      0       0      0      1      0
rpoS       0      0     0      0      1       0      0      0      0
```

Estimate `A` and `P` with prior gene association information. Here regression method is `geo`. The method to use the prior information is `forward`. `sup.drop` is set to `-1`, indicating `backward` approach is chosen.

```
> result<-NTW(X, restK, topD, topK, P.known=NULL, cFlag="sse", pred.net =pred.net,
> result

$est.A
              recA        lexA           ssb recF          dinI        umuDC
recA  0.000000000  0.00000000  0.000000e+00    0  0.0000000000  0.000000000
lexA  0.000000000  0.00000000  0.000000e+00    0  0.0000000000  0.000000000
ssb   0.000000000  0.00000000  0.000000e+00    0  0.0000000000  0.000000000
recF  0.000000000  0.00000000  0.000000e+00    0  0.0000000000  0.000000000
dinI  0.000000000  0.00000000  0.000000e+00    0  0.0000000000  0.000000000
umuDC 0.004977555 -0.02404257 -9.521820e-02    0  0.0048763857 -0.001514474
rpoD  0.000000000  0.00000000  0.000000e+00    0  0.0000000000  0.000000000
rpoH  0.008432960  0.00000000  8.875082e-05    0 -0.0008178425 -0.002561085
rpoS  0.071757103  0.05492974  1.953024e-03    0 -0.4799260016  0.039802589
               rpoD        rpoH         rpoS
recA   0.0000000000  0.00000000 0.0000000000
lexA   0.0000000000  0.00000000 0.0000000000
ssb    0.0000000000  0.00000000 0.0000000000
recF   0.0000000000  0.00000000 0.0000000000
dinI   0.0000000000  0.00000000 0.0000000000
umuDC  0.0022132822  0.00000000 0.0003041515
rpoD   0.0000000000  0.00000000 0.0000000000
rpoH   0.0001450308 -0.03756027 0.0000000000
rpoS  -0.0057605191  0.00000000 0.0243148293

$est.P
      p.recA p.lexA p.ssb p.recF p.dinI p.umuDC p.rpoD p.rpoH p.rpoS
recA       0      0     0      0      0       0      0      0      0
```

lexA	0	0	0	0	0	0	0	0	0
ssb	0	0	0	0	0	0	0	0	0
recF	0	0	0	0	0	0	0	0	0
dinI	0	0	0	0	0	0	0	0	0
umuDC	0	0	1	0	0	0	0	0	0
rpoD	0	0	0	0	0	0	0	0	0
rpoH	0	0	0	0	0	0	0	1	0
rpoS	0	0	0	0	1	0	0	0	0

Arguments of the function `NTW` here are:

- `X`, gene expression data, a matrix with genes as rows and perturbations as columns.
- `restK`, a vector (length equals to `nrow(A)`) with each element to indicate the number of non-zero regressors in the corresponding row of `A`.
- `topD`, a parameter for keeping the best (lowest error in the optimization function) `topD` combinations of non-zero regressors of a single row in `A`.
- `topK`, the number of possible targets of the perturbations, used for pre-estimate the perturbation targets matrix `P`.
- `P.known`, a known perturbation matrix with the same dimensions of `X`.
- `cFlag`, a flag to chose the regression methods: `geo`, `sse` and `ml`.
- `pred.net`, a matrix with the same dimensions of `A` for the prior gene association information. It can be specified only if `cMM.corrected` is `1`. Default is NULL.
- `sub.drop`, an indication to show the pattern for using the prior gene association information. `1` for `forward` pattern and `-1` for `backward` pattern.
- `numP`, a number set to limit the possibilities that one gene will be directly targeted by perturbations. That is at most `numP` perturbations can directly perturb one gene.
- `noiseLevel`, only used in `ml` method, to indicate the noise level in each perturbed experiment.

2.2 `AP.estimation.Srow` to estimate a single row of A and P

```
> IX<-P.preestimation(X, topK= round(2*nrow(X)))
> result.Srow<-AP.estimation.Srow(r=1,cMM.corrected = 1, pred.net,X, IX,topD, rest
> result.Srow
```

```
$P.index
[1] 3

$A.row
[1]  0.004641135 -0.021706411 -0.095177794  0.000000000  0.004436966
[6]  0.000000000  0.002237726  0.000000000  0.000000000
```

The arguments are similar to that in function `NTW` except `r` and `IX`. `r` indicates which row is estimated, while `IX` is a pre-estimated `P` according to gene expression data `X` based on the biological fact that each perturbation in one experiment have limited targets. The outputs are the coefficients of row `r` (`A.row`) and a vector to show which perturbations will target the gene of row `r`.

2.3 `A.estimation.Srow` to estimate a single row of A with P known

```
> P.known<-matrix(round(runif(nrow(X)*ncol(X), min=0, max=1)), nrow(X), ncol(X))
> result.Srow<-A.estimation.Srow(r=1,cMM.corrected = 1, pred.net, X, P.known, topD
> result.Srow

$A.row
[1] -0.448853934 -2.358075150 -0.035002767 -0.124809894 -0.187082967
[6]  0.000000000 -0.001844419  0.000000000  0.000000000
```

The arguments are similar to that in the function `AP.estimation.Srow` except `P.known`. `P.known` is the known `P`. The outputs are the coefficients of row `r` (`A.row`).

References

[1] T S Gardner, D di Bernardo, D Lorenz, and J J Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, Jul 2003.

[2] S Nelander, W Wang, B Nilsson, Q B She, C Pratilas, N Rosen, P Gennemark, and Sander C. Models from experiments: combinatorial drug perturbations of cancer cells. *Mol Syst Biol*, 4(216), 2008.