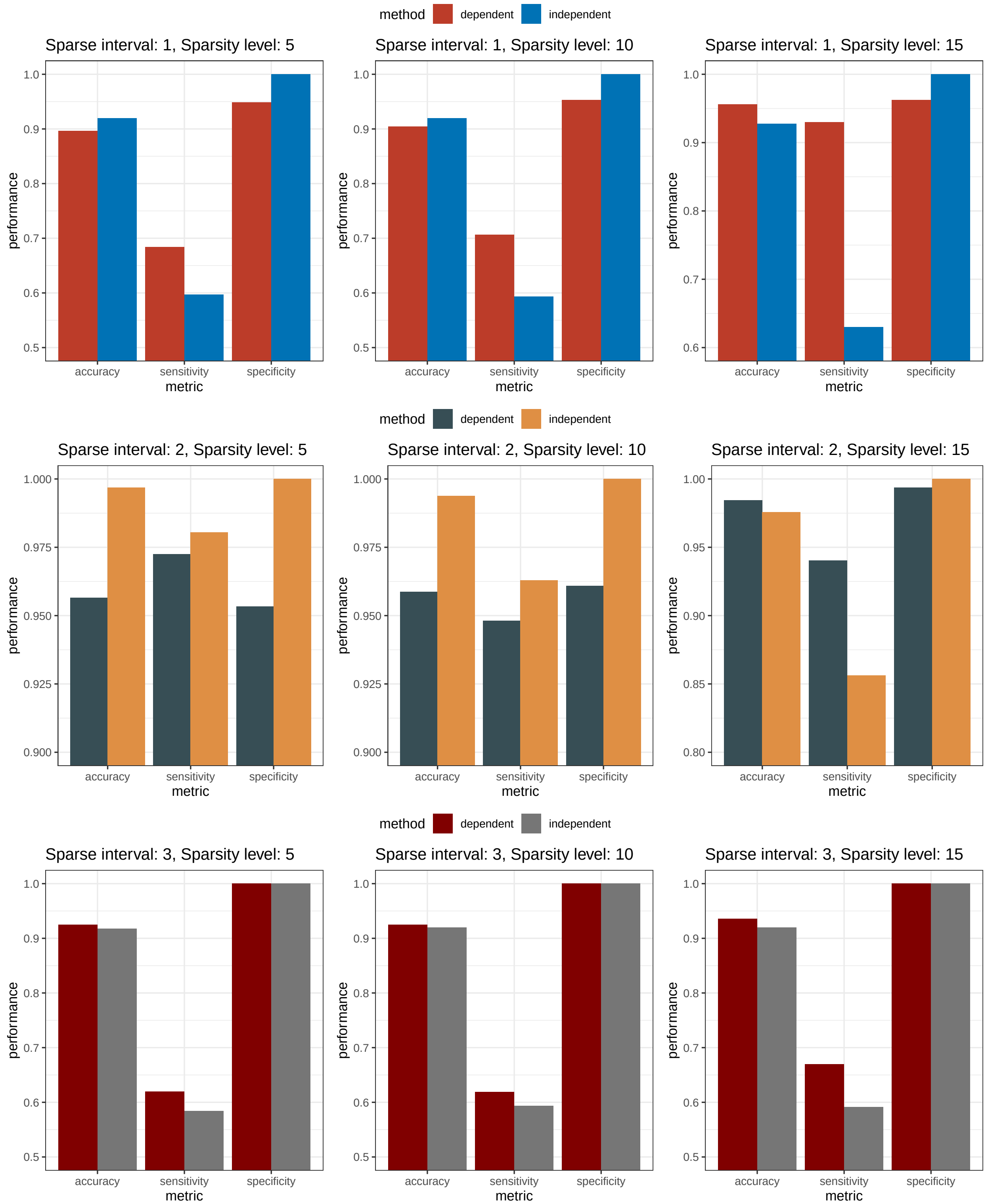
covdepGE versus competitor

Summary

I repeated this experiment 9 times. In each experiment, I chose one of the three covariate intervals to be sparse, and the other two to be non-sparse. The non-sparse intervals had 60 individuals in them, while the sparsity level of the sparse interval varied from 5, 10, or 15 individuals. In each experiment, I recorded the following metrics for both the covariate independent and covariate dependent graphical estimation methods:

- Sensitivity: the percentage of edges recovered by the method
- Specificity: the percentage of non-edges recovered by the method
- Accuracy


method dependent independent
Sparse interval: 1, Sparsity level: 5
Sparse interval: 1, Sparsity level: 10
Sparse interval: 1, Sparsity level: 15
method dependent independent
Sparse interval: 2, Sparsity level: 5
Sparse interval: 2, Sparsity level: 10
Sparse interval: 2, Sparsity level: 15
method dependent independent
Sparse interval: 3, Sparsity level: 5
Sparse interval: 3, Sparsity level: 10
Sparse interval: 3, Sparsity level: 15
performance
metric
accuracy
sensitivity
specificity

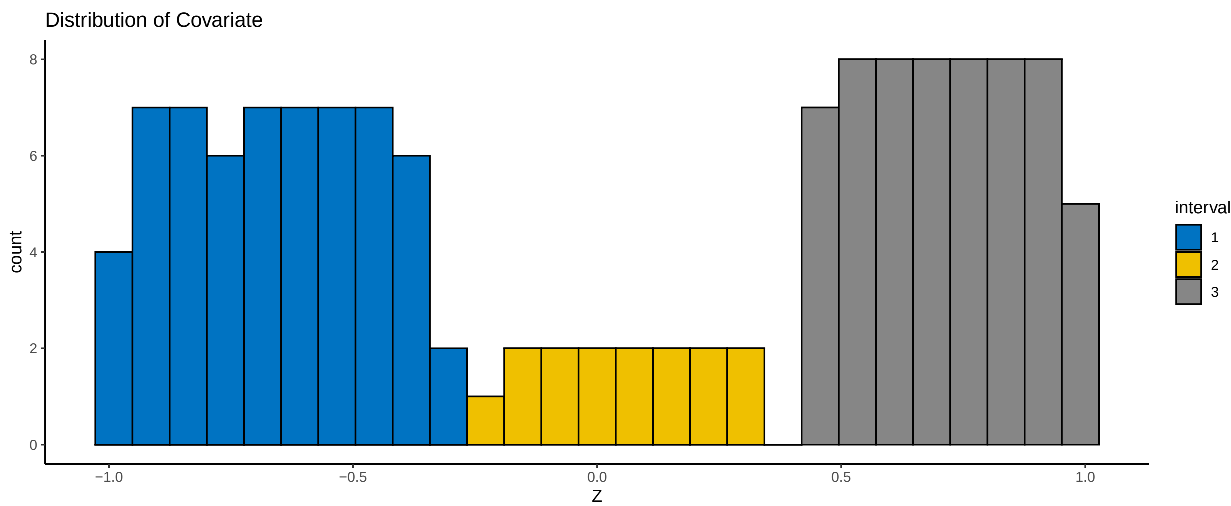
method	sparsity level	sparse interval	sensitivity	specificity	accuracy	unique graphs
dependent	5	1	0.684	0.949	0.896	3
independent	5	1	0.597	1.000	0.920	3
dependent	10	1	0.706	0.953	0.905	3
independent	10	1	0.594	1.000	0.920	3
dependent	15	1	0.930	0.962	0.956	4
independent	15	1	0.630	1.000	0.928	3
dependent	5	2	0.973	0.953	0.956	3
independent	5	2	0.980	1.000	0.997	2
dependent	10	2	0.948	0.961	0.959	4
independent	10	2	0.963	1.000	0.994	2
dependent	15	2	0.940	0.994	0.985	4
independent	15	2	0.856	1.000	0.976	3
dependent	5	3	0.619	1.000	0.924	2
independent	5	3	0.584	1.000	0.917	3
dependent	10	3	0.619	1.000	0.925	2
independent	10	3	0.594	1.000	0.920	3
dependent	15	3	0.670	1.000	0.935	3
independent	15	3	0.591	1.000	0.920	3

Interval	Individual Indices
1	$1, \ldots, 60$
2	$61, \ldots, 75$
3	$76, \ldots, 135$

Data generation

The extraneous covariate is created as the union of three disjoint intervals with nearly adjacent endpoints. Within each interval, the individuals' covariate values are equally spaced.

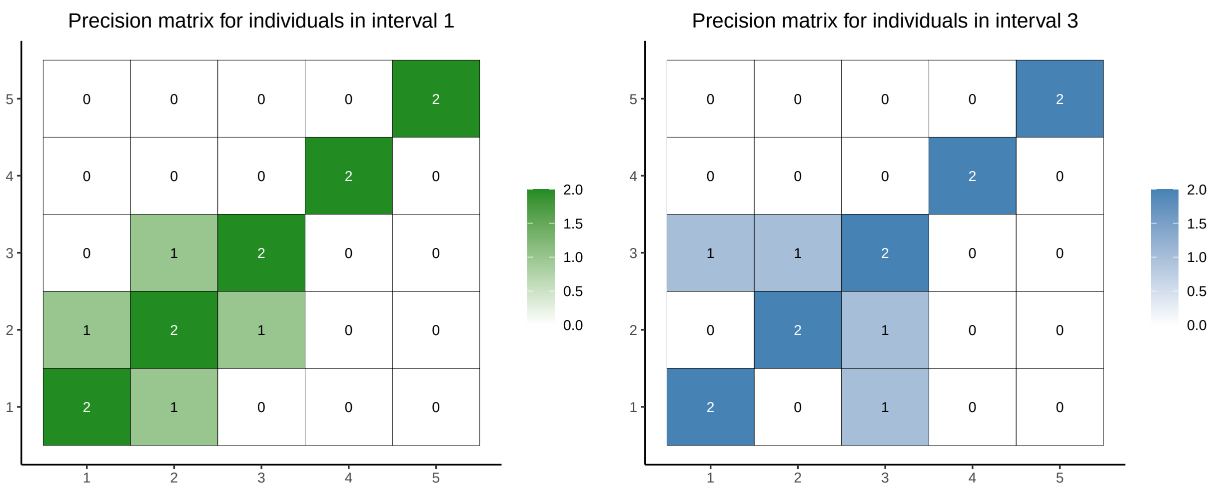
For this demonstration, the second interval is the sparse part of the covariate space, with sparsity level 10. However, in the other 8 experiments, I varied the interval in which the sparsity occured and the degree of the sparisty.

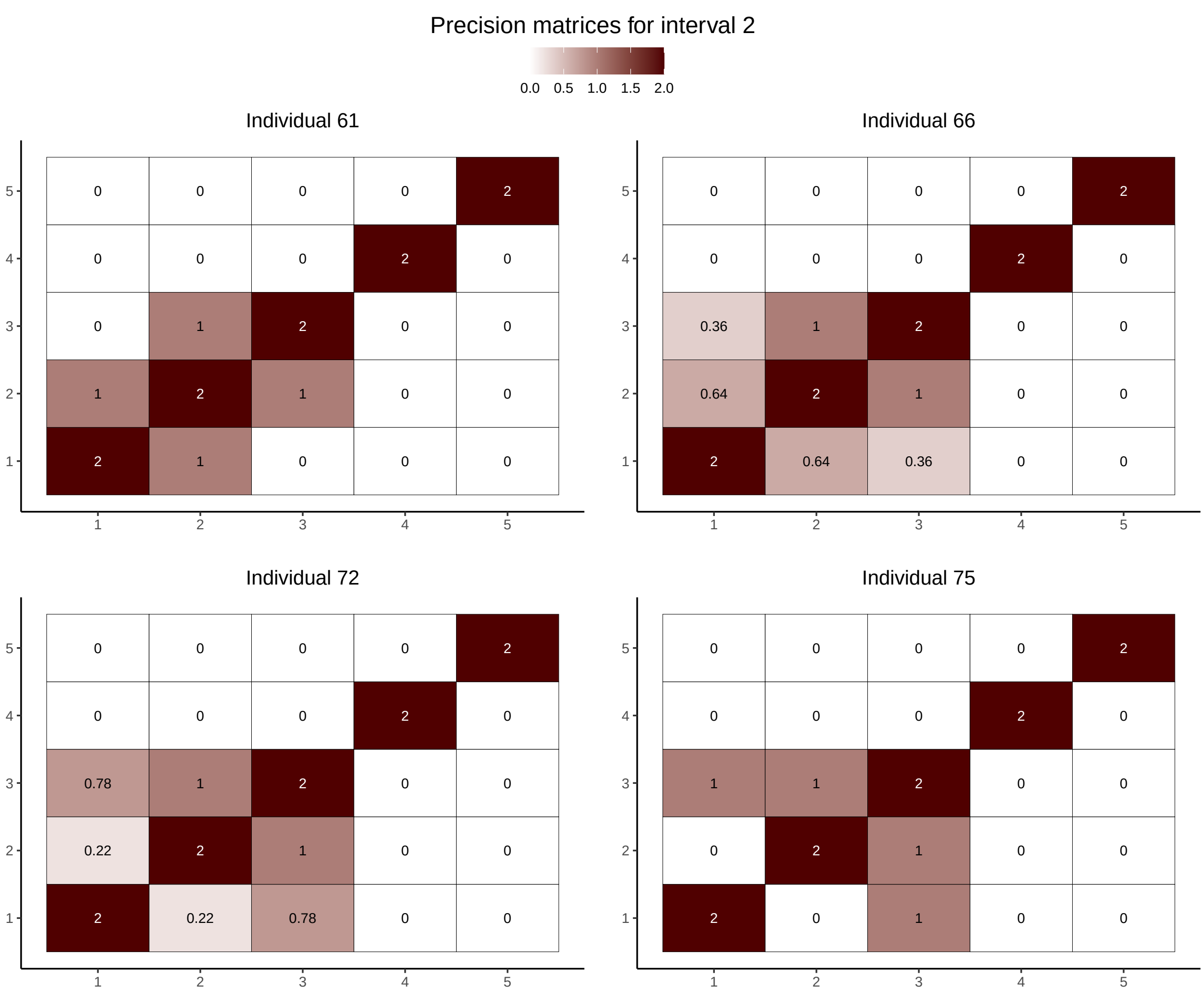


All of the individuals in interval 1 have the same precision matrix, as do all of the individuals in interval 3.

The first individual in interval 2 has the same precision matrix as those in interval 1.

As the individual index in interval 2 increases, the precision matrix continuously shifts from the precision matrix in interval 1 to the precision matrix in interval 3 such that the last individual in interval 2 has the same precision matrix as the individuals in interval 3.





After creating the precision matrix for each individual, I inverted the matrices to obtain the covariance matrices. I then used the covariance matrices to generate each observation from a 5 dimensional Gaussian distribution centered at $\vec{0}$.

Covariate dependent graph estimation

```
# use varbvs to get the hyperparameter sigma
sigmasq <- sapply(1:(p + 1), function(col_ind) mean(varbvs::varbvs(
  data_mat[, -col_ind], Z, data_mat[, col_ind], verbose = F)$sigma))
sigmasq
```

```
## [1] 0.7251686 0.6471107 0.6433086 0.4229079 0.4621893
```

```
mean(sigmasq)
```

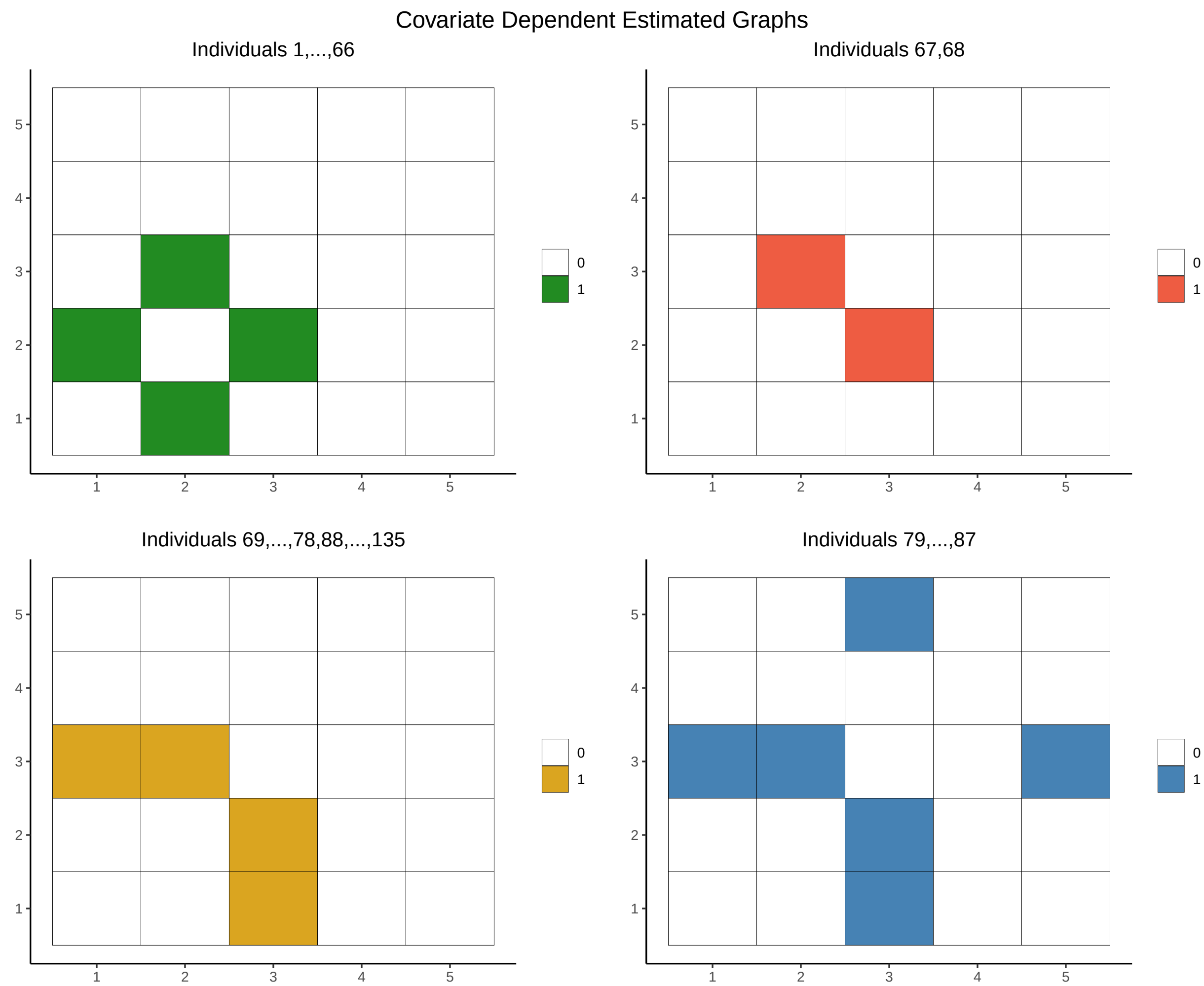
```
## [1] 0.580137
```

```
# estimate the covariance structure dependent of the covariate
out_dep <- covdepGE(data_mat,
  Z, # extraneous covariates
  sigmasq = mean(sigmasq), # hyperparameter residual variance
  var_min = 1e-3, # smallest sigmabeta_sq grid value
  var_max = 5, # largest sigmabeta_sq grid value
  n_sigma = 50, # length of the sigmabeta_sq grid
  pi_vec = 0.1, # prior inclusion probability
  tolerance = 1e-10, # variational parameter exit condition 1
  max_iter_final = 1e3 # variational parameter exit condition 2
)
```

```
## Warning in covdepGE(data_mat, Z, sigmasq = mean(sigmasq), var_min = 0.001, : For
## 2/5 variables, the selected value of sigmabeta_sq was on the grid boundary. See
## return value CAVI_details
```

```
out_dep
```

```
##                        Covariate Dependent Graphical Model
##
## Model ELBO: -44252.48              Unique conditional dependence structures: 4
## n: 135, variables: 5                       Hyperparameter grid size: 50 points
## CAVI converged for 5/5 variables
##
## Model fit completed in 3.577 secs
```



Covariate independent graph estimation

I first applied Gaussian Mixture Model clustering to the extraneous covariate. The number of clusters is automatically selected by optimizing BIC.

For all of the individuals within each of the clusters identified by GMM, I estimated the shared graph by applying `covdepGE` using a constant value for the extraneous covariate, which will result in the same estimate for all individuals within each cluster.

```
# estimate the dependence structure independent of the covariate

# apply Gaussian Mixture model clustering; selects number of clusters based on
# the model that results in the best BIC
gmm <- Mclust(Z)

# find accuracy of the clustering
fossil::rand.index(gmm$classification, as.numeric(cov_df$interval))
```

```
## [1] 0.9838585
```

```r
# find number of clusters in final clustering
(num_clusters <- length(unique(gmm$classification)))
```

```
## [1] 3
```

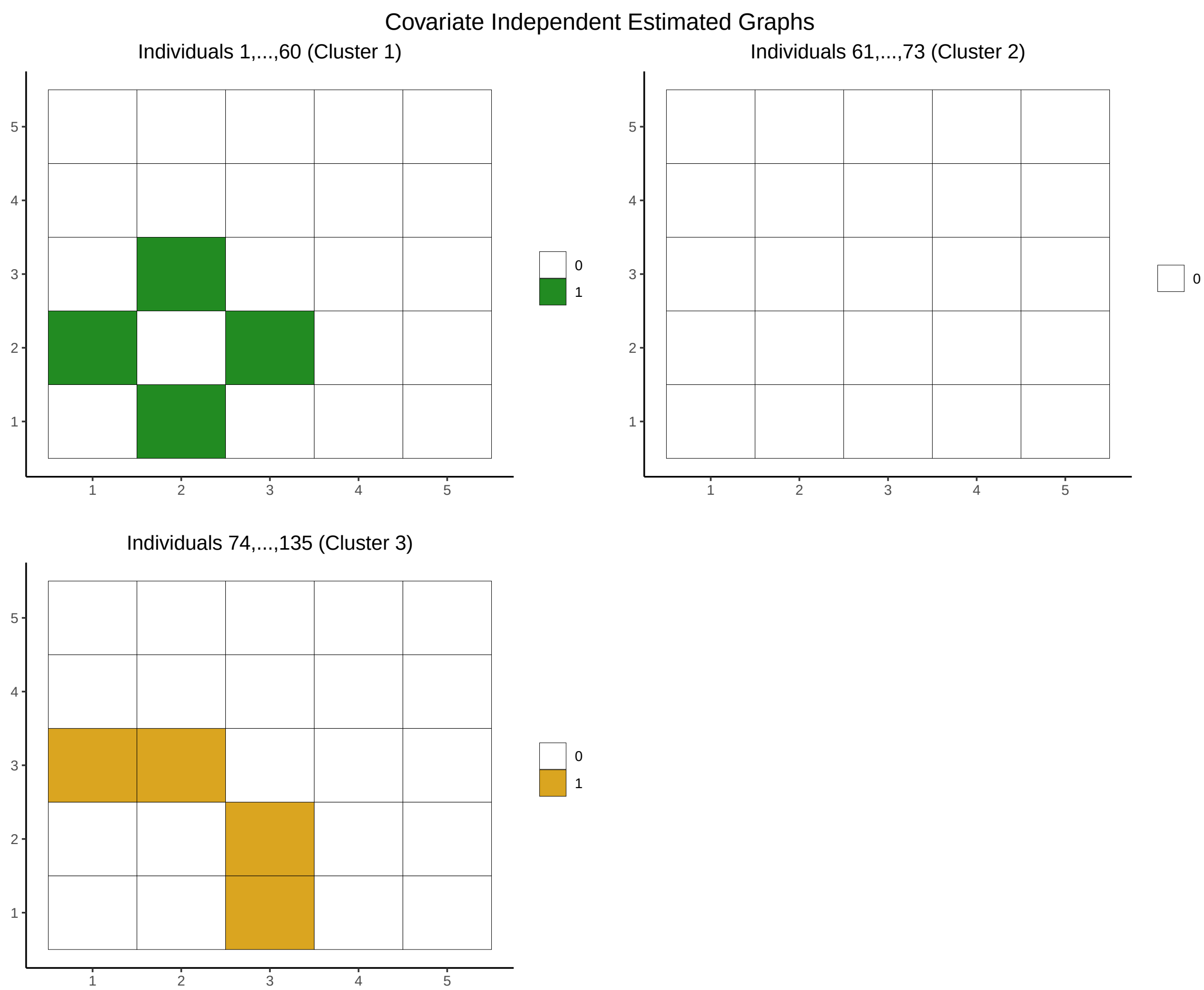
```r
out_indep <- vector("list", num_clusters)

# iterate over each of the clusters identified by GMM
for (k in 1:num_clusters) {

  # fix the datapoints in the k-th cluster
  data_mat_k <- data_mat[gmm$classification == k, ]

  # use varbvs to get the hyperparameter sigma
  sigmasq_k <- sapply(1:(p + 1), function(col_ind) mean(varbvs::varbvs(
    data_mat_k[, -col_ind], NULL, data_mat_k[, col_ind], verbose = F)$sigma))

  # apply the GGM using covdepGE with constant Z, save the resulting graph
  out_indep[[k]] <- covdepGE(data_mat_k,
    rep(0, nrow(data_mat_k)), # extraneous covariates
    sigmasq = mean(sigmasq_k), # hyperparameter residual variance
    var_min = 1e-3, # smallest sigmabeta_sq grid value
    var_max = 5, # largest sigmabeta_sq grid value
    n_sigma = 50, # length of the sigmabeta_sq grid
    pi_vec = 0.1, # prior inclusion probability
    tolerance = 1e-10, # variational parameter exit condition 1
    max_iter_final = 1e3, # variational parameter exit condition 2
    kde = F, # whether to use kde to calculate bandwidths
    scale = F # whether to scale the extraneous covariates
  )
}
```

```
## Warning in covdepGE(data_mat_k, rep(0, nrow(data_mat_k)), sigmasq =
## mean(sigmasq_k), : For 2/5 variables, the selected value of sigmabeta_sq was on
## the grid boundary. See return value CAVI_details

## Warning in covdepGE(data_mat_k, rep(0, nrow(data_mat_k)), sigmasq =
## mean(sigmasq_k), : For 1/5 variables, the selected value of sigmabeta_sq was on
## the grid boundary. See return value CAVI_details

## Warning in covdepGE(data_mat_k, rep(0, nrow(data_mat_k)), sigmasq =
## mean(sigmasq_k), : For 2/5 variables, the selected value of sigmabeta_sq was on
## the grid boundary. See return value CAVI_details
```

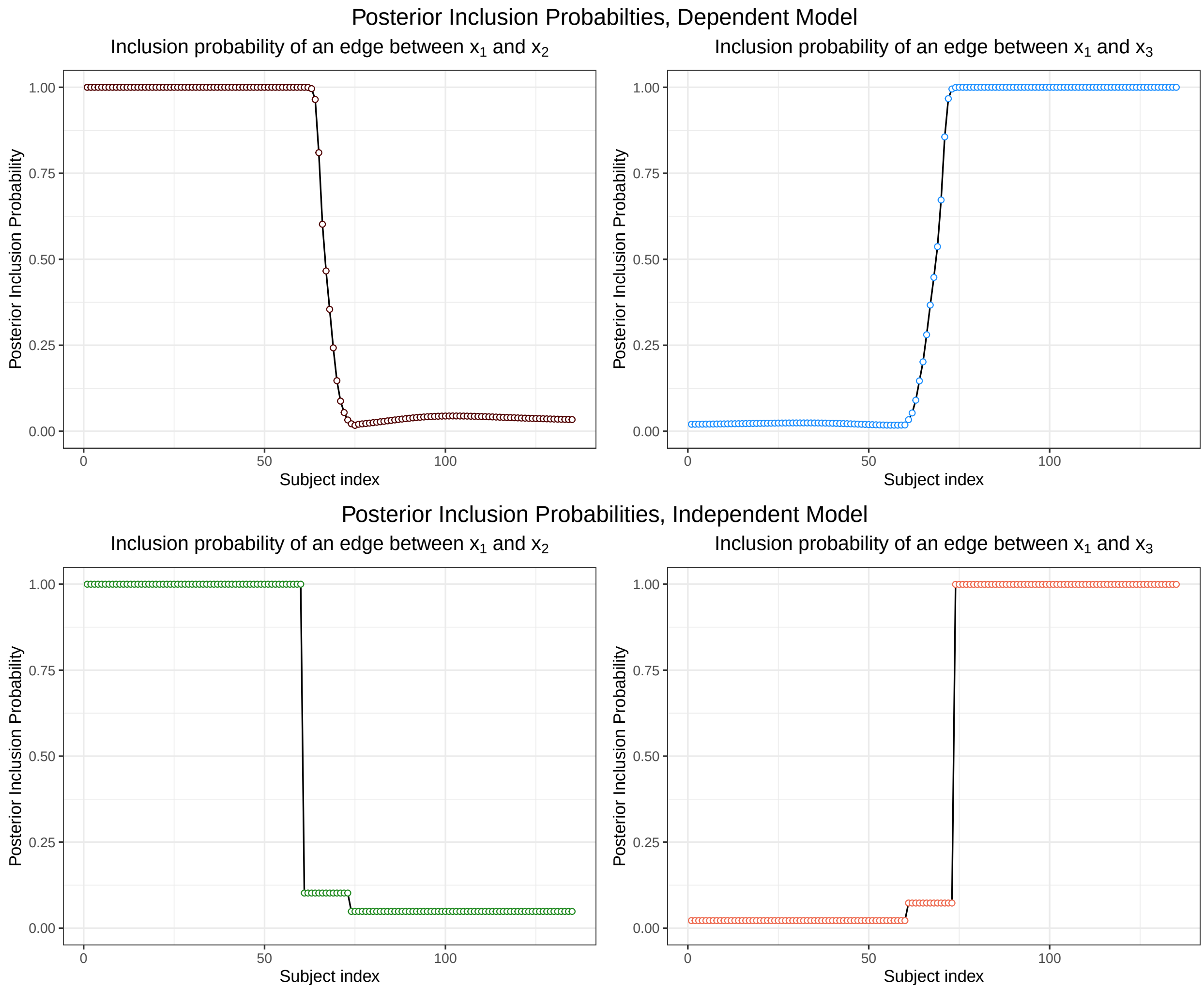
```r
out_indep
```

```
## [[1]]
##                         Covariate Dependent Graphical Model
##
## Model ELBO: -10326.26               Unique conditional dependence structures: 1
## n: 60, variables: 5                         Hyperparameter grid size: 50 points
```

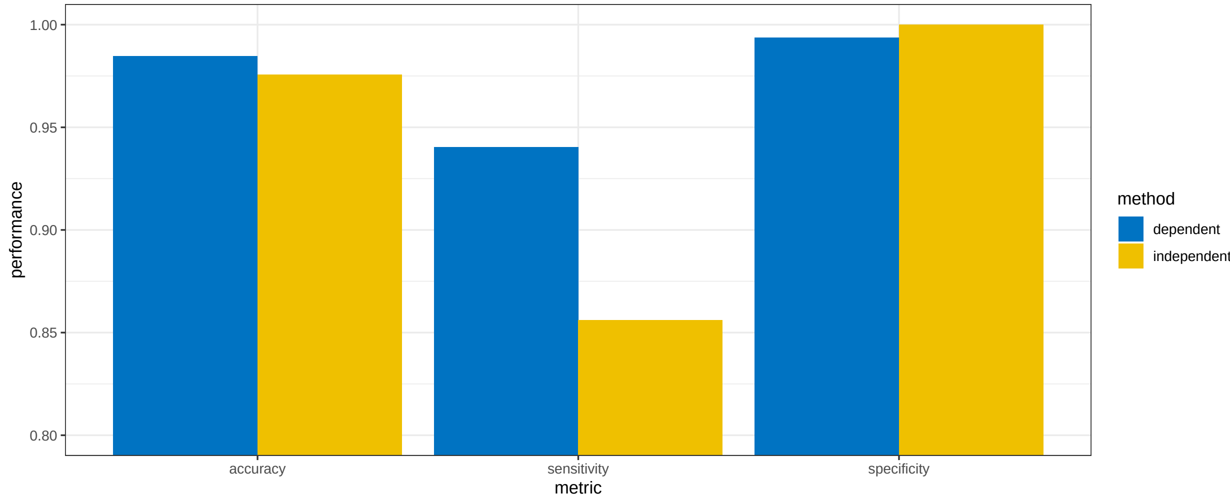
```
## CAVI converged for 5/5 variables
##
## Model fit completed in 0.708 secs
##
## [[2]]
##                       Covariate Dependent Graphical Model
##
## Model ELBO: -584.33                 Unique conditional dependence structures: 1
## n: 13, variables: 5                         Hyperparameter grid size: 50 points
## CAVI converged for 5/5 variables
##
## Model fit completed in 0.046 secs
##
## [[3]]
##                       Covariate Dependent Graphical Model
##
## Model ELBO: -11099.88               Unique conditional dependence structures: 1
## n: 62, variables: 5                         Hyperparameter grid size: 50 points
## CAVI converged for 5/5 variables
##
## Model fit completed in 0.687 secs
```



Performance Analysis



method	metric	performance
dependent	sensitivity	0.940
independent	sensitivity	0.856
dependent	specificity	0.994
independent	specificity	1.000
dependent	accuracy	0.985
independent	accuracy	0.976



ELBO Monitoring

The latest version of `covdepGE` includes `CAVI_details` in the return:

`CAVI_details`: list of $(p + 1)$ lists; the j-th list corresponds to the j-th variable and contains 6 values:

- `sigmabeta_sq`, `pi`: numerics; the grid point that maximized the ELBO for the j-th variable
- `ELBO`: numeric; the maximum value of ELBO for the final CAVI
- `converged_iter`: numeric; the number of iterations to attain convergence for the final CAVI
- `ELBO_history`: numeric vector; ELBO history by iteration for the final CAVI. If `monitor_final_elbo` is `F`, then this value will be `NULL`
- `non_converged`: numeric matrix; each row corresponds to the ELBO history for each of the grid points that did not converge. If `monitor_grid_elbo` is `F`, then the ELBO history is omitted, and only the non-convergent `sigmabeta_sq` and `pi` values are provided. If all pairs resulted in convergence, then this value is `NULL`

```
out <- covdepGE(data_mat, Z, max_iter_final = 50, tolerance = 1e-10,
                monitor_final_elbo = T, monitor_grid_elbo = T,
                monitor_period = 15)
```

```
## Warning in covdepGE(data_mat, Z, max_iter_final = 50, tolerance = 1e-10, :
## Variable 1: CAVI did not converge in 50 iterations for 2/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, max_iter_final = 50, tolerance = 1e-10, :
## Variable 2: CAVI did not converge in 50 iterations for 1/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, max_iter_final = 50, tolerance = 1e-10, : For
## 2/5 variables, the selected value of sigmabeta_sq was on the grid boundary. See
## return value CAVI_details
```

```r
out
```

```
##                        Covariate Dependent Graphical Model
##
## Model ELBO: -50826.25               Unique conditional dependence structures: 4
## n: 135, variables: 5                         Hyperparameter grid size: 8 points
## CAVI converged for 5/5 variables
##
## Model fit completed in 1.072 secs
```

```r
out$CAVI_details
```

```
## [[1]]
## [[1]]$sigmabeta_sq
## [1] 0.5179475
##
## [[1]]$pi
## [1] 0.1
##
## [[1]]$ELBO
## [1] -10060.66
##
## [[1]]$converged_iter
## [1] 44
##
## [[1]]$ELBO_history
##              1        16        31        44
## ELBO -12190.44 -10060.66 -10060.66 -10060.66
##
## [[1]]$non_converged
##                                1        16        31        46        50
## slab var: 0.193, pi: 0.1 -12045.89 -10078.65 -10078.65 -10078.65 -10078.65
## slab var: 0.072, pi: 0.1 -11879.73 -10228.03 -10228.03 -10228.03 -10228.03
##
##
## [[2]]
## [[2]]$sigmabeta_sq
## [1] 0.5179475
##
## [[2]]$pi
## [1] 0.1
##
## [[2]]$ELBO
## [1] -11634.56
##
## [[2]]$converged_iter
## [1] 31
##
## [[2]]$ELBO_history
##              1        16        31
## ELBO -13678.93 -11634.56 -11634.56
##
## [[2]]$non_converged
```

```
##                                    1        16        31        46        50
## slab var: 0.027, pi: 0.1 -13850.06 -12839.07 -12839.07 -12839.07 -12839.07
##
##
## [[3]]
## [[3]]$sigmabeta_sq
## [1] 0.5179475
##
## [[3]]$pi
## [1] 0.1
##
## [[3]]$ELBO
## [1] -12533.35
##
## [[3]]$converged_iter
## [1] 30
##
## [[3]]$ELBO_history
##              1        16        30
## ELBO -14755.66 -12533.35 -12533.35
##
## [[3]]$non_converged
## NULL
##
##
## [[4]]
## [[4]]$sigmabeta_sq
## [1] 0.01
##
## [[4]]$pi
## [1] 0.1
##
## [[4]]$ELBO
## [1] -7792.318
##
## [[4]]$converged_iter
## [1] 8
##
## [[4]]$ELBO_history
##              1         8
## ELBO -7792.321 -7792.318
##
## [[4]]$non_converged
## NULL
##
##
## [[5]]
## [[5]]$sigmabeta_sq
## [1] 0.01
##
## [[5]]$pi
## [1] 0.1
##
## [[5]]$ELBO
```

```
## [1] -8805.357
##
## [[5]]$converged_iter
## [1] 11
##
## [[5]]$ELBO_history
##              1         11
## ELBO -8805.431 -8805.357
##
## [[5]]$non_converged
## NULL
```