ℓ_2^2-learning of vector autoregressive processes

Wessel N. van Wieringen

Dept. of Epidemiology & Data Science, Amsterdam UMC
Dept. of Mathematics, Vrije Universiteit Amsterdam
Amsterdam, the Netherlands

IBC pre-conference course, Riga, 10.07.2022

Amsterdam UMC Universitair Medische Centra   VU UNIVERSITY AMSTERDAM

Auto-regression

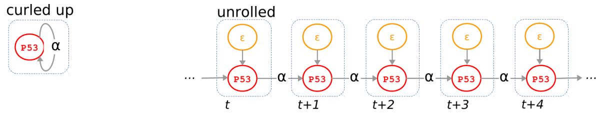
The AR(1) process:

$$Y_{t+1} \quad = \quad \alpha \, Y_t + \varepsilon_{t+1},$$

with $\alpha \in (-1, 1)$, $\varepsilon_t \sim_{\text{i.i.d.}} \mathcal{N}(0, \sigma_\varepsilon^2)$ for all t.

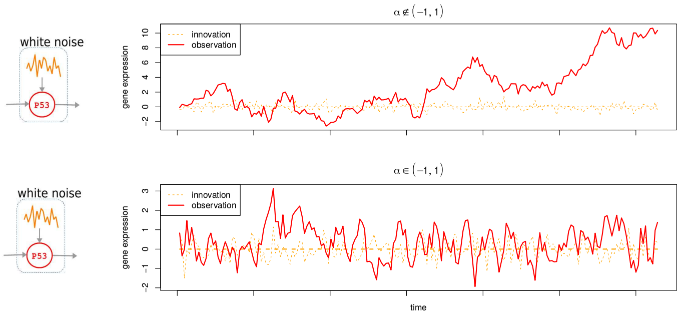
Nomenclature:
α : endogeneous operator, autoregression parameter,
ε_t : exogeneous signal, innovation.

The *auto-regulator* motif



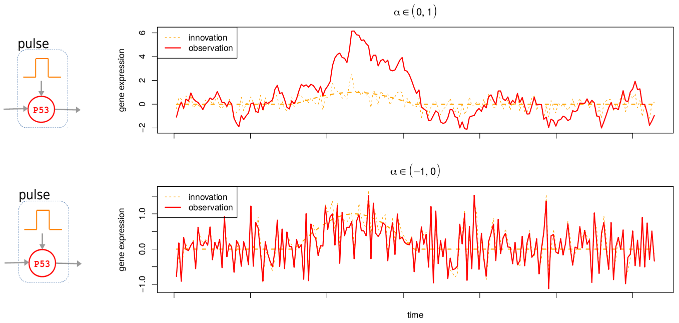
Auto-regression

Stationarity



Auto-regression

Pulse processing



Auto-regression

The AR(1) process can also be formulated as:

$$Y_{t+1} \,|\, Y_t \sim \mathcal{N}(\alpha Y_t, \sigma_\varepsilon^2).$$

This amounts to a 1st order Markov property:

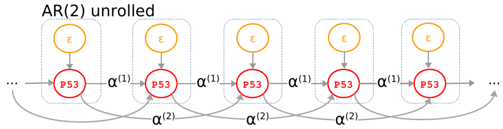
$$Y_{t+1} \,|\, Y_t, Y_{t-1}, Y_{t-2}, \ldots \sim Y_{t+1} \,|\, Y_t.$$

With the 2nd order Markov property, we would arrive at an AR(2) process:

$$Y_{t+1} \quad = \quad \alpha^{(1)} Y_t + \alpha^{(2)} Y_{t-1} + \varepsilon_{t+1},$$

with $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

AR(2) unrolled



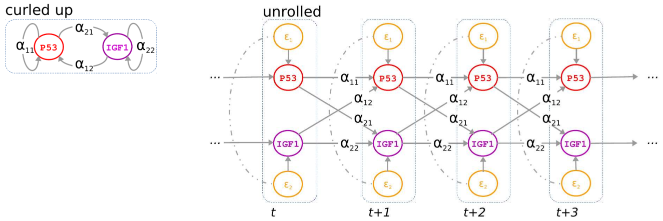
Auto-regression

A VAR(1) process:

$$\begin{aligned} Y_{1,t+1} \quad &= \quad \alpha_{11} \, Y_{1,t} + \alpha_{12} \, Y_{2,t} + \varepsilon_{1,t+1}. \\ Y_{2,t+1} \quad &= \quad \alpha_{21} \, Y_{1,t} + \alpha_{22} \, Y_{2,t} + \varepsilon_{2,t+1}. \end{aligned}$$

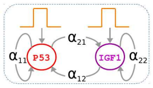
with a stationarity criterion on α's and $(\varepsilon_{1,t}, \varepsilon_{2,t})^\top \sim \mathcal{N}(\mathbf{0}_2, \mathbf{\Omega}_\varepsilon^{-1})$.

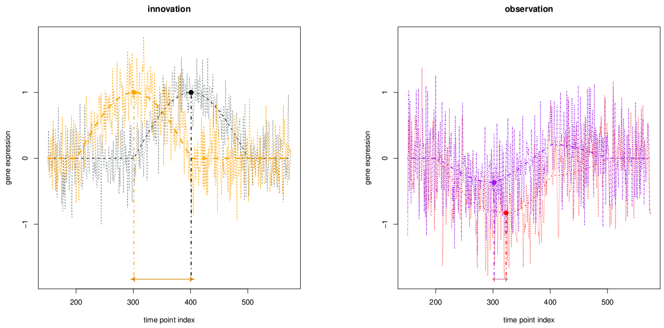
The *coupled auto-regulators* motif



Auto-regression

Coupled auto-regulators may achieve denoising and, necessary for the circadian clock, synchronization.



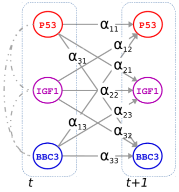


Auto-regression

A VAR(1) process:

$$\begin{aligned} Y_{1,t+1} &= \alpha_{11} Y_{1,t} + \alpha_{12} Y_{2,t} + \alpha_{13} Y_{3,t} + \varepsilon_{1,t+1}, \\ Y_{2,t+1} &= \alpha_{21} Y_{1,t} + \alpha_{22} Y_{2,t} + \alpha_{23} Y_{3,t} + \varepsilon_{2,t+1}, \\ Y_{3,t+1} &= \alpha_{31} Y_{1,t} + \alpha_{32} Y_{2,t} + \alpha_{33} Y_{3,t} + \varepsilon_{3,t+1}. \end{aligned}$$

unrolled

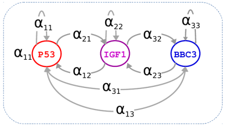


Or, in matrix notation

$$\mathbf{Y}_{t+1} = \mathbf{A}\mathbf{Y}_t + \boldsymbol{\varepsilon}_{t+1}$$

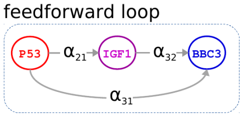
with $|ev_j(\mathbf{A})| < 1$ and $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}_3, \boldsymbol{\Omega}_\varepsilon^{-1})$.

curled up

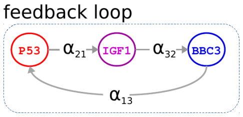


Auto-regression

feedforward loop

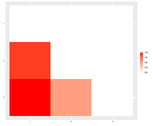


feedback loop

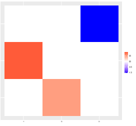


$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ \alpha_{21} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & 0 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 0 & 0 & \alpha_{13} \\ \alpha_{21} & 0 & 0 \\ 0 & \alpha_{32} & 0 \end{pmatrix}$$

A =



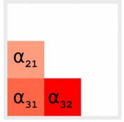
A =

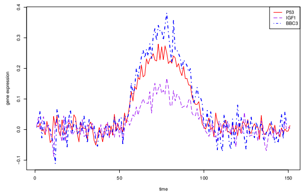


Auto-regression

Pulse processing by the *coherent* feedforward motif

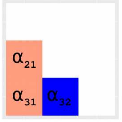
A =

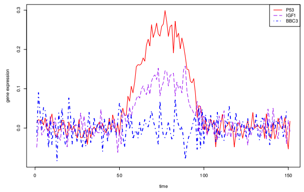




Pulse processing by the *incoherent* feedforward motif

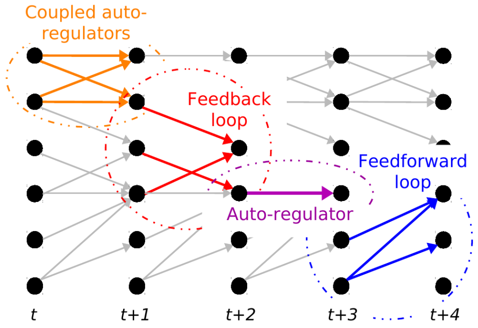
A =





Auto-regression

Systems biology: how will a combination of motifs play out?



Auto-regression

What can we say about the (marginal) dependence among the variates?

A closer look at the recursive relation:

$$\begin{aligned} \mathbf{Y}_{t+1} &= \mathbf{A}\mathbf{Y}_t + \boldsymbol{\varepsilon}_{t+1} \\ &= \mathbf{A}^2\mathbf{Y}_{t-1} + \mathbf{A}\boldsymbol{\varepsilon}_t + \boldsymbol{\varepsilon}_{t+1} \\ &= \mathbf{A}^3\mathbf{Y}_{t-2} + \mathbf{A}^2\boldsymbol{\varepsilon}_{t-1} + \mathbf{A}\boldsymbol{\varepsilon}_t + \boldsymbol{\varepsilon}_{t+1} \\ &= \ldots \end{aligned}$$

Then, e.g.

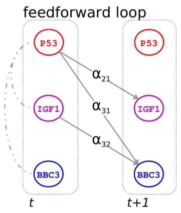
$$\mathrm{Cov}(\mathbf{Y}_{t+2}, \mathbf{Y}_t) = \mathrm{Cov}(\mathbf{A}^2\mathbf{Y}_t + \mathbf{A}\boldsymbol{\varepsilon}_{t+1} + \boldsymbol{\varepsilon}_{t+2}, \mathbf{Y}_t) = \mathbf{A}^2\boldsymbol{\Sigma}_y,$$

where $\boldsymbol{\Sigma}_y$ denotes the process variance and satisfies $\boldsymbol{\Sigma}_y = \mathbf{A}\boldsymbol{\Sigma}_y\mathbf{A}^\top + \boldsymbol{\Sigma}_\varepsilon$.

Nomenclature: *auto-correlation* vs. *cross-correlation*,
$\mathrm{Cov}(Y_{j,t+2}, Y_{j,t})$ vs. $\mathrm{Cov}(Y_{j,t+2}, Y_{j',t})$.

Auto-regression

The *time-series chain graph* (TSCG) harbors the conditional (in)dependencies.



Temporal conditional independencies:

$$Y_{j',t} \perp\!\!\!\perp Y_{j'',t+1} \,|\, \text{other } Y_{j''',t}\text{'s.}$$

Parametric criterion:

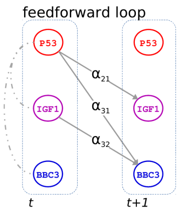
$$[(\mathbf{A})^{-1}]_{j',j''} = 0 \quad \Longleftrightarrow \quad Y_{j'',t} \perp\!\!\!\perp Y_{j',t+1} \,|\, \text{other } Y_{j''',t}\text{'s.}$$

Examples:

$Y_{\mathrm{P53},t+1} \perp\!\!\!\perp Y_{\mathrm{BBC3},t} \,|\, Y_{\mathrm{IGF1},t}$ (*no* edge).
$Y_{\mathrm{P53},t} \not\perp\!\!\!\perp Y_{\mathrm{IGF1},t+1} \,|\, Y_{\mathrm{BBC3},t}$ (*directed* edge).

Auto-regression

The *time-series chain graph* (TSCG) harbors the conditional (in)dependencies.



Contemporaneous cond. independencies:

$$Y_{j',t+1} \perp\!\!\!\perp Y_{j'',t+1} \,|\, \text{other } Y_{j''',t+1}\text{'s, all } Y_{j,t}\text{'s.}$$

Parametric criterion:

$$(\mathbf{\Omega}_{\varepsilon})_{j',j''} = 0 \quad \Longleftrightarrow \quad Y_{j',t+1} \perp\!\!\!\perp Y_{j'',t+1} \,|\, \text{other } Y_{j''',t+1}\text{'s, all } Y_{j,t}\text{'s.}$$

Examples:

$Y_{\mathrm{IGF1},t+1} \perp\!\!\!\perp Y_{\mathrm{BBC3},t+1} \,|\, Y_{\mathrm{TP53},t+1}, Y_{\mathrm{IGF1},t}, Y_{\mathrm{BBC3},t}, Y_{\mathrm{TP53},t}$ (*no* edge).
$Y_{\mathrm{P53},t+1} \not\perp\!\!\!\perp Y_{\mathrm{IGF1},t+1} \,|\, Y_{\mathrm{BBC3},t+1}, Y_{\mathrm{IGF1},t}, Y_{\mathrm{BBC3},t}, Y_{\mathrm{TP53},t}$ (*undirected* edge).

Auto-regression

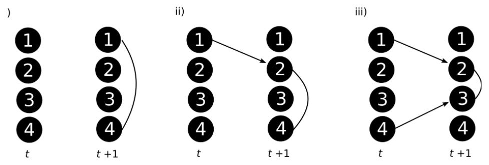
Global conditional (in)dependencies:

$$Y_{j',*} \perp\!\!\!\perp Y_{j'',*} \,|\, \text{other } Y_{j''',*}\text{'s}$$

Wermuth condition for global conditional independency:

- ○ unconnected nodes do not exert influence on the same node.

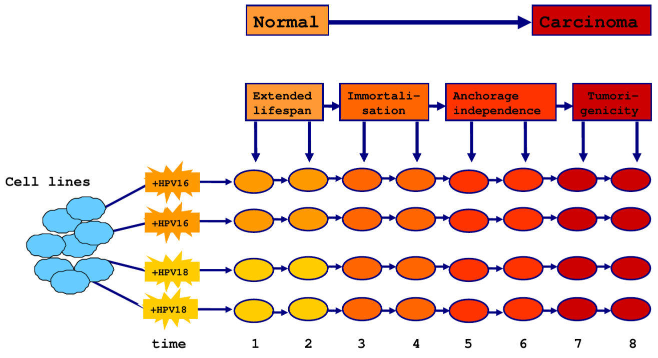
Consider a four-gene VAR(1) process. Then, global conditional independence of genes 1 and 2 forbids motifs like:



Goals

- ○ learn VAR(1) model,
- ○ reconstruct TSCG,
- ○ exploit VAR(1) model and TSCG, and
- ○ explore extensions.

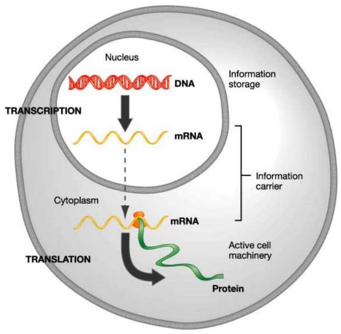
Data (Section 1.2)



Data (Section 1.2)

The *central dogma of molecular biology* discerns three steps in the process that leads from the information encoded in DNA to the proteins in the cell:

- ○ Replication
- ○ Transcription
- ○ Translation



Gene expression data is proportional to # mRNA molecules present in the cell.

Data (Section 1.2)

Load gene expression data related to P53 signalling pathway.

```
# load data
data(hpvP53)

# reformat data
Y <- longitudinal2array(t(exprs(hpvP53rna)))

# inquire design
pData(hpvP53rna)
fData(hpvP53rna)
```

The Y is an array-class object.

```
# interrogate dimension
> dim(Y)
```

Its dimensions: # genes × # time points × # cell lines

Data (Section 1.3)

Visualize data of single gene.

```
# plot time-courses of a single gene
plotVAR1data(Y[5, , , drop=FALSE])
```

Visualize data of single cell line.

```
# K-means clustering of the genes
cellLine <- 1
kClust   <- kmeans(Y[, , cellLine],
                   centers=7, nstart=100)$cluster

# heatmap of reshuffled data
edgeHeat(Y[unlist(lapply(1:max(kClust),
                function(id, clusters){
                which(clusters==id) },
                kClust)), , cellLine])
```

The VAR(1) model (Section 1.3)

Q: the study comprises how many genes, cell lines, and time points?

Q: the BBC3 gene lies on which chromosome?

Q: slicing, e.g., Y[, , 1], an array objects yields ...?

Q: are there genes that are upregulated over time?

Q: are there genes that exhibit different behaviour between cell lines?

Q: are there groups of genes that exhibit similar behaviour over time?

Data (Section 1.3, answer)

Experimental design

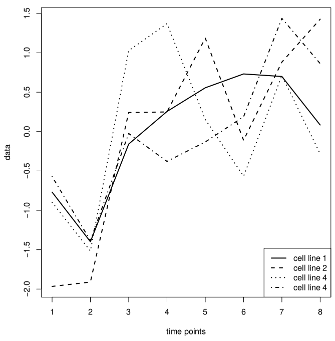
```
    cellLine  hpv  time
1-1        1   16     1
1-2        2   16     1
1-3        3   18     1
1-4        4   18     1
2-1        1   16     2
2-2        2   16     2
...
```

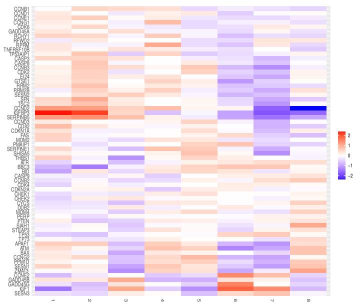
Annotation information

```
                 gene  entrezID          probe  chr
APAF1           APAF1       317   A_23_P36611   12
ATM               ATM       472   A_23_P35916   11
ATR               ATR       545  A_23_P136058    3
BAX               BAX       581  A_23_P208706   19
BBC3             BBC3     27113  A_24_P305312   19
BID               BID       637  A_23_P154929   22
...
```

Data (Section 1.3, answer)

First impression of the data





Estimation

Experimental design

- ○ $j = 1, \ldots, p$ genes,
- ○ $t = 1, \ldots, \mathcal{T}$ time points,
- ○ $i = 1, \ldots, n$ cell lines.

Here, $p = 64$, $\mathcal{T} = 8$, and $n = 4$.

Data

- ○ $Y_{j,t,i}$ is expression of gene j at time point t of cell line i.

Assumptions

- ○ cell lines are independent,
- ○ cell line data $\sim$ same VAR(1) model.

Estimation

The likelihood of cell line i:

$$P(\mathbf{Y}_{*,\mathcal{T},i}, \mathbf{Y}_{*,\mathcal{T}-1,i}, \ldots, \mathbf{Y}_{*,1,i}) \quad = \quad P(\mathbf{Y}_{*,1,i}) \prod_{t=1}^{\mathcal{T}-1} P(\mathbf{Y}_{*,t+1,i} \,|\, \mathbf{Y}_{*,t,i}),$$

due to the 1^{st} order Markov assumption.

Furthermore, due to the VAR(1) model assumption,

$$P(\mathbf{Y}_{*,t+1,i} \,|\, \mathbf{Y}_{*,t,i}) \quad = \quad \phi_{\mathbf{AY}_{*,t,i}, \mathbf{\Omega}_{\varepsilon}^{-1}}(\mathbf{Y}_{*,t+1,i}).$$

Together with the independence assumption, this yields the likelihood:

$$\propto \prod_{i=1}^{n} \prod_{t=1}^{\mathcal{T}-1} (2\pi)^{-p/2} |\mathbf{\Omega}_{\varepsilon}|^{1/2} \exp[-\tfrac{1}{2}(\mathbf{Y}_{*,t+1,i} - \mathbf{AY}_{*,t,i})^{\top} \mathbf{\Omega}_{\varepsilon} (\mathbf{Y}_{*,t+1,i} - \mathbf{AY}_{*,t,i})].$$

Q: why can we drop the 'start probability'?

Estimation

The ridge ML estimator of the VAR(1) model maximizes

$$\text{loglikelihood} - \lambda_a \|\text{vec}(\mathbf{A})\|_2^2 - \lambda_{\omega} \|\mathbf{\Omega}_{\varepsilon}\|_2^2.$$

Targets $\mathbf{A}_0$ and $\mathbf{\Omega}_0$ could be included.

Analytic expressions of estimators are available:

$$\begin{aligned} \widehat{\mathbf{A}}(\lambda_a) &= [\lambda_a \mathbf{I}_{p^2 p^2} + \widehat{\mathbf{\Gamma}}(0) \otimes \mathbf{\Omega}_{\varepsilon}]^{-1} \text{vec}[\mathbf{\Omega}_{\varepsilon} \widehat{\mathbf{\Gamma}}(-1)], \\ \widehat{\mathbf{\Omega}}_{\varepsilon}(\lambda_{\omega}) &= [\tfrac{1}{2}\mathbf{S}_{\varepsilon} + (\lambda_{\omega} \mathbf{I}_{pp} + \tfrac{1}{4}\mathbf{S}_{\varepsilon}^2)^{1/2}]^{-1}, \end{aligned}$$

with sufficient statistics $\widehat{\mathbf{\Gamma}}(0)$ and $\widehat{\mathbf{\Gamma}}(-1)$ and

$$\mathbf{S}_{\varepsilon} \quad = \quad n^{-1}(\mathcal{T}-1)^{-1} \sum_{i=1}^{n} \sum_{t=1}^{\mathcal{T}-1} (\mathbf{Y}_{*,t+1,i} - \mathbf{AY}_{*,t,i})(\mathbf{Y}_{*,t+1,i} - \mathbf{AY}_{*,t,i})^{\top}.$$

Alternate between updating of $\widehat{\mathbf{A}}(\lambda_a)$ and $\widehat{\mathbf{\Omega}}_{\varepsilon}(\lambda_{\omega})$.

Estimation

The ridge ML estimators of $\mathbf{A}$ and $\mathbf{\Omega}_{\varepsilon}$

- are consistent in the $n \to \infty$-limits.
  If $\lambda_a, \lambda_{\omega} \xrightarrow{P} 0$, then $\widehat{\mathbf{A}}(\lambda_a) \xrightarrow{P} \mathbf{A}$ and $\widehat{\mathbf{\Omega}}_{\varepsilon}(\lambda_{\omega}) \xrightarrow{P} \mathbf{\Omega}_{\varepsilon}$.
- have Bayesian counterparts.
  E.g, if $\text{vec}(\mathbf{A}) \,|\, \mathbf{\Omega}_{\varepsilon} \sim \mathcal{N}(\mathbf{0}_{p^2}, \lambda_a^{-1} \mathbf{I}_{p^2 p^2})$, then $\mathbb{E}(\mathbf{A} \,|\, \mathbf{Y}, \mathbf{\Omega}_{\varepsilon}) = \widehat{\mathbf{A}}(\lambda_a)$.
- can be obtained by data augmentation.

Estimation

The auto-regression parameter $\mathbf{A}$ can also be estimated by means of ridge penalized sum-of-squares minimization:

$$\sum_{i=1}^{n} \sum_{t=1}^{\mathcal{T}-1} \|\mathbf{Y}_{*,t+1,i} - \mathbf{AY}_{*,t,i}\|_2^2 - \lambda_a \|\text{vec}(\mathbf{A})\|_2^2.$$

A target $\mathbf{A}_0$ could be included.

An analytic expressions of the estimator is available:

$$\begin{aligned} \widehat{\mathbf{A}}^{(\text{ridge ss})}(\lambda_a) &= [\lambda_a \mathbf{I}_{p^2 p^2} + \widehat{\mathbf{\Gamma}}(0) \otimes \mathbf{\mathit{I}}_{pp}]^{-1} \text{vec}[\widehat{\mathbf{\Gamma}}(-1)], \\ & \quad \text{vs.} \\ \widehat{\mathbf{A}}^{(\text{ridge ml})}(\lambda_a) &= [\lambda_a \mathbf{I}_{p^2 p^2} + \widehat{\mathbf{\Gamma}}(0) \otimes \mathbf{\Omega}_{\varepsilon}]^{-1} \text{vec}[\mathbf{\Omega}_{\varepsilon} \widehat{\mathbf{\Gamma}}(-1)]. \end{aligned}$$

Q: what is the advantage of the penalized sum-of-squares estimator?

Estimation (Section 1.5)

Code of ridge ML estimation

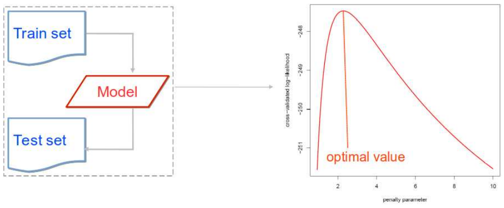
```
# fit the model
VAR1hat <- ridgeVAR1(Y=Y,
                     lambdaA=optLambdas[1],
                     lambdaP=optLambdas[2],
                     fitA    ="ml",
                     targetA=matrix(0, dim(Y)[1], dim(Y)[1]),
                     targetP=matrix(0, dim(Y)[1], dim(Y)[1]),
                     targetPtype="none")

# extract parameter estimates
Ahat  <- VAR1hat$A
Phat  <- VAR1hat$P

# visually inspect the estimates, e.g.
edgeHeat(Ahat, main="ridge␣estimate␣of␣A")
```

Estimation (Section 1.4)

Cross-validation chooses λ to optimize prediction.



- Leave-one-out cross-validation leaves out time points one-by-one.
- Model learnt on training set.
- Fitted model applied to the left-out time point and evaluate the error.
- Average of these errors estimates the 'full data' predictor's error rate.

Estimation (Section 1.4)

R-code of cross-validation

```
# find optimal penalty parameters
optLambdas <- optPenaltyVAR1(Y,
                        lambdaMin =c(0.01, 0.00001),
                        lambdaMax =c(1000, 1),
                        lambdaInit=c(100 , 0.1),
                        optimizer ="nlm", ...)
```

Estimation (Section 1.4)

Investigate optimality of λ choice

```
# define grid for both penalty parameters
lambdaAgrid <- seq(300,    550, length.out=20)
lambdaPgrid <- seq(0.001, 0.01, length.out=20)

# evaluate LOOCV log-likelihood over the grid
LOOCVres <- loglikLOOCVcontourVAR1(lambdaAgrid,
                                   lambdaPgrid,
                                   Y, ...)
```

Plot the LOOCV log-likelihood contour with optimal penalty parameters:

```
# plot LOOCV log-likelihood somewhat nicer
contour(lambdaAgrid, lambdaPgrid, LOOCVres$llLOOCV, ...)

# add optimal penalty parameters
points(optLambdas[1], optLambdas[2], ...)
```

The VAR(1) model (Section 1.4 & 1.5)

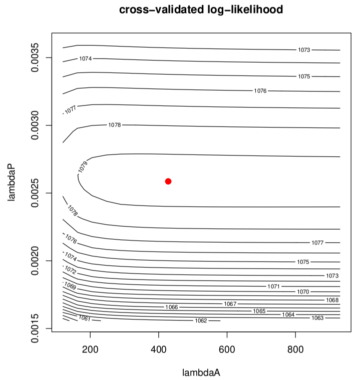
Q: tune the penalty parameters.

Q: estimate the VAR(1) model parameters.

Q: investigate the optimality of the chosen penalty parameters.

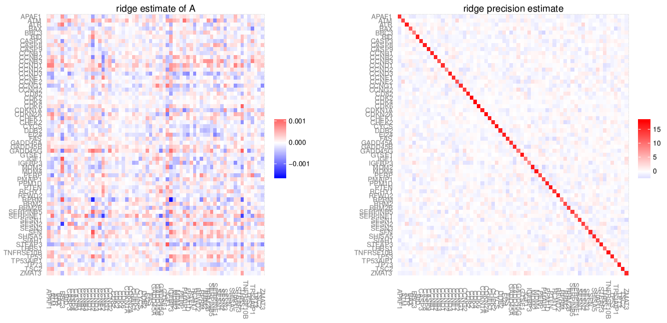
Estimation (Section 1.4, answer)

LOOCV: $\lambda_a^{(\text{opt})} = 427.95465$ and $\lambda_\omega^{(\text{opt})} = 0.00259$.



Estimation (Section 1.5, answer)

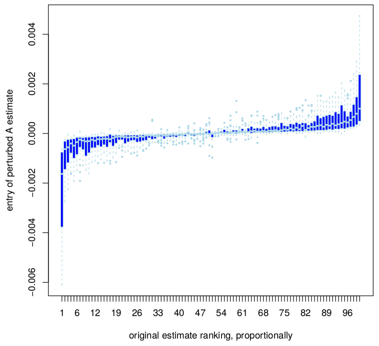
Heatmap of ridge ML estimates of $\mathbf{A}$ (left) and $\mathbf{\Omega}_\varepsilon$ (right).



Estimation (Section 1.6)

Assess stability of $\widehat{\mathbf{A}}(\lambda_a)$.

- Perturb data: remove a design point.
- Re-do estimation, including LOOCV.
- Store elements of $\widehat{\mathbf{A}}_{-(t,i)}(\lambda_a)$.
- Repeat for all design points.
- Plot $\widehat{\mathbf{A}}_{-(t,i)}(\lambda_a)$'s vs $\widehat{\mathbf{A}}(\lambda_a)$.



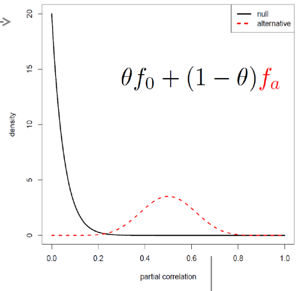
Estimation (Section 1.7)

Support determination

standardized Ω_ε estimate

	1	2	3	4	5
1	1.00	0.51	0.42	-0.03	0.34
2	*	1.00	0.40	0.11	-0.09
3	*	*	1.00	-0.57	0.04
4	*	*	*	1.00	0.07
5	*	*	*	*	1.00

model *as mixture*



$$P\big[(1,2) \notin \mathcal{E}_{\mathcal{G}} \,|\, (\widehat{\mathbf{\Omega}})_{1,2}\big]$$

cut-off

inferred Ω_ε support

	1	2	3	4	5
1	1	1	1	0	1
2	*	1	1	0	0
3	*	*	1	1	0
4	*	*	*	1	0
5	*	*	*	*	1

Estimation (Section 1.7)

R-code of sparsification

```
# support determination of A
zerosA <- sparsifyVAR1(A          =Ahat,
                       SigmaE     =symm(solve(Phat)),
                       threshold  ="top",
                       top        =25,
                       statistics=FALSE,
                       verbose    =FALSE)$zeros

# support determination of precision matrix
zerosP <- sparsify(P         =Phat,
                   threshold="top",
                   top       =10,
                   output    ="light",
                   verbose   =FALSE)$zeros
```

Estimation (Section 1.8)

We re-estimate the parameters with known support.

This undoes some of the shrinkage.

The ridge ML estimates with known support maximize

$$\arg \max_{\substack{\{\mathbf{A} \,:\, \mathbf{C}_a \mathrm{vec}(\mathbf{A})=\mathbf{0}_{q_a}\} \\ \{\mathbf{\Omega}_\varepsilon \,:\, \mathbf{C}_\omega \mathrm{vec}(\mathbf{\Omega}_\varepsilon)=\mathbf{0}_{q_\omega}\}}} \text{loglikelihood} - \lambda_a \|\mathrm{vec}(\mathbf{A})\|_2^2 - \lambda_\omega \|\mathbf{\Omega}_\varepsilon\|_2^2.$$

The penalty parameters λ_a and λ_ω are re-tuned.

Estimation

R-code for calibration, re-tuning of the λ's.

```
# format precision support
supportP <- support4ridgeP(zeros =zerosP,
                           nNodes=nrow(Y))

# optimal penalty parameter determination
optLambdas <- optPenaltyVAR1(Y,
                   lambdaMin   =c(10^(-5), 10^(-5)),
                   lambdaMax   =c(10,      0.1),
                   lambdaInit  =c(5,       0.01),
                   zerosA      =zerosA,
                   zerosP      =zerosP,
                   cliquesP    =supportP$cliques,
                   separatorsP=supportP$separators,
                   zerosAfit   ="sparse")
```

Estimation

R-code for calibration, re-estimation.

```
# re-fit the model
VAR1hat <- ridgeVAR1(Y          =Y,
                     lambdaA    =optLambdas[1],
                     lambdaP    =optLambdas[2],
                     zerosA     =zerosA,
                     cliquesP   =supportP$cliques,
                     separatorsP=supportP$separators,
                     zerosP     =zerosP,
                     zerosAfit  ="sparse")

# extract parameter estimates
Ahat <- VAR1hat$A
Phat <- VAR1hat$P

# heatmap of support of A
edgeHeat(adjacentMat(Ahat), ...)
```

Estimation (Section 1.8)

R-code to plot the reconstructed time-series chain graph.

```
# time-series chain graph
graphVAR1(Ahat,
          Phat,
          nNames=rownames(Ahat),
          type  ="TSCG",
          ...)
```

R-code to plot the (global) conditional independence graph.

```
# determine adjacency matrix
adjMatCIG <- CIGofVAR1(Ahat, Phat)

# plot global conditional independence graph
graphVAR1(Ahat,
          Phat,
          nNames=rownames(Ahat),
          type  ="globalPC")
```

The VAR(1) model (Section 1.7 & 1.8)

Q: sparsify the estimated VAR(1) model parameters.

Q: re-tune the penalty parameters with inferred support.

Q: re-estimate the VAR(1) model parameters with inferred support.

Q: visualize model parameter.

Q: draw the time-series chain graph.

The VAR(1) model (Section 1.7, answer)

For the autoregression parameter we obtain

```
Step 1... determine cutoff point
Step 2... estimate parameters of null distribution and eta0
Step 3... compute p-values and estimate empirical PDF/CDF
Step 4... compute q-values and local fdr
Step 5... prepare for plotting

-> Retained elements:  134
-> Corresponding to 3.27 % of possible elements
```

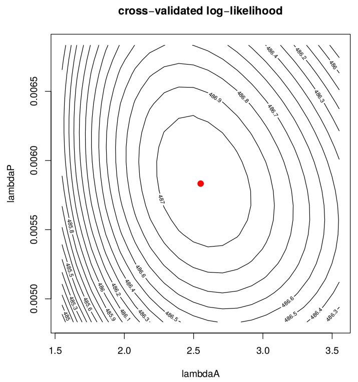
while for the error precision matrix

```
...

- Retained elements:  14
- Corresponding to 0.69 % of possible edges
```

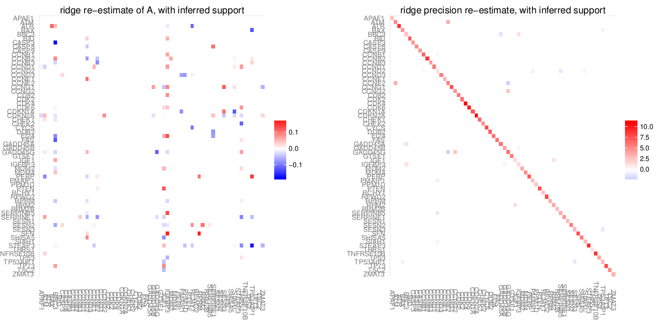
Estimation (Section 1.8, answer)

LOOCV: $\lambda_a^{(\mathrm{opt})} = 2.55082$ and $\lambda_\omega^{(\mathrm{opt})} = 0.00583$, known support.

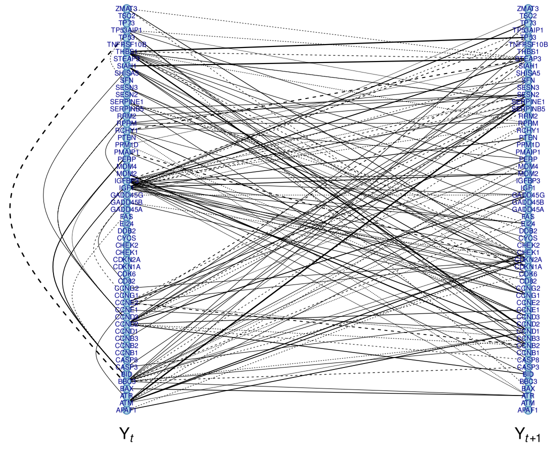


Estimation (Section 1.8, answer)

Heatmap of ridge ML estimates of $\mathbf{A}$ (left) and $\mathbf{\Omega}_\varepsilon$ (right), known support.



Estimation (Section 1.8, answer)



Fit (Section 1.9)

The fit is

$$\widehat{\mathbf{Y}}_{*,t+1,i} \quad = \quad \widehat{\mathbf{A}}(\lambda_a)\mathbf{Y}_{*,t,i}.$$

R-code to obtain the fit.

```
# calculate fits and fit-vs-observation correlations
Yhat <- array(dim=dim(Y))
for (i in 1:dim(Y)[3]){
  Yhat[, -1, i] <- Ahat %*% Y[, -dim(Y)[2], i]
}
```

Ideally, $\widehat{\mathbf{Y}}_{*,t,i}$ is close (in some sense) to $\mathbf{Y}_{*,t,i}$.

Fit (Section 1.9)

R-code to compare $\widehat{\mathbf{Y}}_{j,*,i}$ and $\mathbf{Y}_{j,*,i}$ visually.

```
# specify molecular entity of interest
entityID <- 1

# format data for plotting
label <- paste("Cell", sort(rep(1:dim(Y)[3], dim(Y)[2]-1)))
...
```

R-code to compare Cor($\widehat{\mathbf{Y}}_{j,*,i}$, $\mathbf{Y}_{j,*,i}$).

```
# calculate fits and fit-vs-obs correlations
corFit <- numeric()
for (j in 1:dim(Y)[1]){
  ...
}

# histogram of the fit-vs-obs correlations
hist(corFit, ...)
```

Fit (Section 1.9)

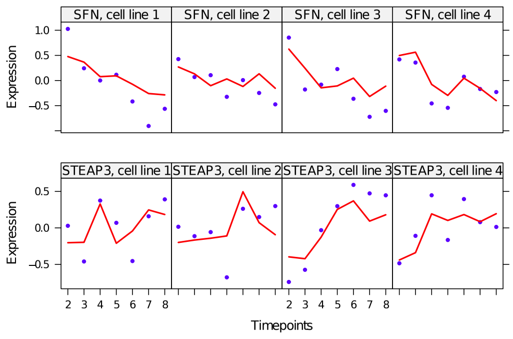
Q: calculate the fit and compare it to the observations.

Q: why is gene 1 a ‘flatliner’?

Q: how does it compare to the other histograms
(correlation with previous observation and overall mean)?

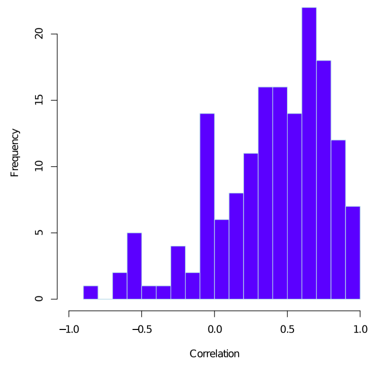
Fit (Section 1.9, answer)

Scatter plots of $\widehat{\mathbf{Y}}_{j,*,i}$ and $\mathbf{Y}_{j,*,i}$.



Fit (Section 1.9, answer)

Histogram of Cor($\widehat{\mathbf{Y}}_{j,*,i}$, $\mathbf{Y}_{j,*,i}$).

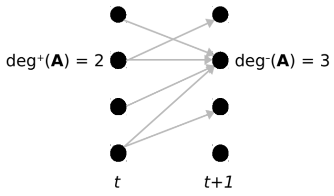


Exploitation (Section 1.10.1)

Calculate node-wise
network statistics.

Take into account
directionality.

For instance, in- and
out-degree of a node.



Code to calculate these statistics.

```
# calculate node-wise network stats
nodeStats <- nodeStatsVAR1(Ahat, Phat, as.table=TRUE)
```

Network statistics are based on both the TSCG and the global CIG graph!

Exploitation (Section 1.10.2)

Mutual information quantifies the information gain on one random variable by observing another. Formally,

$$\begin{aligned}\mathcal{I}(\mathbf{Y}_{*,t+\tau,i}, Y_{j,t,i} \,|\, \mathbf{Y}_{*,t-1,i}) &= \mathcal{H}(\mathbf{Y}_{*,t+\tau,i} \,|\, \mathbf{Y}_{*,t-1,i}) \\ &\quad -\mathcal{H}(\mathbf{Y}_{*,t+\tau,i} \,|\, Y_{j,t,i}, \mathbf{Y}_{*,t-1,i}),\end{aligned}$$

where $\mathcal{H}$ denotes the entropy.

Here the entropy is e.g.

$$\mathcal{H}(\mathbf{Y}_{*,t+\tau,i} \,|\, \mathbf{Y}_{*,t-1,i}) = \log\left\{\det\left[\mathrm{Var}(\mathbf{Y}_{*,t+\tau,i} \,|\, \mathbf{Y}_{*,t-1,i})\right]\right\}.$$

Code to plot the (global) conditional independence graph.

```
# evaluate mutual informations with specified lag
MIs <- mutualInfoVAR1(Ahat, solve(Phat), lag=1)
```

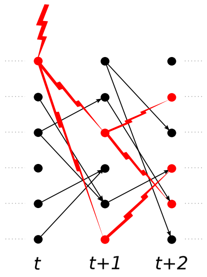
Exploitation (Section 1.10.3)

Impulse response analysis

Assess the change in variates at future time points due to an exogeneous change at the current time:

$$\frac{\partial \mathbf{Y}_{*,t+\tau}}{\partial \varepsilon_{*,t}} \;=\; \mathbf{A}^{\tau}.$$

Facilitates prediction of a knockout effect.



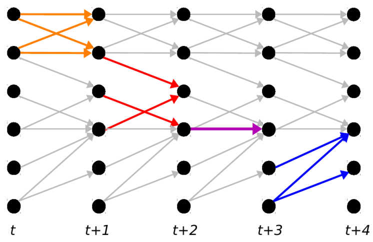
R-code

```
# evaluate impulse response with specified lag
IRs <- impulseResponseVAR1(Ahat, lag=1)
```

Exploitation (Section 1.10.4)

R-code to generate motif frequency table.

```
# motif detection
motifs <- motifStatsVAR1(Ahat)
```



Auto-regulator	: 11
Coupled auto-regulators	: 3
Feedback loop	: 7
Feedforward loop	: 2
...	: ...

Exploitation (Section 1.10.5)

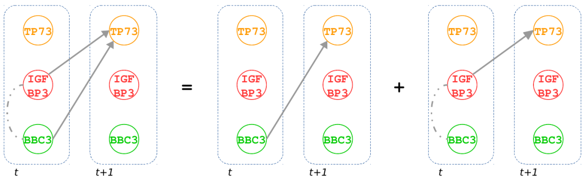
Covariance decomposition

Given TSCG, decompose conditional covariance in terms of paths as:

$$\mathrm{Cov}(Y_{j',t,i}, Y_{j'',t+\tau,i} \,|\, \mathbf{Y}_{*,t-1,i}) \;=\; (\boldsymbol{\Sigma}_{\varepsilon}\mathbf{A}^{\tau})_{j',j''} \;=\; \sum\nolimits_{j'''=1}^{p} (\boldsymbol{\Sigma}_{\varepsilon})_{j',j'''}\,(\mathbf{A}^{\tau})_{j''',j''}.$$

E.g.

$$\mathrm{Cov}(\mathrm{BBC3}_t, \mathrm{TP73}_t \,|\, \ldots) \;=\; -0.002483485 - 0.0006845158$$



Fit (Section 1.10)

Q: Calculate the node-statistics.
On the basis of these statistics can you discern types of genes?

Q: The GADD45G gene is a potential tumor suppressor.
If activated, it may form a novel treatment.
Which other gene's perturbation at time t stimulates GADD45G most a $t+1$? And at $t+2$?

Q: What is the most prevalent motif?
Is there an *incoherent* feedforward loop present in the TSCG?

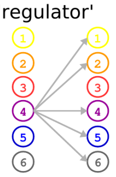
Exploitation (Section 1.10.1, answer)

Node-wise network statistics

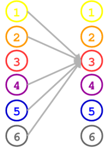
	$\deg^{-}(\mathbf{A})$	$\deg^{+}(\mathbf{A})$	between.	close.	eigen centr.
BBC3	0	17	17	0.00192	1.00000
CCND2	0	12	18	0.00187	0.68783
IGF1	1	14	0	0.00191	0.97635
IGFBP3	0	16	7	0.00190	0.87513
THBS1	0	11	0	0.00188	0.87717
CCNG1	6	0	0	0.00177	0.25154
CDKN2A	12	0	0	0.00181	0.49508
SERPINE1	8	4	0	0.00185	0.70869
SESN2	8	0	0	0.00180	0.26759
STEAP3	9	0	0	0.00179	0.36285

upper five ≈ 'regulators'
lower five ≈ 'regulatees'

'regulator'



'regulatee'



Exploitation (Sections 1.10.2&3, answer)

Mutual information results

- ∘ 'regulator' ($\deg^{+} = 17$) : mutual info of $\mathrm{BBC3}_t$ on $t+1 = 0.05605$,
- ∘ 'regulatee' ($\deg^{-} = 12$) : mutual info of $\mathrm{CDKN2A}_t$ on $t+1 = 0.00000$.

Result of impulse response analyses, e.g.:

- ∘ 'regulator' ($\deg^{+} = 17$) : effect of $\mathrm{BBC3}_t$ on all $t+1 = 0.01497^{*}$,
- ∘ 'regulatee' ($\deg^{-} = 12$) : effect of $\mathrm{CDKN2A}_t$ on all $t+1 = 0.00000^{*}$.

* the average (over all genes) of their absolute changes due to a change in, e.g., BBC3 at the previous time point.

Exploitation (Sections 1.10.3, answer)

Study to impulse response of GADD45G due to a change in all others

```
id <- which(impulseResponseVAR1(Ahat, 1)[34,] != 0)
impulseResponseVAR1(Ahat, 1)[34, id]
```

At $t+1$:

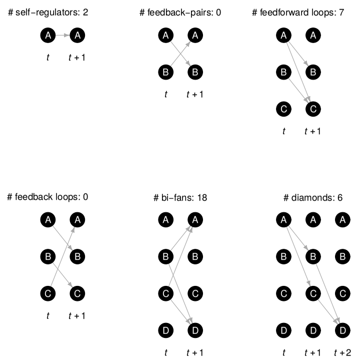
```
    ATM     BBC3    CCND2  GADD45G   IGFBP3    THBS1
 0.0165  -0.0314   0.0563  -0.1136  -0.0278   0.0714
```

At $t+2$:

```
    ATM     BBC3    CCND2  GADD45G   IGFBP3    THBS1
-0.0019   0.0036  -0.0064   0.0129   0.0032  -0.0081
```

Exploitation (Section 1.10.4, answer)

The motif frequency table is plotted ...



Exploitation (Section 1.10.4, answer)

... but also printed on screen:

```
motif summary table:
                          motif frequency
 -> selfregulator     :           2
 -> feedback pair     :           0
 -> feedforward loop  :           7
 -> feedback loop     :           0
 -> bifan             :          18
 -> diamond           :           6
```

A list with details of all motifs is also provided. E.g.,

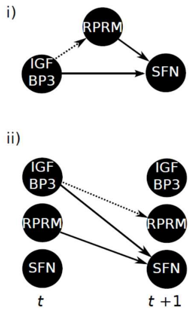
```
$feedforwardloops[[1]]
       t t+1 sign
path1  2  50    1
path2  2  25    1
path3 25  50    0
```

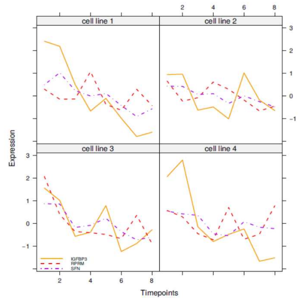
Recall: a motif's (in)coherence is determined by its signs.

Exploitation (Section 1.10.4, answer)

Motif analysis

E.g. feedforward motif





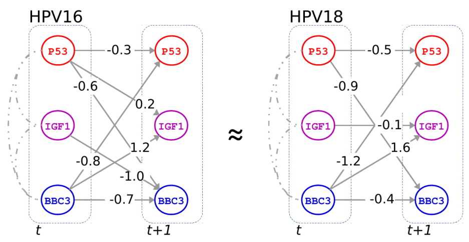
Extensions

Multiple VAR(1) models (Section 2)

Investigate difference between HPV16 and HPV18 transfected cell lines.

```
table(pData(hpvP53rna)[,1:2])
```

Both HPV types lead to cervical cancer. The regulatory networks of their transfected cell lines may share some similarity.



Multiple VAR(1) models (Section 2)

Assume a VAR(1) model per HPV-type, e.g.

$$\mathbf{Y}_{*,t+1,i} \quad = \quad \mathbf{A}_{\mathrm{hpv16}} \mathbf{Y}_{*,t+1,i} + \varepsilon_{*,t+1,i},$$

for i any of the HPV16 transfected cell line and $\varepsilon_{*,t,i} \sim \mathcal{N}(\mathbf{0}_p, \mathbf{\Omega}_\varepsilon^{-1})$.

Thus, a *different* auto-regression parameter but *common* error.

The fused ridge ML estimator of the VAR(1) model maximizes

$$\begin{aligned} \mathrm{loglikelihood} - \lambda_a \| \mathrm{vec}(\mathbf{A}_{\mathrm{hpv16}}) \|_2^2 - \lambda_a \| \mathrm{vec}(\mathbf{A}_{\mathrm{hpv18}}) \|_2^2 \\ - \lambda_f \| \mathrm{vec}(\mathbf{A}_{\mathrm{hpv16}}) - \mathrm{vec}(\mathbf{A}_{\mathrm{hpv18}}) \|_2^2 - \lambda_\omega \| \mathbf{\Omega}_\varepsilon \|_2^2. \end{aligned}$$

Targets $\mathbf{A}_0$ (common to both HPV types) and $\mathbf{\Omega}_0$ could be included.

Multiple VAR(1) models (Section 2)

R-code for fused ridge estimation:

```
# group indicies
id <- c(rep(1, 2), rep(2, 2)) - 1

# search for optimal penalty parameters
optLambdas  <- optPenaltyVAR1fused(Y=Y, id=id, ...)

# fit fused VAR(1) model
VAR1hatF  <- ridgeVAR1fused(Y=Y, id=id, ...)
Ahat16    <- VAR1hatF$As[1:64,]
```

After support inferral of $\mathbf{A}_{\mathrm{hpv16}}$ and $\mathbf{A}_{\mathrm{hpv18}}$, continue either by

- estimation per HPV-type, or
- estimation jointly.

Down-stream analysis proceeds per HPV-type.

Multiple VAR(1) models (Section 2)

Q: jointly estimate the VAR(1) models by means fused ridge estimation.
Spoiler: $(\lambda_a^{\mathrm{opt}}, \lambda_f^{\mathrm{opt}}, \lambda_\omega^{\mathrm{opt}}) = (19.4444, 21.6667, 0.0024)$.

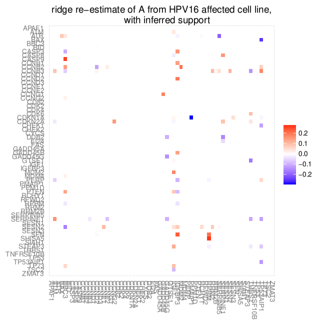
Q: sparsify and callibrate the VAR(1) models.

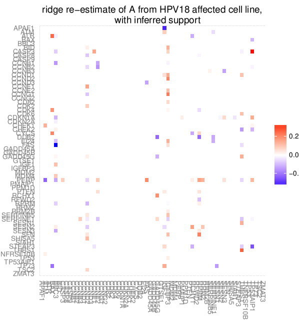
Q: draw heatmaps of the VAR(1) model parameters.

Q: draw the time-series chain graphs of the VAR(1) models.

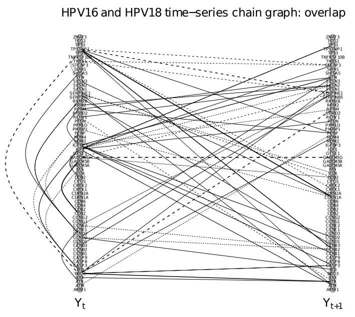
Q: investigate differences in regulation structure.

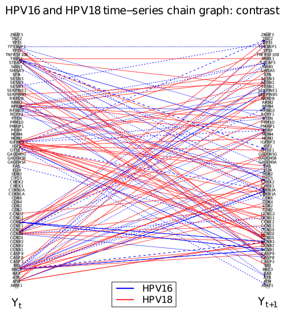
Multiple VAR(1) models (Section 2, answer)





Multiple VAR(1) models (Section 2, answer)

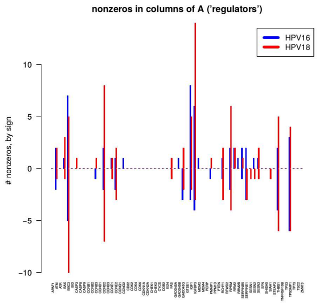


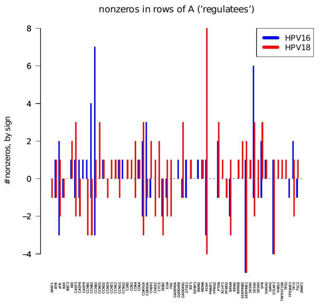


$\#\{(j,j') : (\widehat{\mathbf{A}}_{\mathrm{hpv16}})_{j,j'} \neq 0\} = 88$, and
$\#\{(j,j') : (\widehat{\mathbf{A}}_{\mathrm{hpv18}})_{j,j'} \neq 0\} = 125$

Multiple VAR(1) models (Section 2, answer)

Regulators ≈ the same, while regulatees (e.g. PERP) exhibit some differences.





Q: is this due to differences in # selected elements of $\mathbf{A}$?

The VAR(2) model (Section 3)

The cell line data may be better modelled by a process with a longer memory.

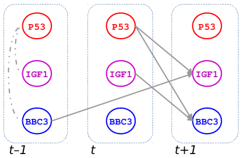
A VAR(2) process:

$$\begin{aligned} Y_{1,t+1} &= && && \varepsilon_{1,t+1}, \\ Y_{2,t+1} &= \alpha_{21}^{(1)} Y_{1,t} + && \underline{\alpha_{13}^{(2)} Y_{3,t-1}} + && \varepsilon_{2,t+1}, \\ Y_{3,t+1} &= \alpha_{31}^{(1)} Y_{1,t} + \alpha_{32}^{(1)} Y_{2,t} + && && \varepsilon_{3,t+1}. \end{aligned}$$

In matrix notation

$$\mathbf{Y}_{t+1} \;=\; \mathbf{A}_1 \mathbf{Y}_t + \mathbf{A}_2 \mathbf{Y}_{t-1} + \varepsilon_{t+1}$$

with $\varepsilon_t \sim \mathcal{N}(\mathbf{0}_p, \mathbf{\Omega}_\varepsilon^{-1})$.



The VAR(2) model

The VAR(2) model can be reformulated as a VAR(1) model:

$$\begin{pmatrix} \mathbf{Y}_{t+1} \\ \mathbf{Y}_t \end{pmatrix} \;=\; \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{I}_{pp} & \mathbf{0}_{pp} \end{pmatrix} \begin{pmatrix} \mathbf{Y}_t \\ \mathbf{Y}_{t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{t+1} \\ \mathbf{0}_p \end{pmatrix}.$$

The same applies to VAR-type models with a higher lag.

The VAR(2) model (Section 3)

R-code:

```
# search for optimal penalty parameters
optLambda <- optPenaltyVAR2(Y, ...)

# fit VAR(2) model
VAR2hat <- ridgeVAR2(Y=Y, lambdaA1=optLambda[1],
                          lambdaA2=optLambda[2], ...)
A1hat <- VAR2hat$A1
A2hat <- VAR2hat$A2
...

# determine support for As and P
zeros <- sparsifyVAR2(Ahat1, Ahat2,
                      SigmaE=symm(solve(Phat)), ...)

# node statistics table
stats <- nodeStatsVAR2(Ahat1, Ahat2, Phat, ...)
```

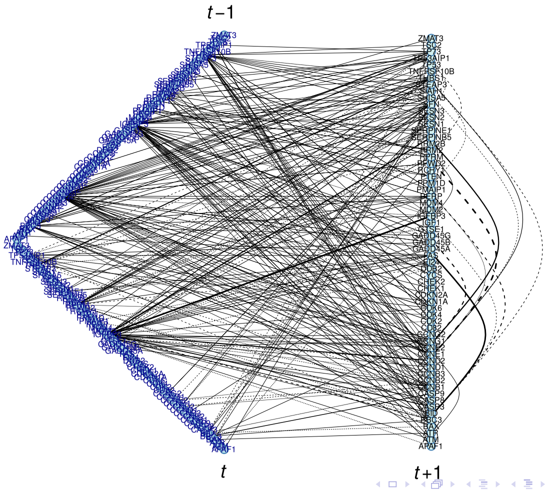
The VAR(2) model (Section 3)

Q: estimate, sparsify, and callibrate the VAR(2) model.

Q: draw the time-series chain graph of the VAR(2) model.

Q: investigate whether the inclusion of the second lag term $\mathbf{A}_2\mathbf{Y}_{t-1}$ is worthwhile.

The VAR(2) model (Section 3, answer)



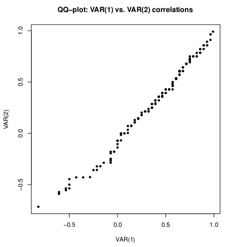
The VAR(2) model (Section 3, answer)

Much smaller elements of $\widehat{\mathbf{A}}_2$ than $\widehat{\mathbf{A}}_1$ as:

initial	$\lambda_{a_1} = 0.0398$	$\lambda_{a_2} = 13.0090$	$\lambda_\omega = 0.0068$
calibration	$\lambda_{a_1} = 0.3310$	$\lambda_{a_2} = 2.0324$	$\lambda_\omega = 0.0063$

The VAR(1) model with
$\#\{(j,j') : (\widehat{\mathbf{A}}\ )_{j,j'} \neq 0\} = 134\}$.
is more parsimonious than
the VAR(2) model with
$\#\{(j,j') : (\widehat{\mathbf{A}}_1)_{j,j'} \neq 0\} = 107\}$ and
$\#\{(j,j') : (\widehat{\mathbf{A}}_2)_{j,j'} \neq 0\} = 138\}$.

The ‘fit’ of the VAR(1) and VAR(2) are comparable.

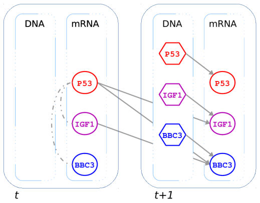


The VARX(1) model (Section 4)

Incorporate time-varying covariates into the VAR(1) model.

A VARX(1) process:

$$\begin{aligned}
Y_{1,t+1} &= \beta_{11} \underline{X_{1,t+1}} + & \varepsilon_{1,t+1}, \\
Y_{2,t+1} &= \beta_{22} \underline{X_{2,t+1}} + \alpha_{21} Y_{1,t} + & \varepsilon_{2,t+1}, \\
Y_{3,t+1} &= \beta_{33} \underline{X_{3,t+1}} + \alpha_{31} Y_{1,t} + \alpha_{32} Y_{2,t} + \varepsilon_{3,t+1}.
\end{aligned}$$



In matrix notation

$$\mathbf{Y}_{t+1} = \mathbf{B}\mathbf{X}_{t+1} + \mathbf{A}\mathbf{Y}_t + \boldsymbol{\varepsilon}_{t+1}$$

with $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}_p, \boldsymbol{\Omega}_{\varepsilon}^{-1})$.

The VARX(1) model (Section 4)

DNA copy number : # copies of a genomic segment present in cell.

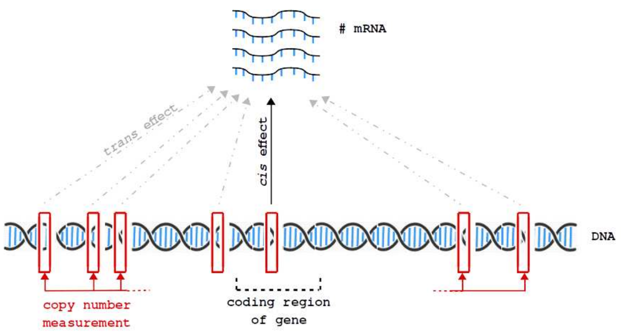
- 2 : autosomal chromosomes,
- 1 : X or Y chrom. in males,
- 0 : Y chromosome in females,
- anything goes in cancer.

A proportionality law:
More genomic (=DNA) copies of a gene
$\Longrightarrow$ more product (=mRNA) it encodes for is formed.

Debate
Is proportionality a localized phenomenon?

The VARX(1) model (Section 4)

We allow only the *cis* (and not the *trans*) effect



This implies a diagonal support of **B**.

The VARX(1) model (Section 4)

R-code:

```
# load covariate data
X <- longitudinal2array(t(copynumber(hpvP53cn)))

# load the constrained elements to zero for B
zerosB <- which(cn2rna==0, arr.ind=TRUE)

# search for optimal penalty parameters
optLambda <- optPenaltyVARX1(Y=Y, X=X, lagX=0,
                             zerosB=zerosB, ...)

# fit VAR(1) model
VAR1hat <- ridgeVARX1(Y=Y, X=X,
                      lambdaA=optLambda1[1],
                      lambdaB=optLambda1[2],
                      lambdaP=optLambda1[3],
                      lagX=0, zerosB=zerosB)

Ahat <- VAR1hat$A
Bhat <- VAR1hat$B
...
```

The VARX(1) model (Section 4)

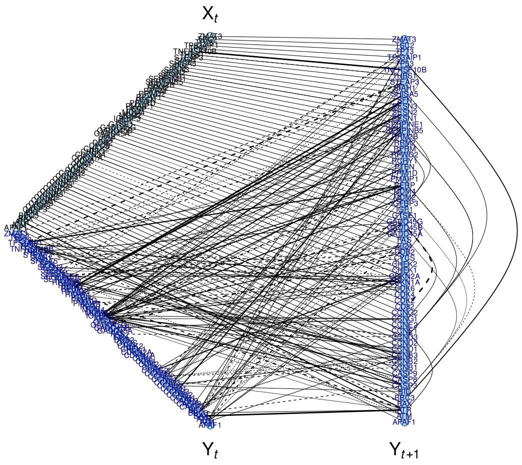
Q: estimate, sparsify, and callibrate the VARX(1) model.

Q: draw the time-series chain graph of the VARX(1) model.

Q: investigate the proportionality law.

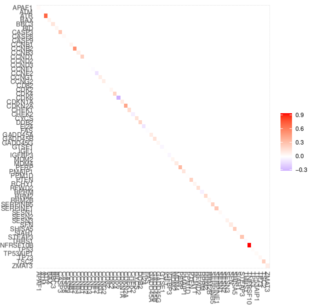
Q: is there complementary regulation?

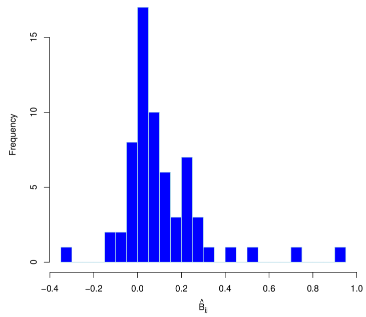
The VARX(1) model (Section 4, answers)



The VARX(1) model (Section 4, answers)

Corroboration of the proportionality law.

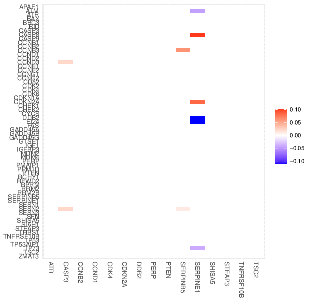




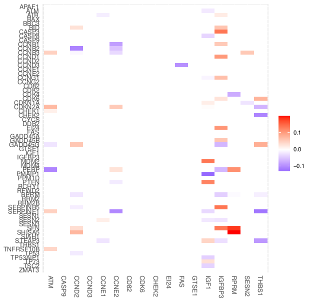
The VARX(1) model (Section 4, answers)

Complementary regulation mechanisms?

$\widehat{\mathbf{A}}$ of the largest $\widehat{\mathbf{B}}$



$\widehat{\mathbf{A}}$ of the smallest $\widehat{\mathbf{B}}$



References

These slides are based on:

- Miok, V., Wilting, S.M., van Wieringen, W.N. (2017). Ridge estimation of the VAR(1) model and its time series chain graph from multivariate time-course omics data. *Biometrical Journal*, 59(1), 172–191
- Miok, V., Wilting, S.M., van Wieringen, W.N. (2019). Ridge estimation of network models from time-course omics data. *Biometrical Journal*, 61(2), 391–405.
- van Wieringen, W.N. (2020) ragt2ridges: Ridge Estimation of Vector Auto-Regressive (VAR) Processes. R package version 0.3.4. `https://CRAN.R-project.org/package=ragt2ridges`.

License