ℓ_2^2 learning of static and dynamic Gaussian graphical models using the `rags2ridges` and `ragt2ridges` packages

C.F.W. Peeters[1], **Wessel N. van Wieringen**[2,3,4]

[1] Mathematical & Statistical Methods group (Biometris), Wageningen University & Research,
Droevendaalsesteeg 1, 6708 PB Wageningen, The Netherlands

[2] Amsterdam UMC, location Vrije Universiteit Amsterdam,
Epidemiology and Data Science, De Boelelaan 1117, Amsterdam, The Netherlands.

[3] Amsterdam Public Health, Methodology, Amsterdam, The Netherlands.

[4] Department of Mathematics, Vrije Universiteit Amsterdam,
De Boelelaan 1111, 1081 HV Amsterdam, The Netherlands.

Brief background

Networks are ubiquitous in modern science. They are a visual means to grasp and communicate a complex of interrelations, providing a systems view on a set of entities. A network is a graph comprising nodes and edges. The nodes depict (the random variables representative of characteristics of) the entities, while the edges indicate relatedness between the node pair that it connects. Networks thus provide an encompassing perspective on the whole in terms of its constituents. This may yield insights not offered by traditional reductionistic (read: univariate) approaches. However, networks are not handed to us by nature but need to be reconstructed from data. Here we briefly illustrate how two `R` packages, `rags2ridges` and `ragt2ridges`, facilitate network reconstruction.

To reconstruct a network from data we need a link between the data and the network. This link is provided by graphical models. Such models comprise a probabilistic description of the whole and a network that captures certain probabilistic properties of this description. For a static network our probabilistic description is a multivariate Gaussian distribution, while for dynamic networks we assume a vector autoregressive process. For both models, the probabilistic property is some form of conditional (in)dependence. In particular, the (absence or) presence of an edge in the network represents such conditional (in)dependence, respectively. In the case of a static network, the conditional independencies correspond to zeroes in the inverse of the distribution's covariance matrix. Similar parametric criteria for conditional independencies exist in the dynamic network case.

Both packages have been developed to reconstruct networks from high-dimensional data. Such data are typically undersampled in relation to the number of model parameters. This is a common issue in the age of high-throughput techniques. In the biomedical context this issue is often encountered in omics studies, where a patient is characterized by techniques that measure the abundance of many molecule types simultaneously. Such undersampling hampers the estimation of the model parameters. This issue can be overcome by regularization: augmenting the loss function, e.g. the likelihood, with a penalty to ensure a well-defined parameter estimator. Both packages implement methodology that takes a so-called ridge or ℓ_2^2-approach to penalization. This is in part motivated by mathematical convenience, but more importantly it aligns with the dense reality of biology.

Below we give a quick taste of the rags2ridges and ragt2ridges packages to give the reader an impression of what these packages can bring to network reconstruction. We focus on the following situations:

i. Extracting a single network from steady-state data (rags2ridges);

ii. Simultaneously extracting multiple networks from multiple related data sets and/or data consisting of distinct (disease) subclasses (rags2ridges);

iii. Extracting networks from time-course data (ragt2ridges).

For each of these situations, the packages provide means to reconstruct and exploit the networks in order to enhance their practical value.

rags2ridges

We first illustrate the extraction, visualization, and analysis of a single network by means of the rags2ridges package. We then shortly turn to the joint extraction, visualization, and analysis of multiple networks.

For our illustration, we employ data stemming from a study into Alzheimer's Disease (AD) that aims to identify disease-specific changes in the underlying biochemical process. Hereto, peripheral fluids of patients have been sampled and interrogated metabolomically. The data comprise metabolomic profiles of 127 patients, comprising individuals with (AD class 2) and without (AD class 1) a known genetic predisposition for AD. Each profile contains the abundance of 230 metabolites. These metabolites each belong to one of four compound families: amines, organic acids, lipids, or oxidative stress compounds. More details on the data can be found in de Leeuw *et al.* (2017). The package and data are loaded into memory by the first few lines of the R-code block in Listing 1.

Listing 1: R code

```r
# needed package
library("rags2ridges")

# load, extract and scale data for AD Class 2
data("ADdata")
ADclass2 <- scale(t(ADmetabolites[,sampleInfo$ApoEClass=="Class␣2"]))

# precision matrix estimation with given penalty value**
P <- ridgeP(covML(ADclass2), lambda = .15)

# extract Network**
P0 <- sparsify(P, threshold="localFDR", FDRcut=.999)

# visualize Network with node-coloring**
PcorP  <- pruneMatrix(P0$sparseParCor)
Colors <- rownames(PcorP)
Colors[grep("Amine",     rownames(PcorP))] <- "lightblue"
Colors[grep("Org.Acid",  rownames(PcorP))] <- "orange"
Colors[grep("Lip",       rownames(PcorP))] <- "yellow"
Colors[grep("Ox.Stress", rownames(PcorP))] <- "purple"

# plot network
Ugraph(PcorP, type="fancy", lay="layout_with_fr",
       Vcolor=Colors, Vsize=7, Vcex=.3)

# find and visualize the communities for the extracted network
Commy <- Communities(PcorP, Vcolor=Colors, Vsize=7,  Vcex=.3)
```

As an illustration we concentrate on the reconstruction of the metabolic network for AD class 2, which comprises 87 individuals. Hereto we fit a Gaussian graphical model. Effectively, this amounts to the estimation of the inverse of the covariance matrix of a multivariate normal. As the variable dimension (230) exceeds the sample size (87), we use ridge-regularization as carried out by the `ridgeP`-function in the `R`-code. One then obtains an estimate of the regularized inverse covariance matrix for a given value of the penalty parameter. The value of the penalty parameter can also be chosen in a data-driven manner. For instance, by K-fold cross-validation as provided through the `optPenalty`-function. The resulting inverse covariance estimate does not harbor any off-diagonal zero entries, i.e. it does not represent a sparse network. For the purpose of network reconstruction, the inverse covariance matrix estimate is 'sparsified', i.e. its support is determined by the identification of elements that are indistinguishable from zero. The `sparsify`-function implements several pragmatic but also more sophisticated, probabilistically motivated procedures to this end (such as a local false discovery rate). The sparsified matrix then represents the reconstructed network and can be plotted for visual inspection by the `Ugraph`-function. Here, we have added attributes to the nodes: they are colored according to compound family (left-hand side of Figure 1). This highlights the role of the compound families in the network topology. The amines and organic acids (light blue and orange nodes) form a core structure, while the oxidative stress and lipid compounds (purple and yellow nodes) can be found more at its periphery.

The `rags2ridges`-package offers, to enhance the practical value of the results, several options for downstream analysis of the reconstructed network. A natural starting point is the `GGMnetworkStats`-function that calculates various node statistics like centrality measures. These provide a quantification of the nodes' importance in the network. An alternative analysis, implemented in the `GGMpathStats`-function, quantifies the contribution of all paths that connect a node pair. This sheds light on the strongest routes by which a signal propagates in the network from one node to the other. A different perspective is provided by community analysis of the network through the `Communities`-function. This identifies functional modules (or subnetworks) within the full network. The result is portrayed in the right-hand side panel of Figure 1, which overlays the network with the found communities.

All the above functionality has also been implemented in the `rags2ridges`-package for the multi-group case, with the aim to identify differential network features among groups or subclasses. This implementation carries identical function names suffixed with `.fused`. The downstream functions implement the same analyses as above, but now executed group-wise. The `ridgeP.fused`-function, however, performs the joint estimation of multiple, class-specific inverse covariance matrices using a fused ridge penalty. This fusion enables the estimation to borrow information across groups. This may yield estimates that have been shrunken towards each other, should the data support that.

ragt2ridges

The `ragt2ridges` package is `rags2ridges`' sister package but focusses on dynamic networks. Like its sister, `ragt2ridges` implements methodology to reconstruct dynamic networks but also provides handles for their downstream exploitation.

A brief illustration of the `ragt2ridges` package uses an *in vitro* oncogenomics study with a longitudinal experimental design conducted to unravel the dynamic interactions among genes during cervical carcinogenesis. The human papilloma virus (HPV), a carcinogenic entity, is inserted

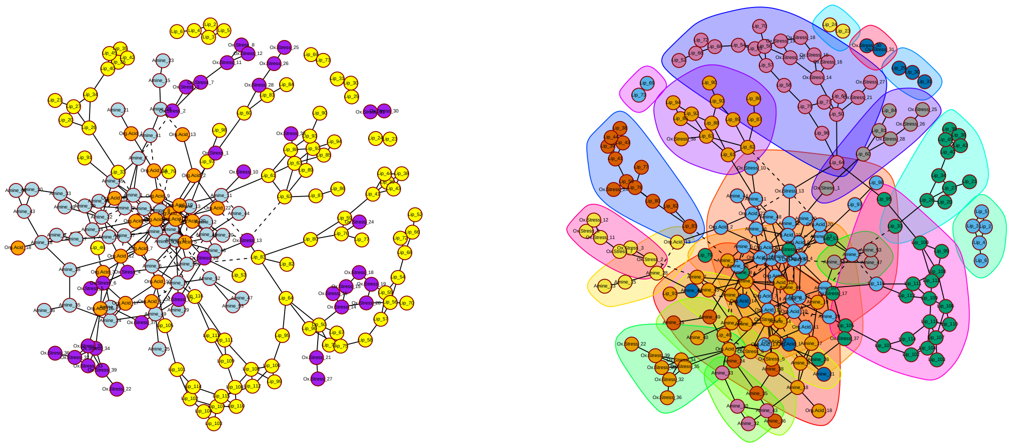

Figure 1: Left: conditional independence graph with node colors corresponding to metabolite compound family. Right: the conditional independence graph overlayed with communities.

into normal cells, yielding an immortalized cell line that faithfully mimics cervical cancer development morphologically and genetically. As the infected cell line goes through distinct phenotypic phases, cells are profiled transcriptomically at eight time points distributed over the transformation process. The observed changes in transcript levels shed light on the underlying process of carcinogenesis. A full description of the experiment can be found in Babion *et al.* (2020). The data are loaded into memory by the first few lines in the block of R code below (Listing 2) and stored in the Y object of the array class.

Listing 2: R code

```r
# load package and data
library(ragt2ridges)
library(Biobase)
data(hpvP53)

# reformat and zero center data
Y <- centerVAR1data(longitudinal2array(t(exprs(hpvP53rna))))

# fit the model
VAR1hat <- ridgeVAR1(Y=Y, lambdaA=100, lambdaP=1)

# support determination
zerosA <- sparsifyVAR1(A=VAR1hat$A, SigmaE=symm(solve(VAR1hat$P)),
                       threshold="top", top=50)$zeros
VAR1hat$A[zerosA] <- 0
VAR1hat$P <- sparsify(VAR1hat$P, threshold="top", top=10)$sparseParCor

# plot time-series chain graph
graphVAR1(VAR1hat$A, VAR1hat$P, nNames=featureNames(hpvP53rna))
```

```
# motif detection
motifStatsVAR1(VAR1hat$A)
```

The experimental data are analyzed by means of a vector autoregressive (VAR) model, describing the temporal and contemporaneous relations among the genes. The VAR model is a regression-type model. It explains the current vector of observations by linear combinations of the elements of such vectors from preceding time points plus a noise vector. The estimation of the VAR model is however hampered by the high-dimensionality of the cervical cell line data. This is overcome by ridge penalized maximum likelihood estimation of the VAR model as performed by the `ridgeVAR1` function in Listing 2. For simplicity, the penalty parameters are set in the `R`-code but need – of course – be chosen in a more informed fashion by, e.g. a cross-validation scheme as provided through the `optPenaltyVAR1`-function. The implemented ridge estimation procedure allows for the incorporation of both quantitative and qualitative information regarding the unknown parameters, in particular the absence of temporal and contemporaneous relations. Moreover, attention has been paid to ensure a computational and memory-efficient implementation.

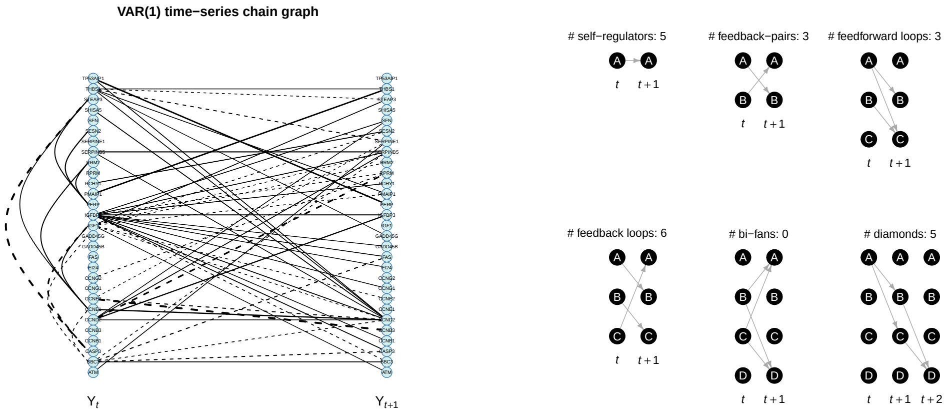

Figure 2: Left: time-series chain graph with temporal (straight) and contemporaneous (bended) edges. Right panel: visual representation of the motif tally.

The `ragt2ridges` package offers various strategies for the downstream utilization of the estimated VAR model. For starters, the `spasifyVAR1`-function provides a selection procedure for the identification of interesting temporal and contemporaneous edges of the time series chain graph. In the `R`-code above, simply the strongest edges are selected but a more sophisticated, probabilistically motivated option is also implemented. With the support of the time-series chain graph reconstructed, the `graphVAR1`-function offers various ways to visualize it (see the left panel of Figure 2 for an example). Descriptive node statistics of this network are calculated by the `nodeStatsVAR1`- and `mutualInfoVAR1`-functions. Alternatively, the `motifStatsVAR1`-function tallies network motifs, i.e. subnetworks that are associated with particular dynamic behaviour. The right-hand side panel of Figure 2 shows the output of this function. The `impulseResponseVAR1`-function offers a different view through the quantification of the downstream effect of a node's perturbation.

The illustration and `R`-code above implicitly center on the use of a VAR(1) model, i.e. a VAR model with lag one. Such a model only uses the directly preceding observation vector to explain the current one. The `ragt2ridges`-package also include functionality to learn a VAR(2) model, a VAR model with lag two. Or, a VARX(1) model, a VAR model with lag one and time-varying covariates.

Conclusion

We hope to have given a little taste of what our two packages are capable of and how they may assist you in your network analysis. Should you previously not have reconstructed networks from data, we hope that our software exposé lowers the threshold to do so and convinces you that network analysis is worth a try. Finally, should you attend the International Biometric Conference 2022 in Riga, we will teach a pre-conference course that explores the possibilities of these packages more in-depth.

References

Babion, I., Miok, V., Jaspers, A., Huseinovic, A., Steenbergen, R. D., van Wieringen, W. N., and Wilting, S. M. (2020). Identification of deregulated pathways, key regulators, and novel miRNA-mRNA interactions in HPV-mediated transformation. *Cancers*, **12**(3), 700.

de Leeuw, F. A., Peeters, C. F. W., Kester, M. I., Harms, *et al.* (2017). Blood-based metabolic signatures in Alzheimer's disease. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, **8**, 196–207.