Semi supervised and soft modeling with applications

Przemysław Biecek, Ewa Szczurek, Małgorzata Zynek-Litwin

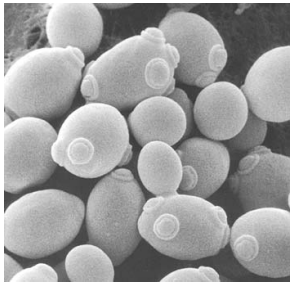
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw

Max Planck Institute for Molecular Genetics, Berlin

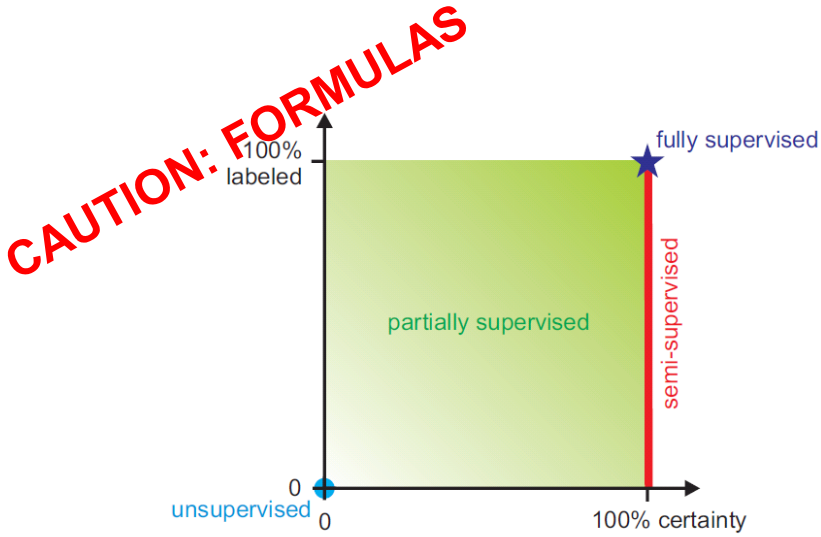
Department of Nephrology and Transplantation Medicine, Wrocław Medical University

ICB Seminar, 2011

Outline

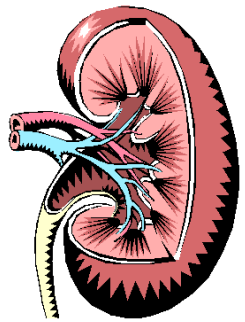


Ste12 Transcription Factor



Fully supervised modeling
Semi-supervised modeling
Partially supervised modeling
Unsupervised modeling

The R Package bgmm:
Mixture Modeling with Uncertain Knowledge



Prediction of the expression of
metallopeptidase inhibitor 2 (TIMP2)

Finding Ste12 target genes

Ste12 is a Transcription Factor (TF) which regulates a diverse set of genes required for different developmental phenotypes in yeast.

The environment-dependent association of Ste12 with other proteins can lead to its binding to different sets of genes (Zeitlinger et al., 2003).

We are interested in genes that are targeted by Ste12 and differently respond to pheromone stimulation (environment).

Data comes from (Roberts et al., 2000).

Finding Ste12 target genes

- Four sets of conditions. For each gene we have following gene expressions measured as log intensities from microarray experiments. The question is: for which genes the response for pheromone differs between wild type and Ste12 mutant.

	Untreated	50 nM of alpha-factor (pheromone)
Wild type	X_{11}	X_{12}
Ste12 mutant (knock-out)	X_{21}	X_{22}

- For balanced two way ANOVA the interaction effect might be defined as

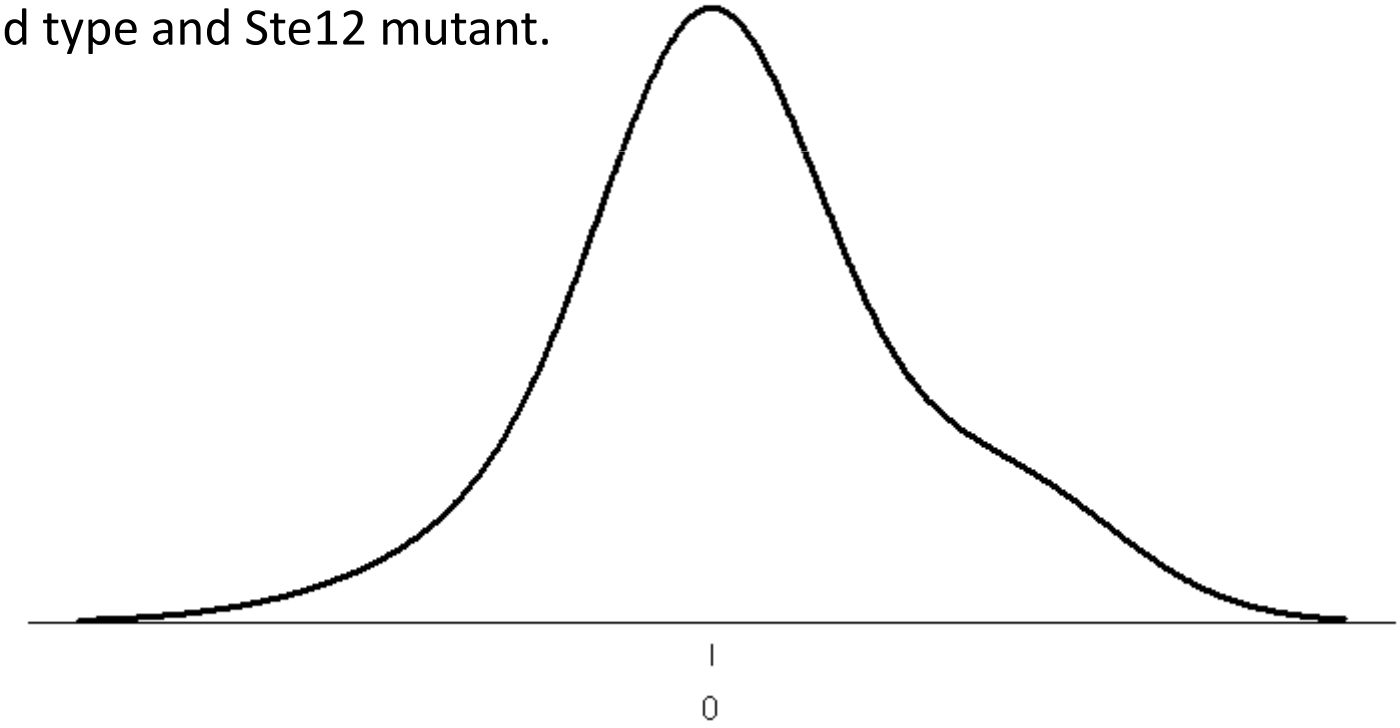
$$X_{22} - X_{21} - X_{12} + X_{11}$$

Finding Ste12 target genes

The distribution of interaction effects among genes.

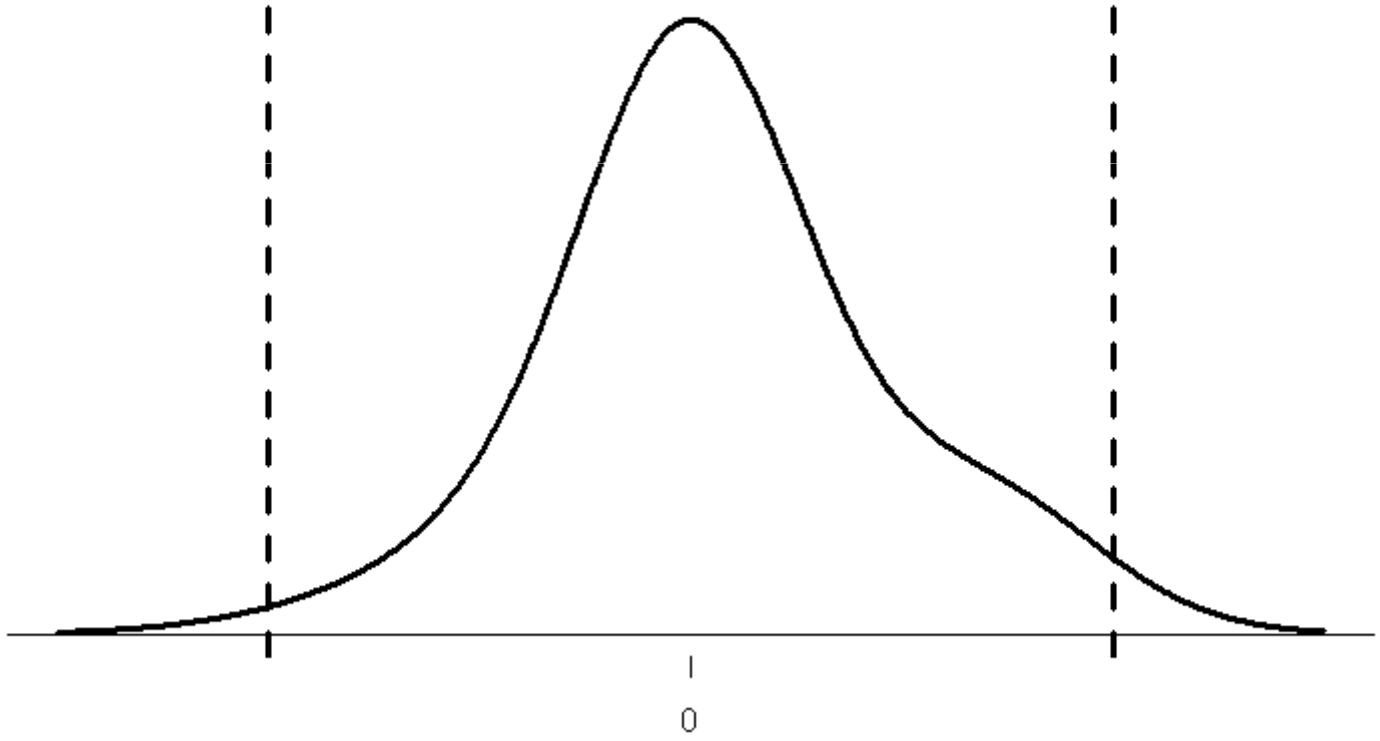
Here only a model is presented, the observed distribution is not so smooth.

The question: for which genes the answer for pheromone differs between wild type and Ste12 mutant.



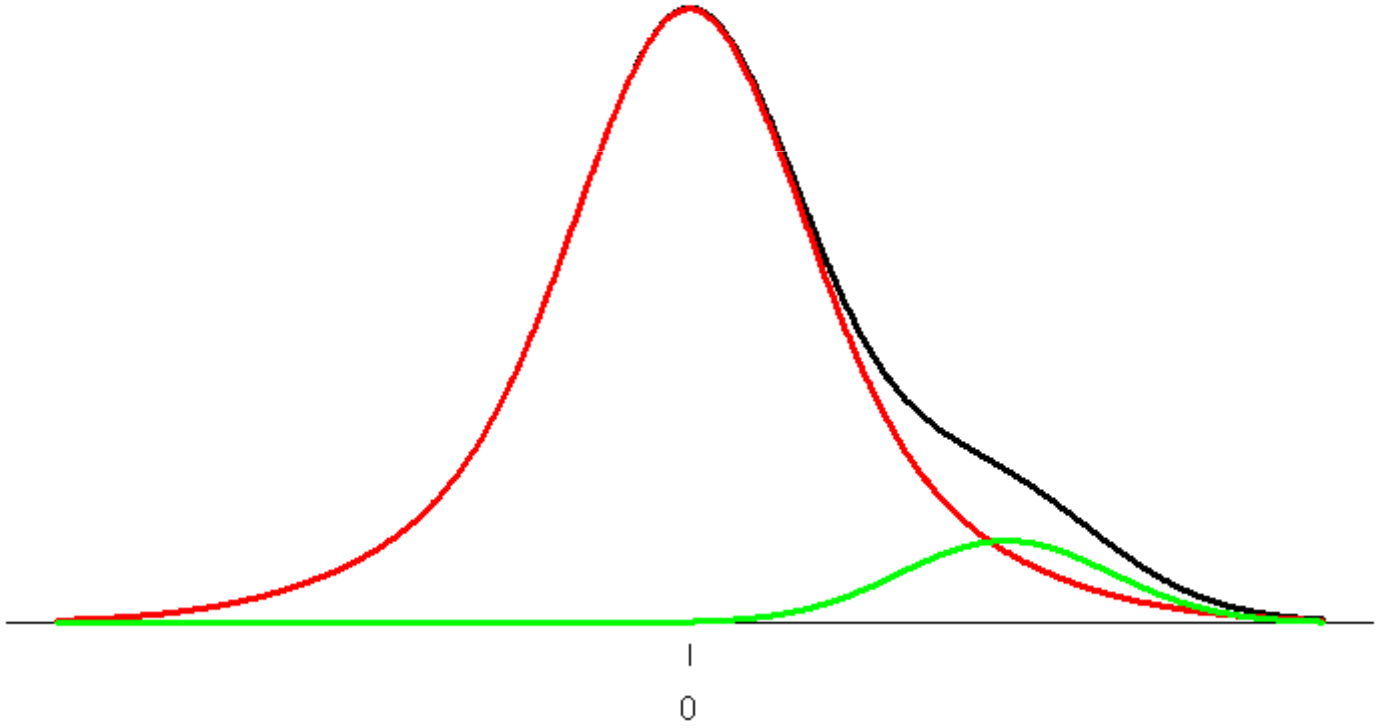
Finding Ste12 target genes

Approach 1. Test for significance of the interaction effect in the two way ANOVA model.



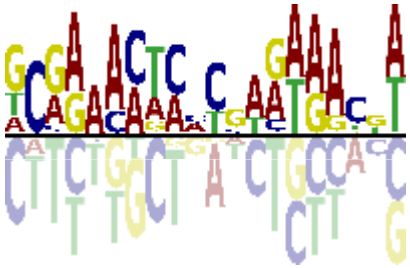
Finding Ste12 target genes

Approach 2. Build a Gaussian mixture model and predict the posterior probability that gene i responds differently to pheromone depending on the presence of Ste12.



Finding Ste12 target genes

From literature studies (Harbison et al., 2004) we know a set of genes with predicted or experimentally confirmed binding of Sta12 in the gene promoter.



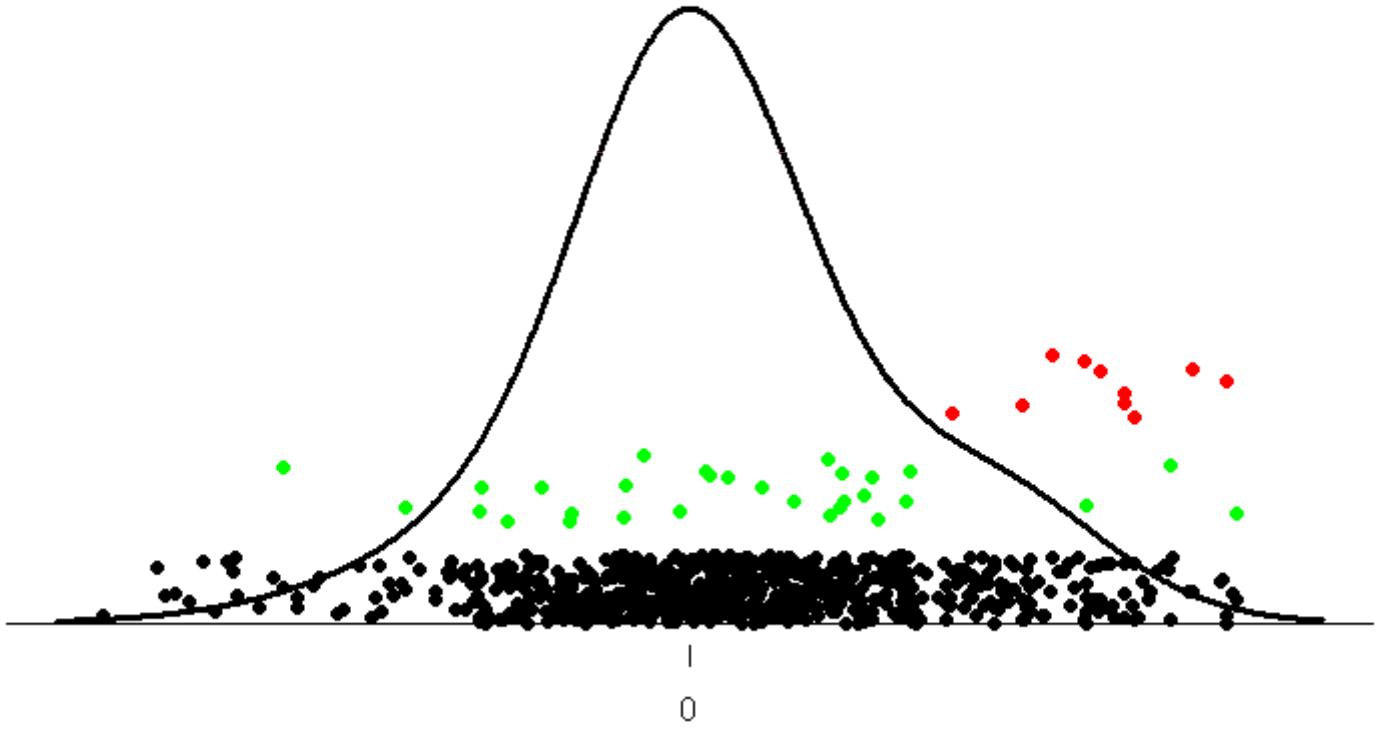
For genes with predicted binding site of Sta12 we belief that Sta12 may affect the gene response to the pheromone.

For genes with experimentally confirmed bindings we are almost sure that Sta12 affect expression of these genes.

For some genes (e.g. housekeeping genes) we are almost sure that Sta12 does not affect expressions of these genes.

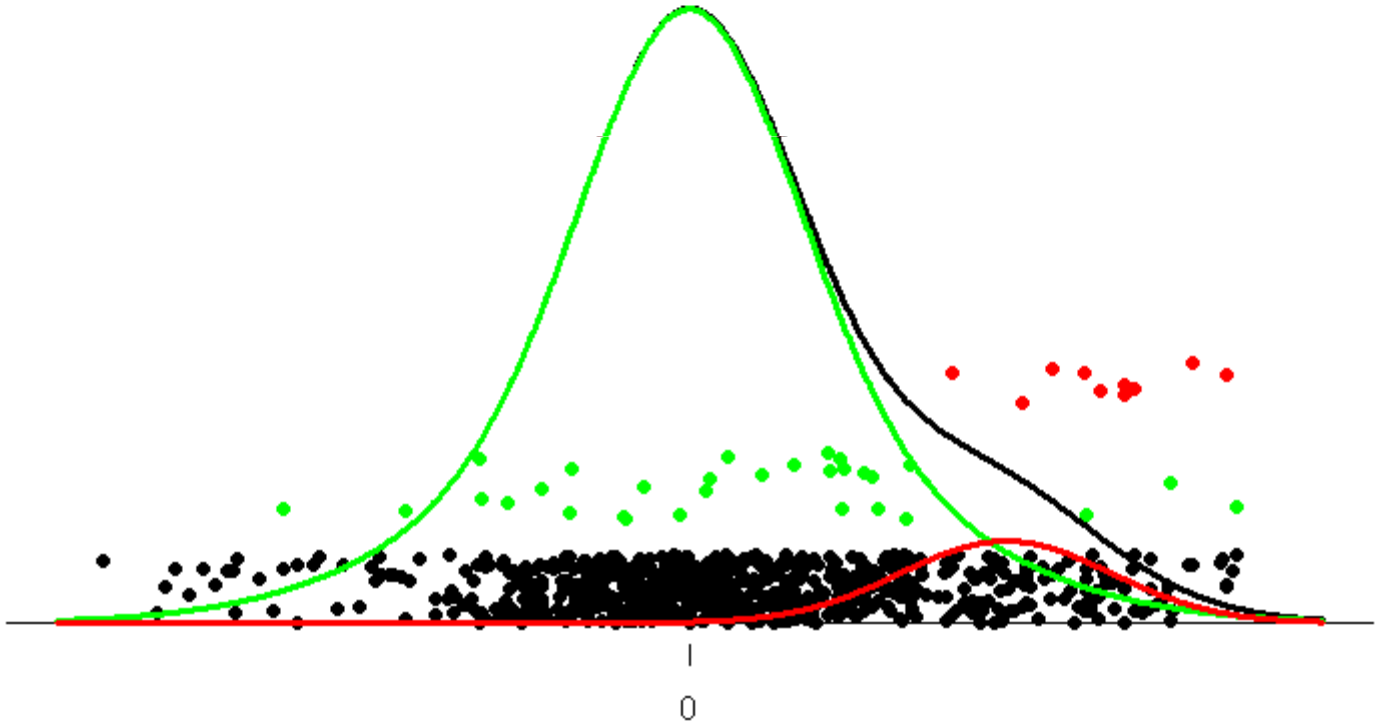
Finding Ste12 target genes

Approach 3. Classification based on genes known or assumed labels.



Finding Ste12 target genes

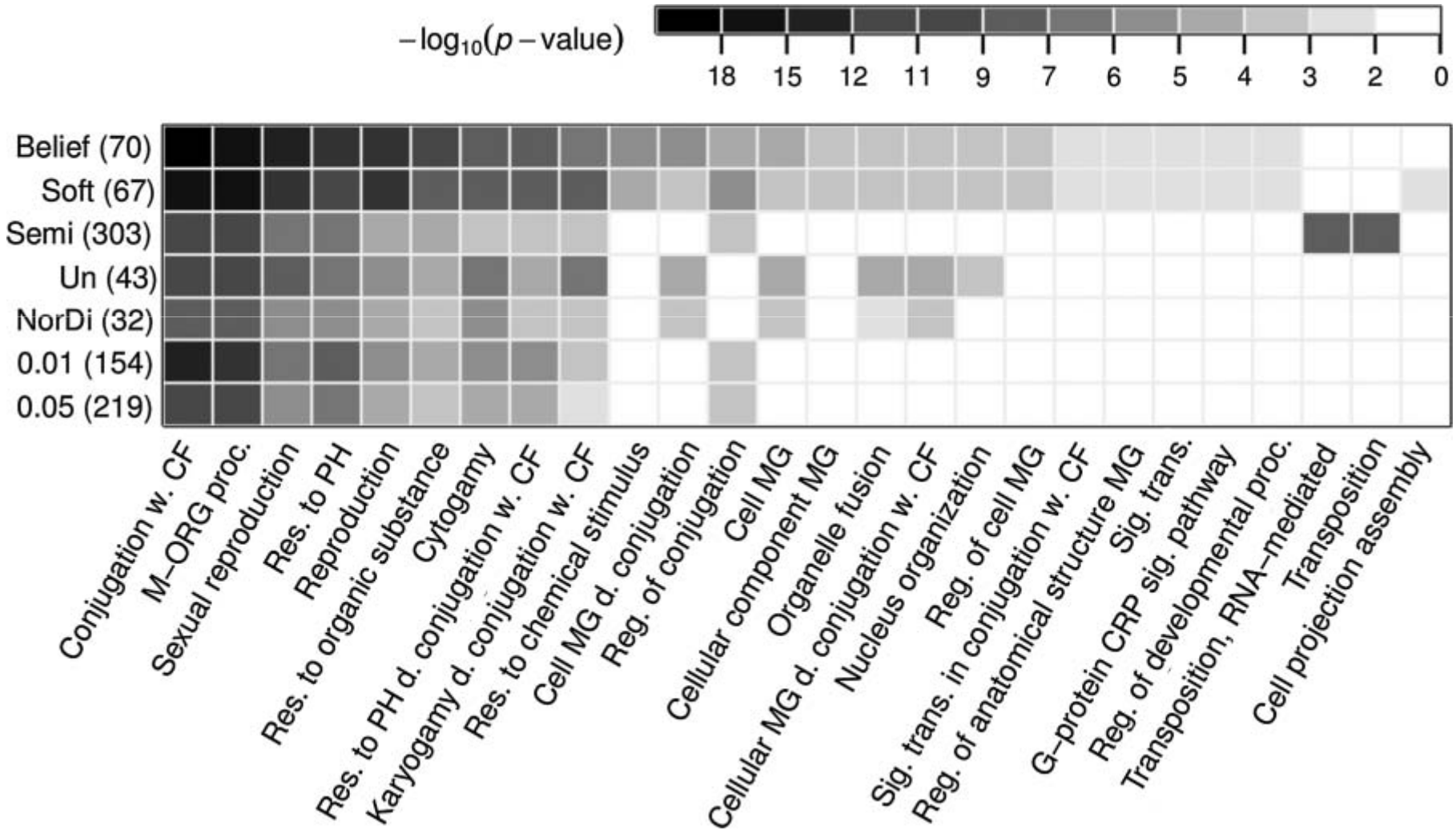
Our Approach. Build a Gaussian mixture model taking into account additional knowledge / labeled samples.



Finding Ste12 target genes

- From each approach we get a set of genes marked as the response genes, i.e. genes that response to pheromone and are regulated by Sta12.
- In general the validation here is hard, since we do not know which genes behave in the expected way, there is no gold standard.
- To validate our results we use enrichment analysis for Gene Ontology terms.

Finding Ste12 target genes



Gaussian Mixture Modeling

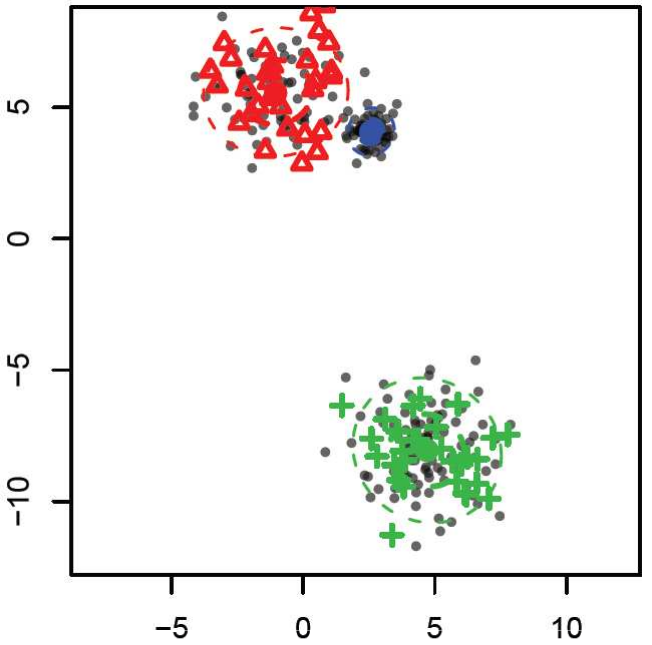
Consider a Gaussian mixture model, defined by a pair of random variables (X; Y), where X takes values in R^d while Y takes values in the set 1, ..., k. Here k defines the number of mixture components. X|Y = y follows a multidimensional Gaussian distribution with parameters μ_y, Σ_y. The marginal density

$$f(x) = \sum_{y=1}^{k} \pi_y f(x, \theta_y)$$

Variants of Gaussian mixture modeling

1. Fully supervised modeling: Variable Y is observed for all samples (lda(), qda() in the MASS package).
2. Semi-supervised modeling: Variable Y is observed for m samples (0<m<n).
3. Partially supervised modeling: Variable Y is observed for m samples (0<m<n), but those observations are uncertain, i.e. their values are given with some probability, we will discuss two partially supervised mixture modeling methods: belief-based and soft-label mixture modeling.
4. Unsupervised modeling: Variable Y is not observed for any sample (mclust).

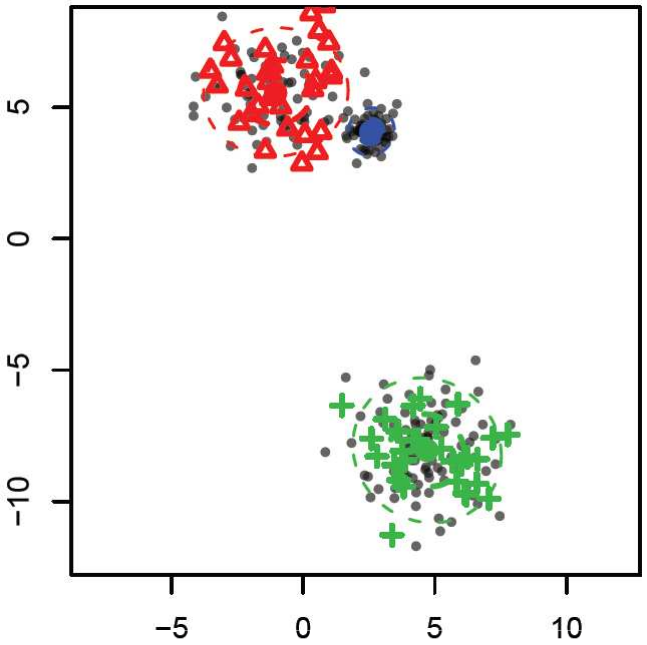
Goals



Having labeled and unlabeled cases in the training set

- Find a structure in the distribution of X (as in the cluster analysis)
- Predict labels Y for unlabeled samples or new samples (as in the classification)

Problems



- How to estimate model parameters π_y, μ_y, Σ_y?
- How to choose the number of model components k?
- How to choose the structure for model parameters?

Underlying theory

- For fully supervised modeling, parameters may be estimated directly (as in linear discriminant analysis or in quadratic discriminant analysis).
- For other variants we use the EM (expectation maximization algorithm). Y is threated as partially or fully unobserved variable. EM is used to maximize the join likelihood $L(\pi_y, \mu_y, \Sigma_y, X, Y)$ in an iterative manner.
  - In step E the expected value of the likelihood with respect to the conditional distribution of Y given under the current estimate of the parameters π_y, μ_y, Σ_y,
  - In step M new parameters are obtained as values that maximize the objective function.

Unsupervised modeling

- The likelihood function

$$l(\mathcal{X}, P, \Phi) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \pi_k f(x_i, \theta_j) \right)$$

normal distribution

parameters for jth Gaussian component

mixture proportions

- Estimates for model parameters

$$t_{i,j}^{(q+1)} = \pi_j^{(q)} f\left(x_i, \theta_j^{(q)}\right) / \sum_{l=1}^{k} \pi_l^{(q)} f\left(x_i, \theta_l^{(q)}\right)$$

$$\pi_j^{(q+1)} = \sum_{i=1}^{n} t_{i,j}^{(q+1)} / n$$

$$\mu_j^{(q+1)} = \left(\sum_{i=1}^{n} x_i t_{i,j}^{(q+1)} \right) / \left(\sum_{i=1}^{n} t_{i,j}^{(q+1)} \right)$$

$$\Sigma_j^{2\ (q+1)} = \left(\sum_{i=1}^{n} \left(x_i - \mu_j^{(q+1)} \right)^T \left(x_i - \mu_j^{(q+1)} \right) t_{i,j}^{(q+1)} \right) / \left(\sum_{i=1}^{n} t_{i,j}^{(q+1)} \right)$$

Soft-label modeling

- The likelihood function

priors are weighted by plausibilities

$$l(\mathcal{X}, P, \Phi) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} p_{i,j} \pi_k f(x_i, \theta_j) \right)$$

- Estimates for model parameters

$$t_{i,j}^{(q+1)} = p_{i,j} \pi_j^{(q)} f\left(x_i, \theta_j^{(q)}\right) / \sum_{l=1}^{k} p_{i,l} \pi_l^{(q)} f\left(x_i, \theta_l^{(q)}\right), \quad i \leq M,$$

$$\pi_j^{(q+1)} = \sum_{i=1}^{n} t_{i,j}^{(q+1)} / n,$$

$$\mu_j^{(q+1)} = \left(\sum_{i=1}^{n} x_i t_{i,j}^{(q+1)} \right) / \left(\sum_{i=1}^{n} t_{i,j}^{(q+1)} \right)$$

$$\Sigma_j^{2\ (q+1)} = \left(\sum_{i=1}^{n} \left(x_i - \mu_j^{(q+1)}\right)^T \left(x_i - \mu_j^{(q+1)}\right) t_{i,j}^{(q+1)} \right) / \left(\sum_{i=1}^{n} t_{i,j}^{(q+1)} \right)$$

Belief-based modeling

- The likelihood function

priors are replaced by beliefs

$$l(\mathcal{X}, B, \Phi) = \sum_{i=1}^{m} \log\left(\sum_{j=1}^{k} b_{i,j} f(x_i, \theta_j)\right) + \sum_{i=m+1}^{n} \log\left(\sum_{j=1}^{k} \pi_j f(x_i, \theta_j)\right),$$

- Estimates for model parameters

$$t_{i,j}^{(q+1)} = \begin{cases} b_{i,j} f\left(x_i, \theta_j^{(q)}\right) / \sum_{l=1}^{k} b_{i,l} f\left(x_i, \theta_l^{(q)}\right) & i \le M \\ \pi_j f\left(x_i, \theta_j^{(q)}\right) / \sum_{l=1}^{k} \pi_l f\left(x_i, \theta_l^{(q)}\right) & i > M \end{cases}$$

$$\pi_j^{(q+1)} = \sum_{i=m+1}^{n} t_{i,j}^{(q+1)} / (n-m)$$

$$\mu_j^{(q+1)} = \left(\sum_{i=1}^{n} x_i t_{i,j}^{(q+1)}\right) / \left(\sum_{i=1}^{n} t_{i,j}^{(q+1)}\right)$$

$$\Sigma_j^{2\ (q+1)} = \left(\sum_{i=1}^{n} \left(x_i - \mu_j^{(q+1)}\right)^T \left(x_i - \mu_j^{(q+1)}\right) t_{i,j}^{(q+1)}\right) / \left(\sum_{i=1}^{n} t_{i,j}^{(q+1)}\right)$$

Differences among GMM variants

- Semi-suprvised, soft and belief-based modeling incorporate information about partially available labels.
- The fuzzy modeling (soft and belief-based) gives more robust results.
- The key difference between soft and belief-based modeling is in the interpretation of weights. For soft modeling weights, called plausibilities, modify the posteriors and need to be specified for all samples. For belief-based modeling weights, called beliefs, act as posteriors.

Model selection: # of components

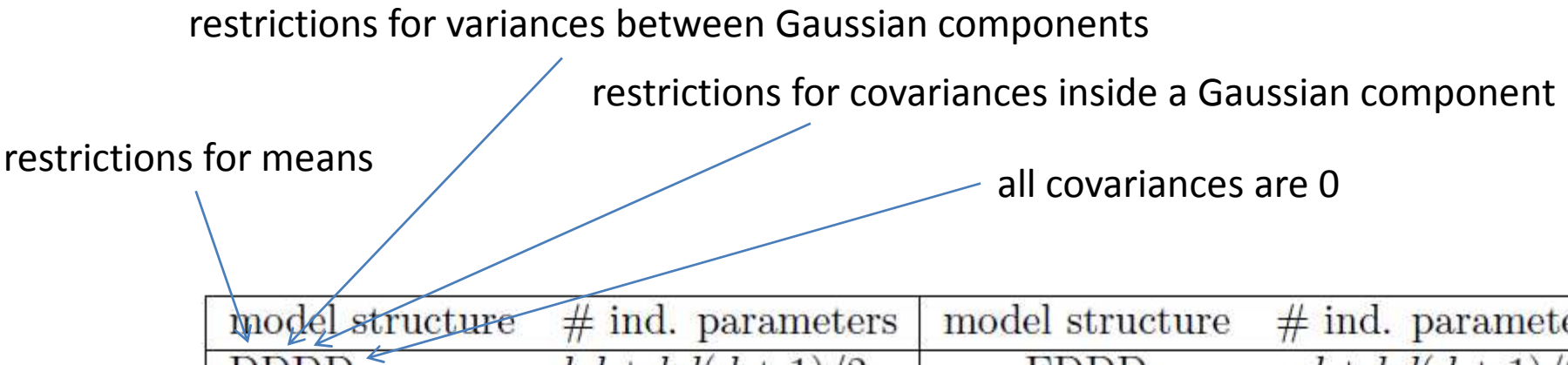
- The GIC (Generalized Information Criteria)

$$GIC(\Phi) = -2l(\mathcal{X}^{-}, \Phi) + p|\Phi|,$$

- X- stands for the set of unlabeled observations. This allows for a comparison of the models fitted by unsupervised modeling with models fitted by the semi- or partially supervised methods.
- Φ stands for the set of all model parameters. Might be smaller than k*d*(d+3)/2 parameters while one can put some restrictions over model parameters π_y, μ_y, Σ_y.

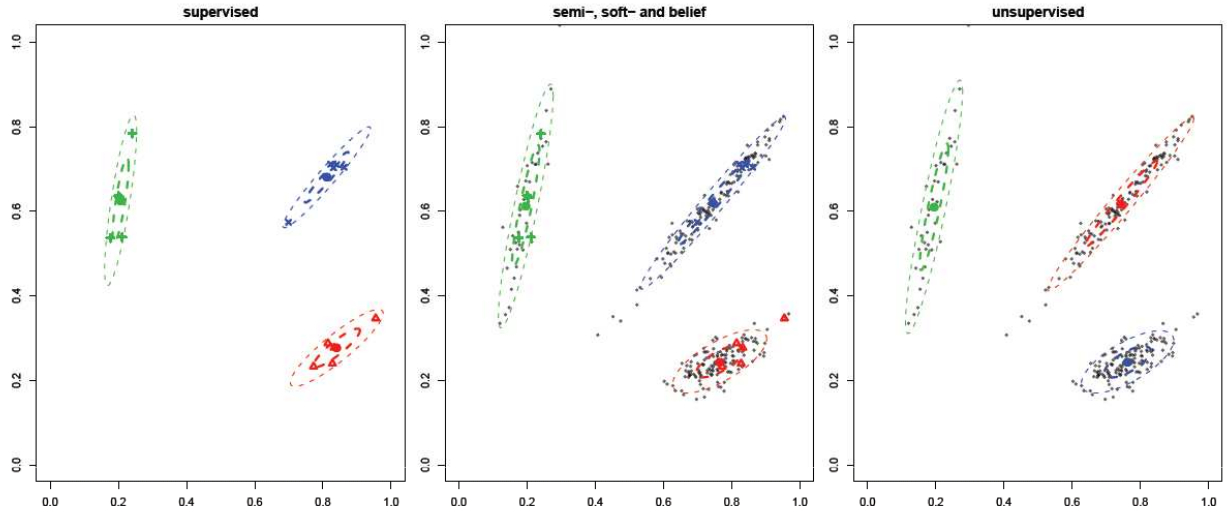
Model selection: the model structure

In some cases it is advisable to lower number of model parameters by putting some restrictions over π_y, μ_y, Σ_y.



model structure	# ind. parameters	model structure	# ind. parameters
DDDD	$kd + kd(d+1)/2$	EDDD	$d + kd(d+1)/2$
DDD0	$kd + kd$	EDD0	$d + kd$
DDED	$kd + 2k$	EDED	$d + 2k$
DDE0	$kd + k$	EDE0	$d + k$
DEDD	$kd + d(d+1)/2$	EEDD	$d + d(d+1)/2$
DED0	$kd + d$	EED0	$d + d$
DEED	$kd + 2$	EEED	$d + 2$
DEE0	$kd + 1$	EEE0	$d + 1$

The bgmm package

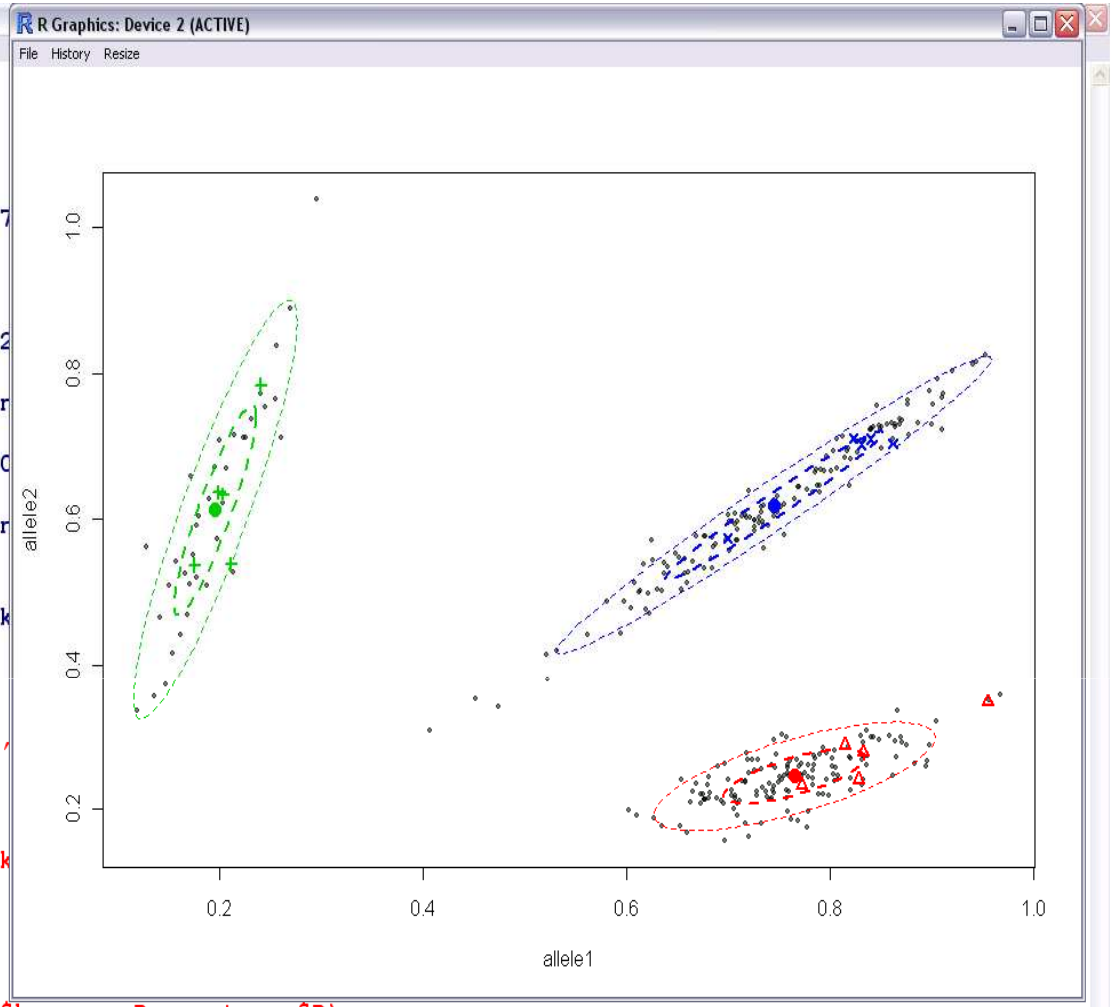


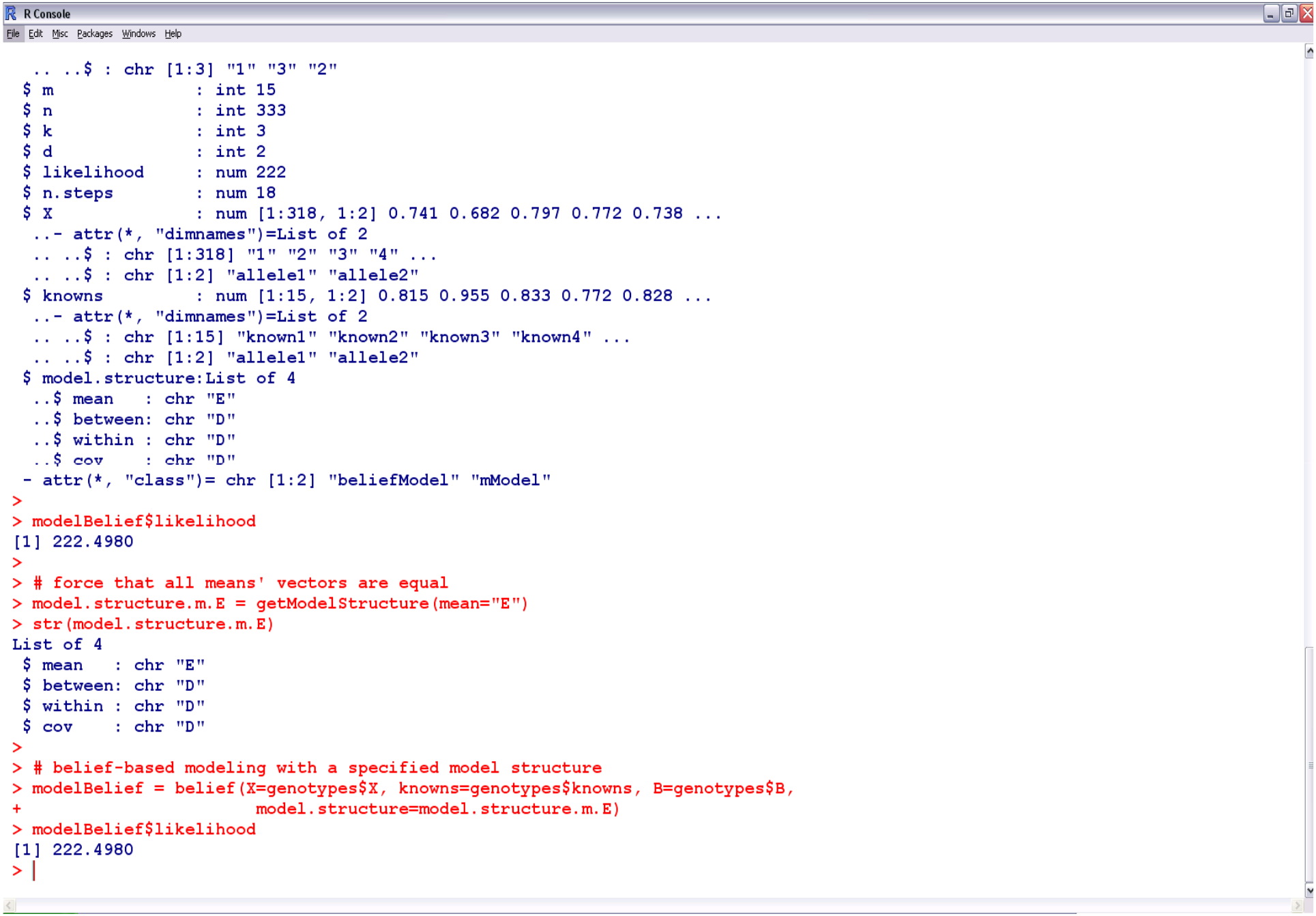

```
R> library("bgmm")
R> data("genotypes")

R> modelSupervised = supervised(
R> plot(modelSupervised)

R> modelSemiSupervised = semisupervised(X=genotypes$X, knowns=genotypes$knowns,
      class=genotypes$labels)
R> plot(modelSemiSupervised)

R> modelBelief = belief(X=genotypes$X, knowns=genotypes$knowns, B=genotypes$B)
R> plot(modelBelief)

R> modelSoft = soft(X=genotypes$X, knowns=genotypes$knowns, P=genotypes$B)
R> plot(modelSoft)

R> modelUnSupervised = unsupervised(X=genotypes$X, k=3)
R> plot(modelUnSupervised)
```

R Console

Edit Misc Packages Windows Help

```
> library(bgmm)
> data(genotypes)
> str(genotypes)
List of 4
 $ X      : num [1:318, 1:2] 0.741 0.682 0.797 0.772 0.7
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:318] "1" "2" "3" "4" ...
  .. ..$ : chr [1:2] "allele1" "allele2"
 $ knowns: num [1:15, 1:2] 0.815 0.955 0.833 0.772 0.82
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:15] "known1" "known2" "known3" "knowr
  .. ..$ : chr [1:2] "allele1" "allele2"
 $ B      : num [1:15, 1:3] 0.95 0.95 0.95 0.95 0.95 0.0
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:15] "known1" "known2" "known3" "knowr
  .. ..$ : chr [1:3] "1" "3" "2"
 $ labels: Named int [1:15] 1 1 1 1 1 3 3 3 3 3 ...
  ..- attr(*, "names")= chr [1:15] "known1" "known2" "k
>
> # supervised modeling
> # applied only to the 15 labeled observations
> modelSupervised = supervised(knowns=genotypes$knowns,
> plot(modelSupervised)
>
> # semi-supervised modeling
> modelSemiSupervised = semisupervised(X=genotypes$X, k
+                              class = genotypes$labels)
> plot(modelSemiSupervised)
>
> # belief-based modeling
> modelBelief = belief(X=genotypes$X, knowns=genotypes$knowns, B=genotypes$B)
> plot(modelBelief)
>
> # soft-label modeling
> modelSoft = soft(X=genotypes$X, knowns=genotypes$knowns, P=genotypes$B)
> plot(modelSoft)
>
> # unsupervised modeling
> modelUnSupervised = unsupervised(X=genotypes$X, k=3)
>
```




R Console
File Edit Misc Packages Windows Help
.. ..$: chr [1:3] "1" "3" "2"
$ m : int 15
$ n : int 333
$ k : int 3
$ d : int 2
$ likelihood : num 222
$ n.steps : num 18
$ X : num [1:318, 1:2] 0.741 0.682 0.797 0.772 0.738 ...
..- attr(*, "dimnames")=List of 2
.. ..$: chr [1:318] "1" "2" "3" "4" ...
.. ..$: chr [1:2] "allele1" "allele2"
$ knowns : num [1:15, 1:2] 0.815 0.955 0.833 0.772 0.828 ...
..- attr(*, "dimnames")=List of 2
.. ..$: chr [1:15] "known1" "known2" "known3" "known4" ...
.. ..$: chr [1:2] "allele1" "allele2"
$ model.structure:List of 4
..$ mean : chr "E"
..$ between: chr "D"
..$ within : chr "D"
..$ cov : chr "D"
- attr(*, "class")= chr [1:2] "beliefModel" "mModel"
>
> modelBelief$likelihood
[1] 222.4980
>
> # force that all means' vectors are equal
> model.structure.m.E = getModelStructure(mean="E")
> str(model.structure.m.E)
List of 4
$ mean : chr "E"
$ between: chr "D"
$ within : chr "D"
$ cov : chr "D"
>
> # belief-based modeling with a specified model structure
> modelBelief = belief(X=genotypes$X, knowns=genotypes$knowns, B=genotypes$B,
+ model.structure=model.structure.m.E)
> modelBelief$likelihood
[1] 222.4980
>

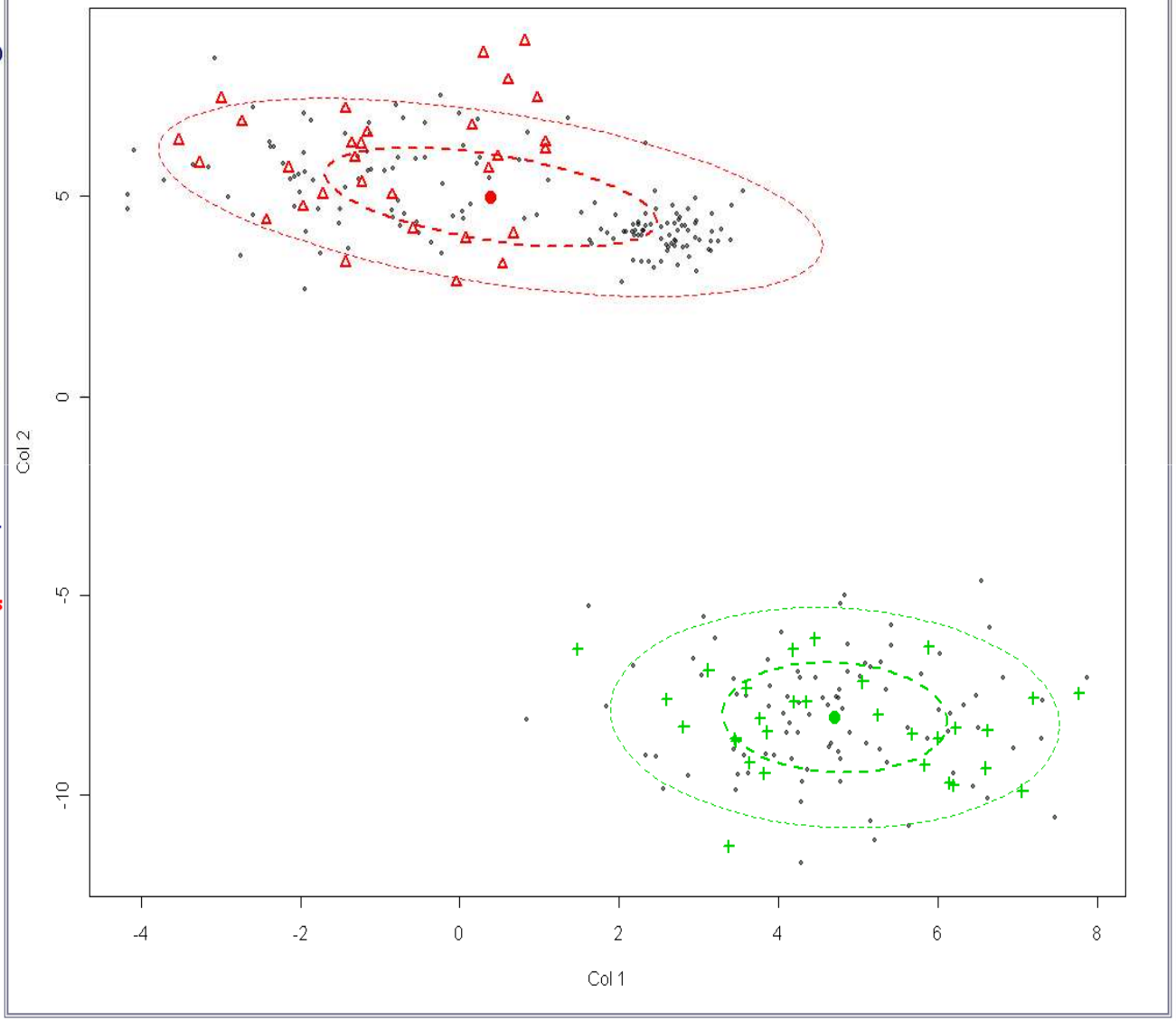
R Console

File Edit Misc Packages Windows Help

```
> set.seed(1313)
> simulated = simulateData(d=2, k=3, n=3
> str(simulated)
List of 5
 $ X           : num [1:240, 1:2] 3.42 0
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ knowns      : num [1:60, 1:2] 0.2953
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ B           : num [1:60, 1:2] 0.98 0.
 $ model.params:List of 7
  ..$ pi  : num [1:3] 0.333 0.333 0.333
  ..$ mu  : num [1:3, 1:2] -0.918 4.665
  ..$ cvar: num [1:3, 1:2, 1:2] 1.572 2.
  ..$ m   : num 60
  ..$ n   : num 300
  ..$ k   : num 3
  ..$ d   : num 2
 $ Ytrue       : int [1:300] 1 2 1 2 1 1
 - attr(*, "class")= chr "simulated"
>
> model = belief(X = simulated$X, knowns
> plot(model)
>
```

R Graphics: Device 2 (ACTIVE)

File History Resize



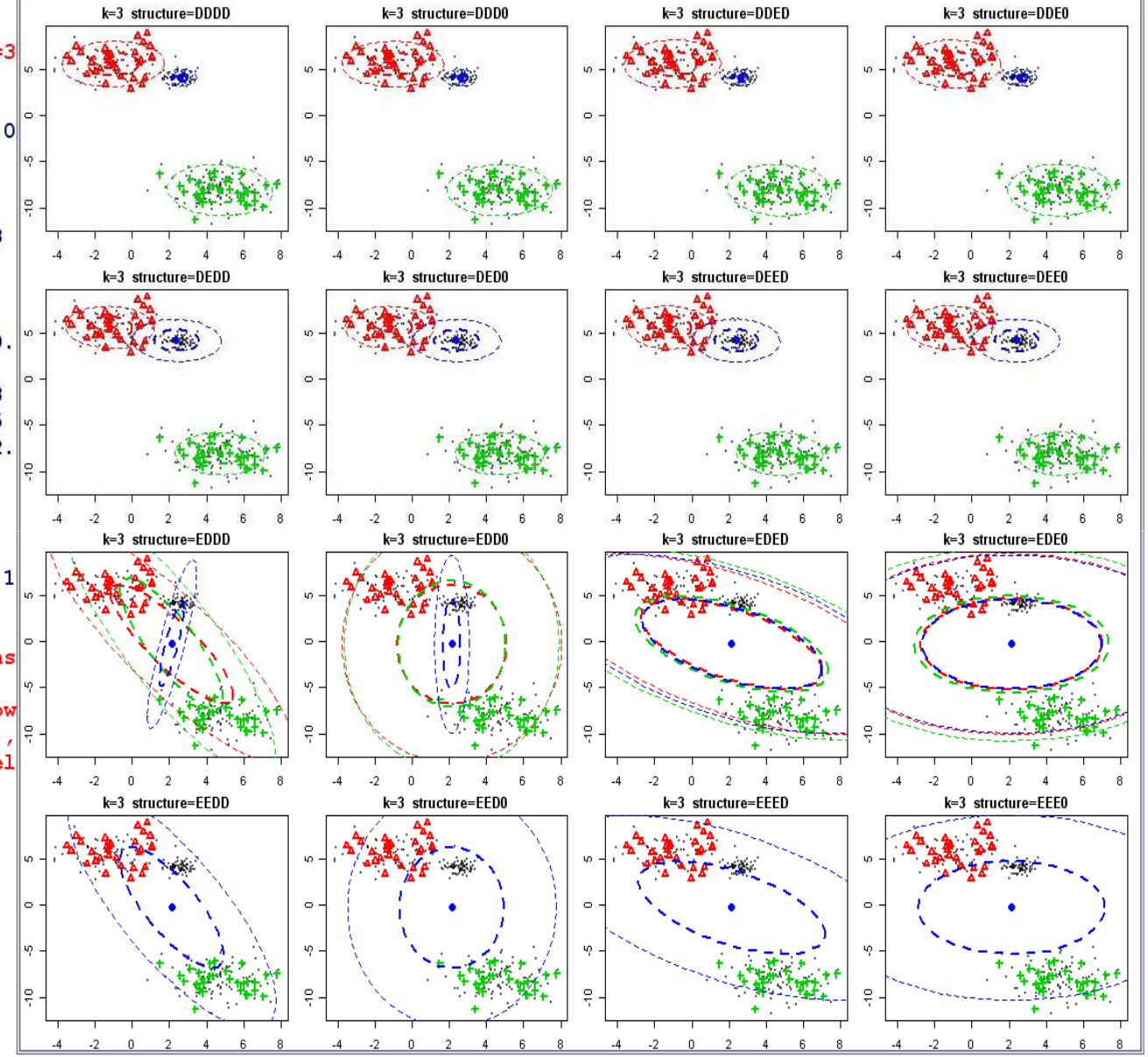
R Console

File Edit Misc Packages Windows Help

```
> set.seed(1313)
> simulated = simulateData(d=2, k=3, n=3
> str(simulated)
List of 5
 $ X            : num [1:240, 1:2] 3.42 0
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ knowns       : num [1:60, 1:2] 0.2953
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ B            : num [1:60, 1:2] 0.98 0.
 $ model.params:List of 7
  ..$ pi   : num [1:3] 0.333 0.333 0.333
  ..$ mu   : num [1:3, 1:2] -0.918 4.665
  ..$ cvar: num [1:3, 1:2, 1:2] 1.572 2.
  ..$ m    : num 60
  ..$ n    : num 300
  ..$ k    : num 3
  ..$ d    : num 2
 $ Ytrue        : int [1:300] 1 2 1 2 1 1
 - attr(*, "class")= chr "simulated"
>
> model = belief(X = simulated$X, knowns
> plot(model)
> models1=mModelList(X=simulated$X, know
+               kList=3, mean=c("D","E"),
+               cov=c("D","0"), funct=bel
>
> plot(models1)
>
```

R Graphics: Device 2 (ACTIVE)

File History Resize



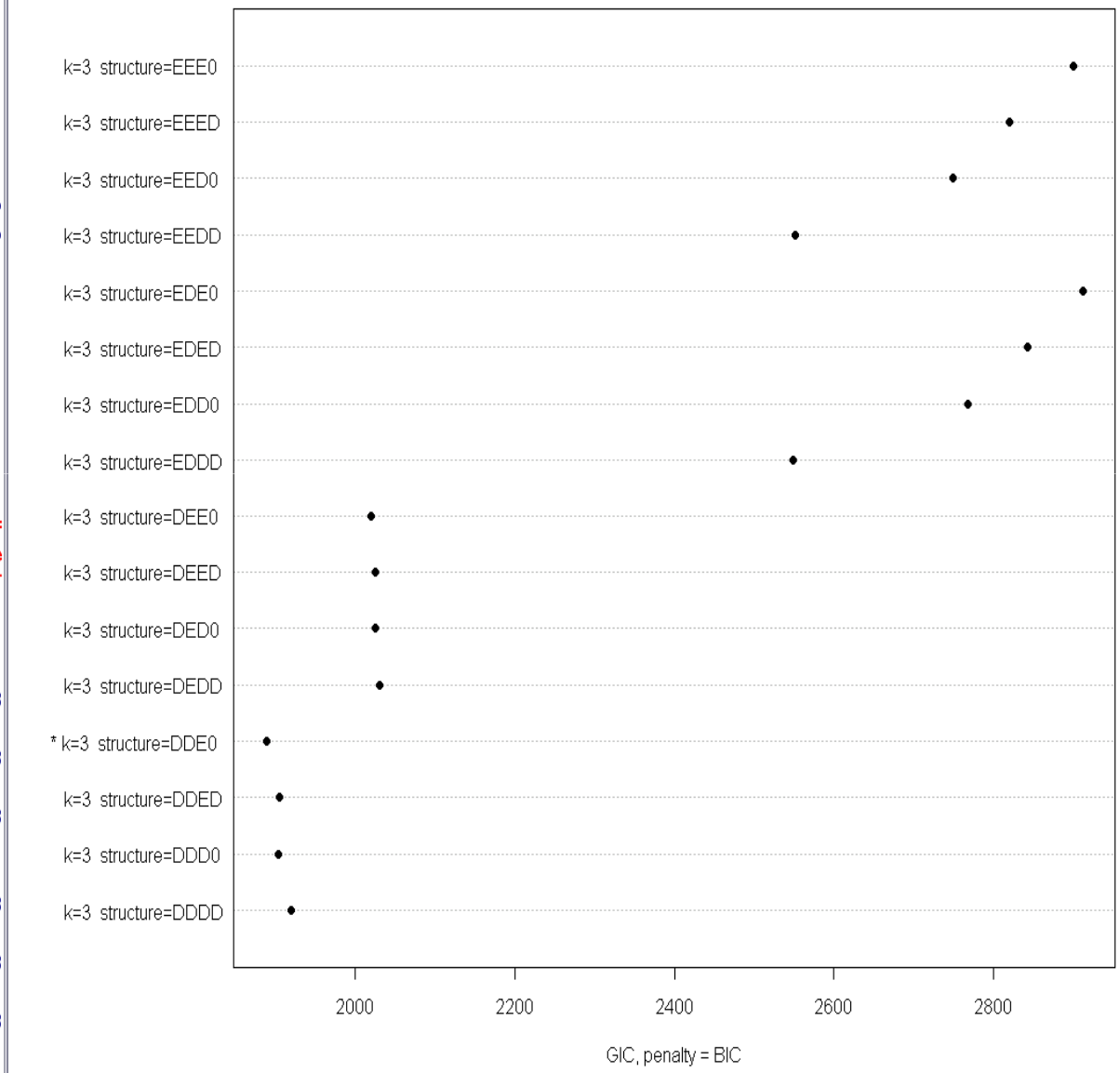
R Console

File Edit Misc Packages Windows Help

```
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ knowns       : num [1:60, 1:2] 0.2953 6.1
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ B            : num [1:60, 1:2] 0.98 0.02
 $ model.params:List of 7
  ..$ pi   : num [1:3] 0.333 0.333 0.333
  ..$ mu   : num [1:3, 1:2] -0.918 4.665 2.5
  ..$ cvar: num [1:3, 1:2, 1:2] 1.572 2.165
  ..$ m    : num 60
  ..$ n    : num 300
  ..$ k    : num 3
  ..$ d    : num 2
 $ Ytrue        : int [1:300] 1 2 1 2 1 1 2
 - attr(*, "class")= chr "simulated"
>
> model = belief(X = simulated$X, knowns =
> plot(model)
> models1=mModelList(X=simulated$X, knowns=
+              kList=3, mean=c("D","E"), be
+              cov=c("D","0"), funct=belief
>
> plot(models1)
> plotGIC(models1, penalty="BIC")
k=3  structure=DDDD k=3  structure=DDD0 k=3
           1918.678            1902.549
k=3  structure=DEED k=3  structure=DEE0 k=3
           2024.863            2019.771
k=3  structure=EEDD k=3  structure=EED0 k=3
           2551.163            2750.330
> plotGIC(models1, penalty="BIC")
k=3  structure=DDDD k=3  structure=DDD0 k=3
           1918.678            1902.549
k=3  structure=DEED k=3  structure=DEE0 k=3
           2024.863            2019.771
k=3  structure=EEDD k=3  structure=EED0 k=3
           2551.163            2750.330
>
```

R Graphics: Device 2 (ACTIVE)

File History Resize



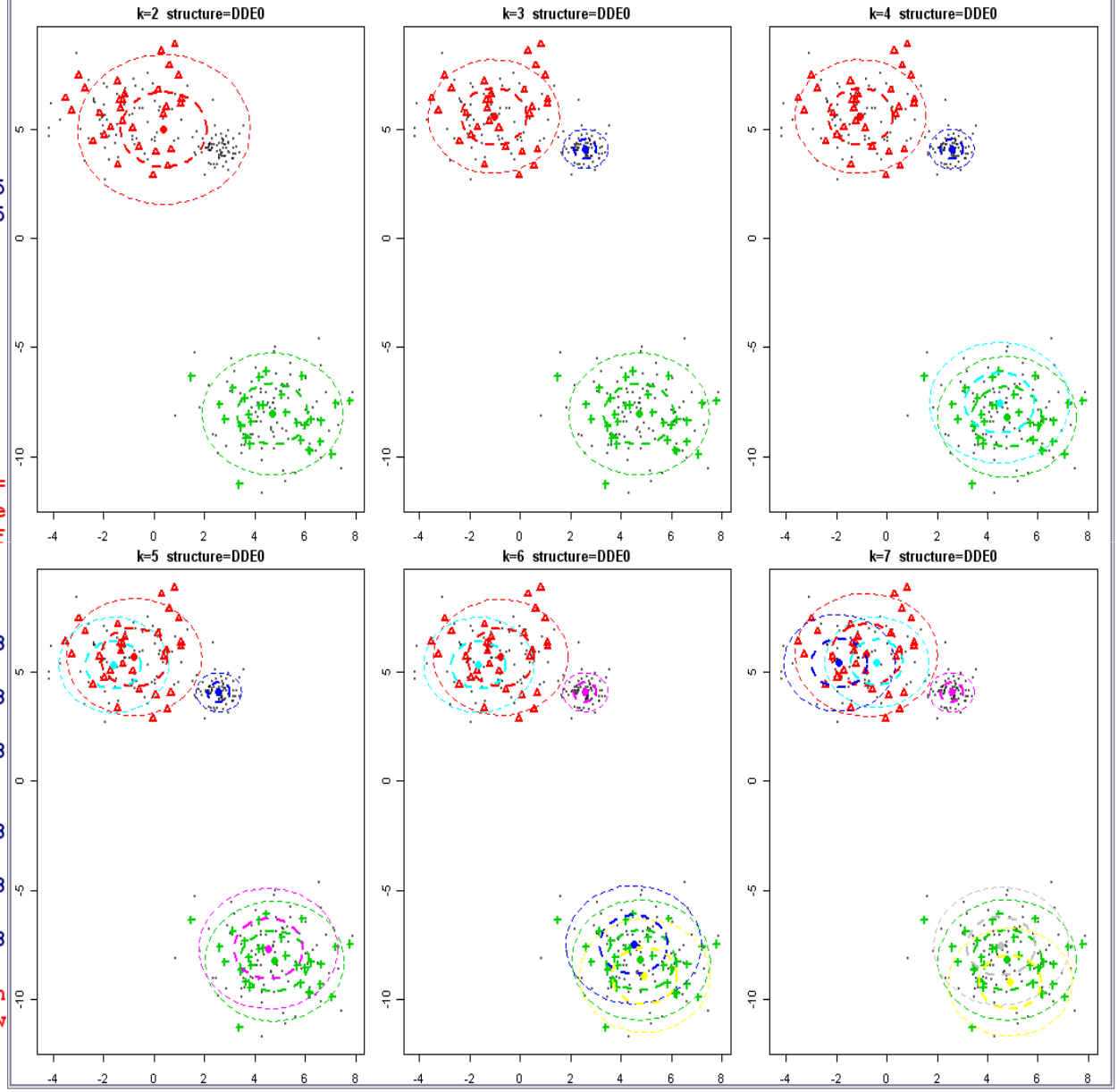
R Console

File Edit Misc Packages Windows Help

```
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "Col 1" "Col 2"
 $ B           : num [1:60, 1:2] 0.98 0.02
 $ model.params:List of 7
  ..$ pi   : num [1:3] 0.333 0.333 0.333
  ..$ mu   : num [1:3, 1:2] -0.918 4.665 2.5
  ..$ cvar: num [1:3, 1:2, 1:2] 1.572 2.165
  ..$ m    : num 60
  ..$ n    : num 300
  ..$ k    : num 3
  ..$ d    : num 2
 $ Ytrue       : int [1:300] 1 2 1 2 1 1 2
 - attr(*, "class")= chr "simulated"
>
> model = belief(X = simulated$X, knowns =
> plot(model)
> models1=mModelList(X=simulated$X, knowns=
+              kList=3, mean=c("D","E"), be
+              cov=c("D","0"), funct=belief
>
> plot(models1)
> plotGIC(models1, penalty="BIC")
k=3  structure=DDDD k=3  structure=DDD0 k=3
           1918.678            1902.549
k=3  structure=DEED k=3  structure=DEE0 k=3
           2024.863            2019.771
k=3  structure=EEDD k=3  structure=EED0 k=3
           2551.163            2750.330
> plotGIC(models1, penalty="BIC")
k=3  structure=DDDD k=3  structure=DDD0 k=3
           1918.678            1902.549
k=3  structure=DEED k=3  structure=DEE0 k=3
           2024.863            2019.771
k=3  structure=EEDD k=3  structure=EED0 k=3
           2551.163            2750.330
> models2 = beliefList(X=simulated$X, known
+                kList=2:7, mean="D", betw
> plot(models2)
>
```

R Graphics: Device 2 (ACTIVE)

File History Resize



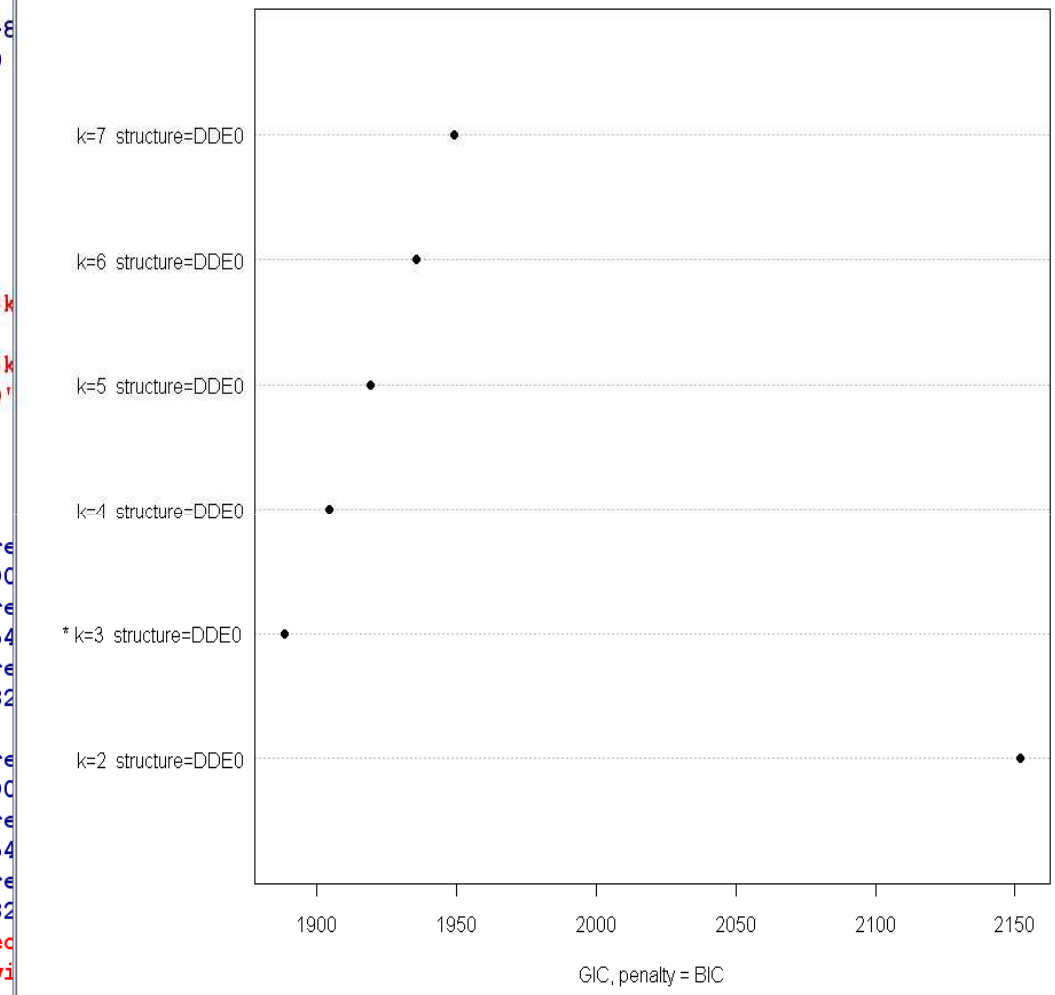
R Console

File Edit Misc Packages Windows Help

```
 $ B           : num [1:60, 1:2] 0.98 0.02 0.98 0.02 0
 $ model.params:List of 7
  ..$ pi   : num [1:3] 0.333 0.333 0.333
  ..$ mu   : num [1:3, 1:2] -0.918 4.665 2.588 5.478 -8
  ..$ cvar: num [1:3, 1:2, 1:2] 1.572 2.165 0.219 0 0
  ..$ m    : num 60
  ..$ n    : num 300
  ..$ k    : num 3
  ..$ d    : num 2
 $ Ytrue       : int [1:300] 1 2 1 2 1 1 2 2 1 2 ...
 - attr(*, "class")= chr "simulated"
>
> model = belief(X = simulated$X, knowns = simulated$k
> plot(model)
> models1=mModelList(X=simulated$X, knowns=simulated$k
+              kList=3, mean=c("D","E"), between=c("D'
+              cov=c("D","0"), funct=belief)
>
> plot(models1)
> plotGIC(models1, penalty="BIC")
k=3  structure=DDDD k=3  structure=DDD0 k=3  structure
          1918.678           1902.549           190
k=3  structure=DEED k=3  structure=DEE0 k=3  structure
          2024.863           2019.771           254
k=3  structure=EEDD k=3  structure=EED0 k=3  structure
          2551.163           2750.330           282
> plotGIC(models1, penalty="BIC")
k=3  structure=DDDD k=3  structure=DDD0 k=3  structure
          1918.678           1902.549           190
k=3  structure=DEED k=3  structure=DEE0 k=3  structure
          2024.863           2019.771           254
k=3  structure=EEDD k=3  structure=EED0 k=3  structure
          2551.163           2750.330           282
> models2 = beliefList(X=simulated$X, knowns=simulatec
+                kList=2:7, mean="D", between="D", wi
> plot(models2)
> plotGIC(models2, penalty="BIC")
k=2  structure=DDE0 k=3  structure=DDE0 k=4  structure=DDE0 k=5  structure=DDE0 k=6  structure=DDE0 k=7  structure=DDE0
          2152.105           1888.448           1904.570           1919.359           1935.556           1949.283
> |
```

R Graphics: Device 2 (ACTIVE)

File History Resize



Applications in kidney studies

In a study performed in the Clinic of Nephrology Wroclaw Medical University it was shown that expression of TIMP2 cytokine plays important role in kidney deterioration.

The TIMP2 expression is measured on a continuous scale. We find it useful to transform TIMP2 expression to a binary variable with values: low/high expression.

In the next project hosted in the same clinic we found it advisable to use information about TIMP2 level. The second study was performed on a larger number of patients. Due to various reasons the TIMP2 level was only measured only for one fifth of patients.

The partially supervised modeling was used to build a prediction rule for the TIMP2 level i.e. is used for imputations.


Thank you!