Revealing the Personal Effects of Nutrition with Mixed-Effect Bayesian Network

Jari Turkia, jari.turkia@cgi.com
University of Eastern Finland

Abstract

This notebook describes the implementation of the modelling method and the experimental analysis that is described in the main article.

In this work we propose a Bayesian network as an appealling way to model and predict the effects of nutrition. The Bayesian network is a directed graphical model where nodes of the graph are random variables and the edges between the nodes indicate an effect between variables. In the nutritional modelling we assign the amount of nutrients at the person's diet and the indicators of person's well-being those as random variables, and then we study the effects between them. The goal is then to find the connections for the graph that most probably describes the correct conditional relationships between nutrients and their effects.

In this article we use this method to analyze a dataset from the Sysdimet study (Lankinen et al. 2011) that contains repeated measurements of 17 nutrients, some basic information about patients (gender, medication) and blood test results. To ease the graph search, we focus only on two-level, bipartite, graphs where nutrients and personal details are assumed to affect blood tests, and only the magnitude of effect is left to be estimated. As we are interested in the personal variations of these nutritional effects we use a mixed-effect parametrization of Bayesian network (Bae et al. 2016). This allows us to estimate both typical and personal magnitudes of the nutritional effects with a same model.

Formal definition of the problem. Let us denote the graph of interconnected nutrients and responses with G. We can then formulate the modeling problem as finding the graph G that is the most probable given the data D

$$P(G|D) \tag{1}$$

By using the Bayes' Rule we can be split this probability into proportions of the data likelihood of the given graph and any prior information we might have about suitable graphs

$$P(G|D) \propto P(D|G)P(G) \tag{2}$$

This turns the problem to finding a graph that is most probable given data. To enforce our assumption of biologically plausible graphs, we set the graph prior $P(G)$ to zero for all other graphs than the previously descibed bipartite graphs with nutrients affecting the bodily responses.

The joint probability of the graph $P(G|D)$ can furthermore factorized into separate local probability distributions (Bae et al. 2016, Koller and Friedman (2009)) by their *Markov blankets*. These probability distributions can be estimated separately and the novelty of our approach lies in their hierarchical mixed-effect estimation.

Hierarchical estimation of the local distributions. We are interested in the personal variations of how different nutrients affect. For this we need repeated measurements of nutrients in person's diet and the corresponding bodily responses. By collecting these personal sets of measurements we obtain a dataset from where we can estimate both typical and personal effects of nutrition. Modeling correctly this hierarchical structure of data should give us the best fitting model and also the parameters that explain the magnitudes of effects.

Concretely, we can use generalized linear mixed-effect models to describe the local probability distributions of the Bayesian network. We assume that every random variable at the network can be explained with a probability distribution from an exponential family of distributions with linear link function with input from its parent nodes. We furthermore assume that the effect between variables can be factorized into typical and personal parts. Here typical part of the effect is denoted with coefficient β and personal part as b

$$P(Y_i|pa(Y_i), \phi_i, G_i) = EXPFAM(Y_i|pa(X_i)\beta_i + pa_Z(X_i)b_i, \sigma^2) \tag{3}$$

Note that while this formulation generalizes to graphs with multiple levels, we have a prior assumption (${P}({G})$) that the structure of our dataset is a bipartite graph.
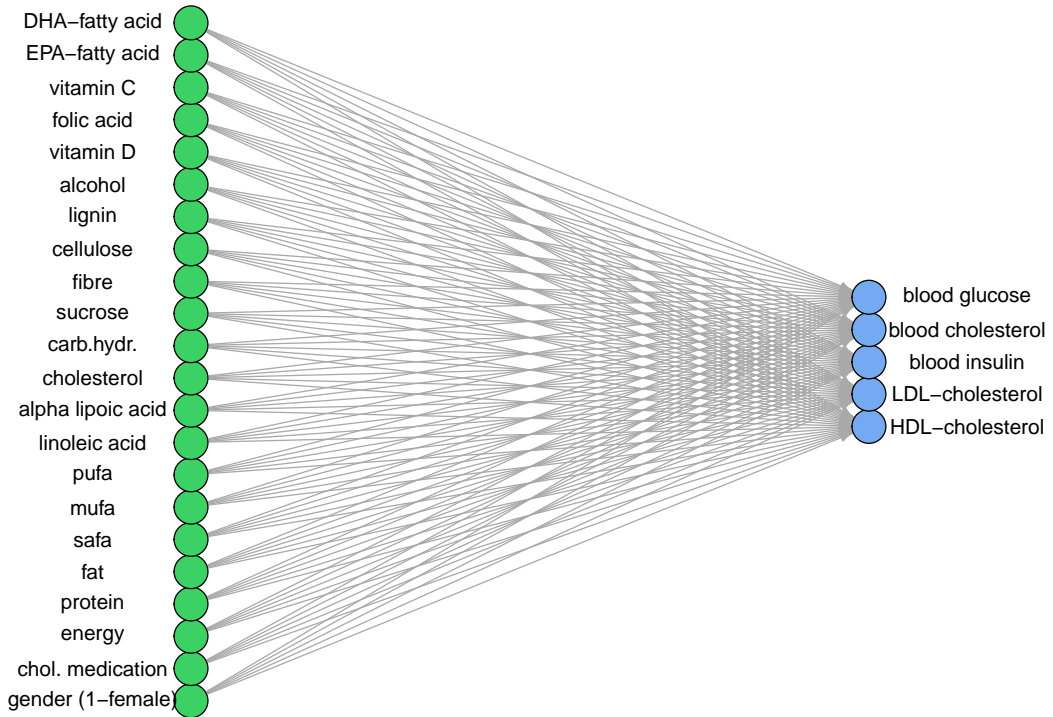

We assume a connection from every nutrient and personal information to every blood test value at the dataset. The model search is then focused on to estimating optimal coefficients β and b that we use to define the strength of the connection. It is also important to find correct probability distributions for the random variables.

Our starting assumption is that both nutrients and the blood test values are normally distributed. This assumption is naive as it allows the blood tests to have negative values. We normalize all the input values to same scale, so that the estimated regression coefficients can be used as a indicators of connection strenght.

```
sysdimet_normal <- mebn.bipartite_model(reaction_graph = initial_graph,
                                        inputdata = sysdimet,
                                        predictor_columns = assumedpredictors,
                                        assumed_targets = assumedtargets,
                                        group_column = "SUBJECT_ID",
                                        local_estimation = mebn.sampling,
                                        local_model_cache = "models/BLMM_normal",
                                        stan_model_file = "mebn/BLMM_normal.stan",
                                        normalize_values = TRUE)
```

We can assess the model fit by plotting the posterior predictive distributions (PPC) over the distributions of the true blood test values. As expected, the Gaussian random variables are not fitting well.

fshdl
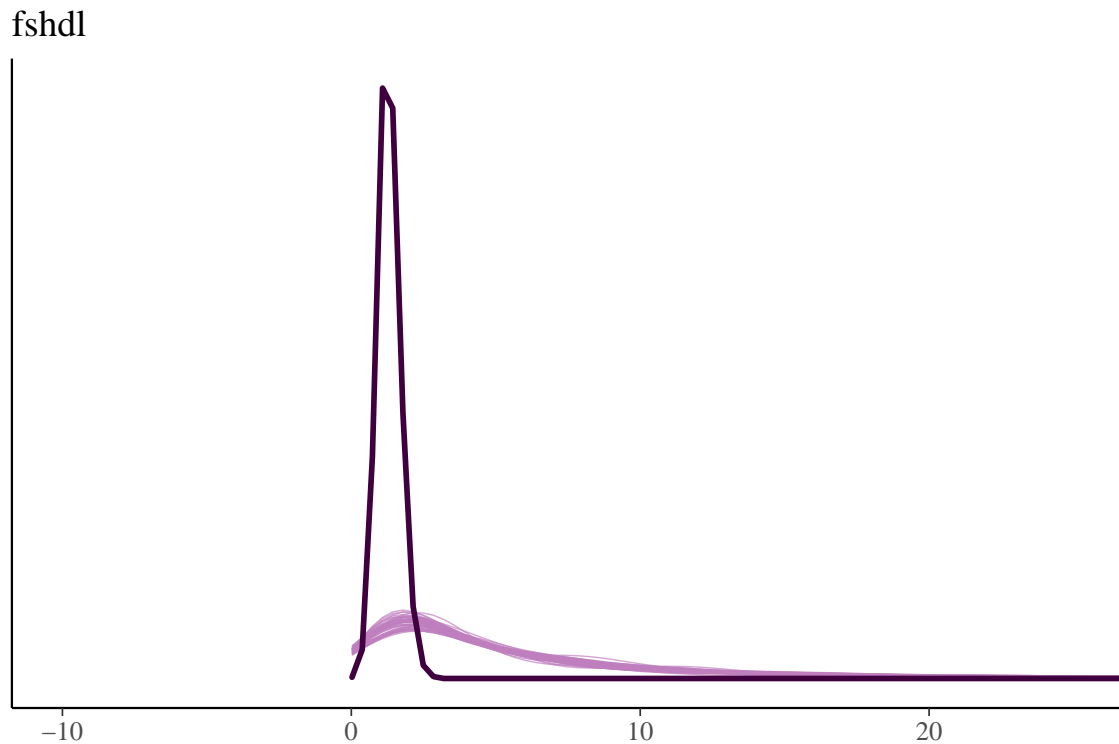
Developing the model

- Biometric data is usually more multiplicative than additive. This is why log-normal distribution makes sense. Log-normal / Gamma

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0021403 Citation: Limpert E, Stahel WA (2011) Problems with Using the Normal Distribution – and Ways to Improve Quality and Efficiency of Data Analysis. PLoS ONE 6(7): e21403. https://doi.org/10.1371/journal.pone.0021403

- We use identity link as it keeps the regression coefficients on the same absolute scale than with Gaussian models

$$P(Y_i|pa(Y_i), \phi_i, G_i) = Gamma(Y_i|\alpha, \alpha/pa(X_i)\beta_i + pa_Z(X_i)b_i) \qquad (4)$$

```
initial_gamma_graph <- mebn.new_graph_with_randomvariables(datadesc)
```

```
sysdimet_gamma <- mebn.bipartite_model(reaction_graph = initial_gamma_graph,
                                       inputdata = sysdimet,
                                       predictor_columns = assumedpredictors,
                                       assumed_targets = assumedtargets,
                                       group_column = "SUBJECT_ID",
                                       local_estimation = mebn.sampling,
                                       local_model_cache = "models/BLMM_gamma/hierarchical_idlink",
                                       stan_model_file = "mebn/BLMM_gamma_hierarchical.stan",
                                       normalize_values = TRUE)
```

Checking the fit of the local models
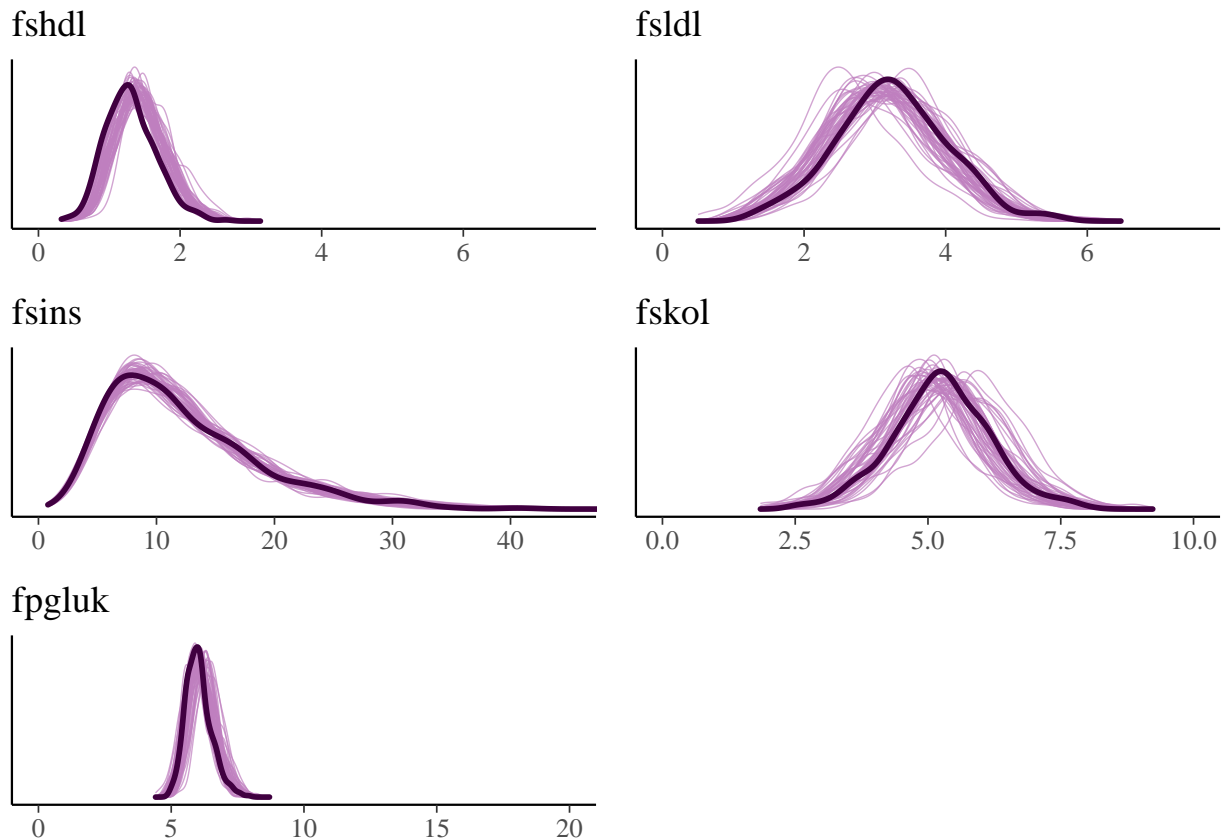
As we see here, the regression coefficients are approximately the same for both Normal and Gamma models, but the confidence is better for the Gamma model

```
beta_coefs <- mebn.compare_typicals(sysdimet_normal, sysdimet_gamma)
head(beta_coefs)
```

```
##                                 effect    model1    model2
## 1 gender (1-female) -> HDL-cholesterol  0.27092   0.29188
## 2   chol. medication -> HDL-cholesterol -0.06641  -0.06707
## 3             energy -> HDL-cholesterol -0.05982  -0.29840
## 4            protein -> HDL-cholesterol  0.01686   0.03121
## 5                fat -> HDL-cholesterol -0.24538   0.19198
## 6               safa -> HDL-cholesterol  0.14914  -0.01356
```

Comparing different local models with LOO

Besides the visual inspection, we can also compare the models using LOO-PSIS information criteria. It uses log probability that is calculated as part of the Stan models.

```
normal_vs_gamma <- mebn.LOO_comparison(assumedtargets, "BLMM_normal", "BLMM_gamma/hierarchical_idlink")
normal_vs_gamma
```

```
##   distribution        BLMM_normal BLMM_gamma/hierarchical_idlink
## 1        fshdl    -801.238 (3.928)                -895.79 (64.318)
## 2        fsldl   -3149.55 (45.452)              -2039.735 (141.853)
## 3        fsins  -1272.615 (25.446)               -1063.643 (15.302)
## 4        fskol  -5469.744 (66.412)                -974.159 (50.588)
## 5       fpgluk -27242.896 (167.91)                -469.038 (27.527)
```

```
# https://haozhu233.github.io/kableExtra/awesome_table_in_html.html
```
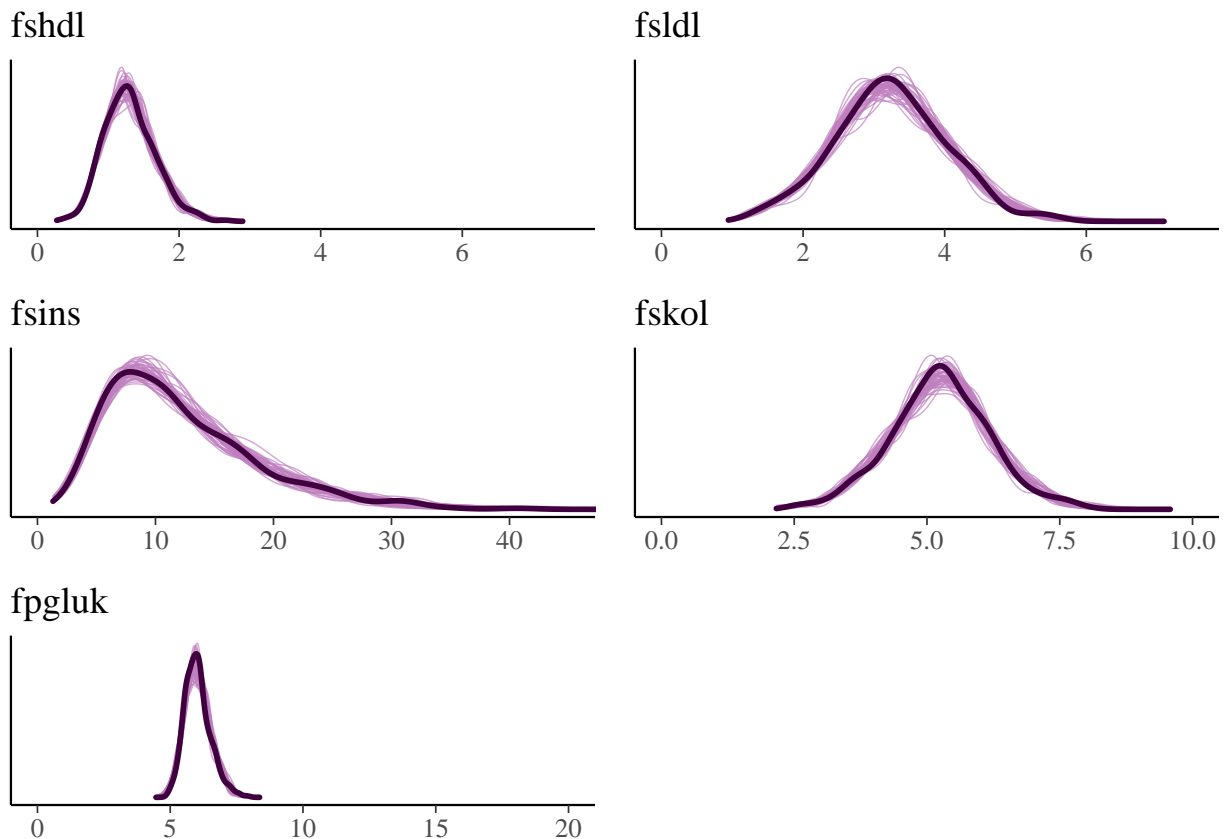
(Is P(D|G_i) = SUM(ELPD_i)?)

As the expected log predictive density (ELPD) of gamma model is higher, we should choose that for further development.

Varying response time: AR-structures

```
...
    // - calculate residuals
    e[n] = Y[n] - mu[n];

    // - estimate autoregression coefficient for observations of one patient
    if (group_size > 1)
      mu[n] += e[n-1] * ar1;

...
```

```
ar0_vs_ar1 <- mebn.LOO_comparison(assumedtargets, "BLMM_gamma/hierarchical_idlink", "BLMM_gamma/ar1")
ar0_vs_ar1
```

```
##   distribution BLMM_gamma/hierarchical_idlink       BLMM_gamma/ar1
## 1        fshdl                   -895.79 (64.318)   204.102 (16.277)
## 2        fsldl               -2039.735 (141.853)  -171.428 (16.463)
## 3        fsins                -1063.643 (15.302) -1070.504 (15.523)
## 4        fskol                 -974.159 (50.588)  -308.398 (16.388)
## 5       fpgluk                 -469.038 (27.527)    -186.111 (19.1)
```

```
ar1_comparison <- mebn.AR_comparison(assumedtargets, "BLMM_gamma/ar1")
ar1_comparison
```

```
##   distribution    AR(1)   CI-10%  CI-90%
## 1        fshdl  0.08869 -0.07473 0.25418
## 2        fsldl  0.00874 -0.15134 0.17924
## 3        fsins -0.00968 -0.15569 0.13973
## 4        fskol  0.12575 -0.04315 0.30844
## 5       fpgluk  0.10030 -0.03282 0.24079
```

Typical effects of nutrition

We have now reached a reasonable good fit for the local distributions of the Bayesian network. Our code has extracted a graphical model from these full posterior distributions of random variables and parameters. The graph consists mean values of the posteriors and also their credible intervals. This is a lighter data structure for further analysis and allows also graph operations besides the regression modeling.

Besides the random variables, the graph also includes the regression coefficients β and b that denote the typical and personal magnitudes of effects for different nutrients. For better overview, let us plot the graph of typical nutritional effects. This visualization shows typical coefficients β as weight of the connection between blood test and nutrients at diet affecting it. For clarity, the visualization only shows 15 principal components explaining the variance of each blood test.

```
mebn.plot_typical_effects(sysdimet_gamma_ar1, 15)
```
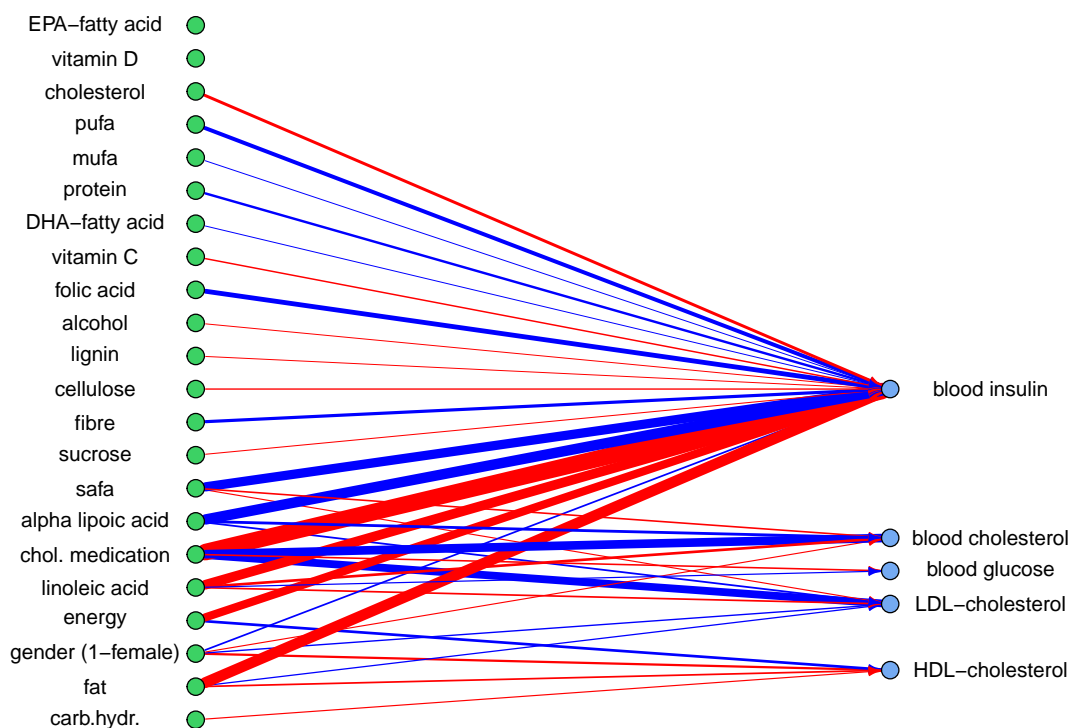

Personal effects

Beside the typical effects, our main interest lies on the personal effects b and their variances σ_b. If there is a large variance for the personal effect it means that there exists personal differences in reaction types. These differences are our main interest to find.

In this dataset there exist 20*5=100 possible effects between nutrients and blood tests. For finding the most interesting effects with most personal variance, we extract and sort b_sigma-hyperparameters from the graph.

```
##                                   effect variance  CI-10%  CI-90%
## 47              energy -> blood insulin  3.88391 1.39840 6.25608
## 45   gender (1-female) -> blood insulin  3.37135 1.89307 4.74329
## 46     chol. medication -> blood insulin  1.59519 0.26960 3.14330
```

```
51   mufa -> blood insulin 0.70719 0.13930 1.31536
61   alcohol -> blood insulin 0.70027 0.15894 1.21128
52   pufa -> blood insulin 0.68871 0.12248 1.30574
50   safa -> blood insulin 0.58117 0.09570 1.19980
67 gender (1-female) -> blood cholesterol 0.50325 0.28243 0.70513
53   linoleic acid -> blood insulin 0.50259 0.08422 0.99303
57   sucrose -> blood insulin 0.48189 0.07653 0.96654
48   protein -> blood insulin 0.45802 0.06947 0.94827
23   gender (1-female) -> LDL-cholesterol 0.44292 0.27573 0.60618
60   lignin -> blood insulin 0.44076 0.07831 0.87382
63   folic acid -> blood insulin 0.43577 0.07226 0.88768
66   DHA-fatty acid -> blood insulin 0.42098 0.07120 0.84157
65   EPA-fatty acid -> blood insulin 0.40134 0.06641 0.79980
49   fat -> blood insulin 0.38449 0.06806 0.77410
62   vitamin D -> blood insulin 0.37727 0.05959 0.76792
68 chol. medication -> blood cholesterol 0.36859 0.08022 0.66050
59   cellulose -> blood insulin 0.34831 0.05924 0.69903
```

Let us the examine closer those of the effects that have most probable variance over 0.30

```
effects_with_most_variance <- ordered_personal_variance[ordered_personal_variance$variance >= 0.30,]
number_of_varying_effects <- nrow(effects_with_most_variance)
```

For better undestanding, we can also visualize these personal variances again as a graph

```
mebn.plot_personal_variations(sysdimet_gamma_ar1, number_of_varying_effects)
```
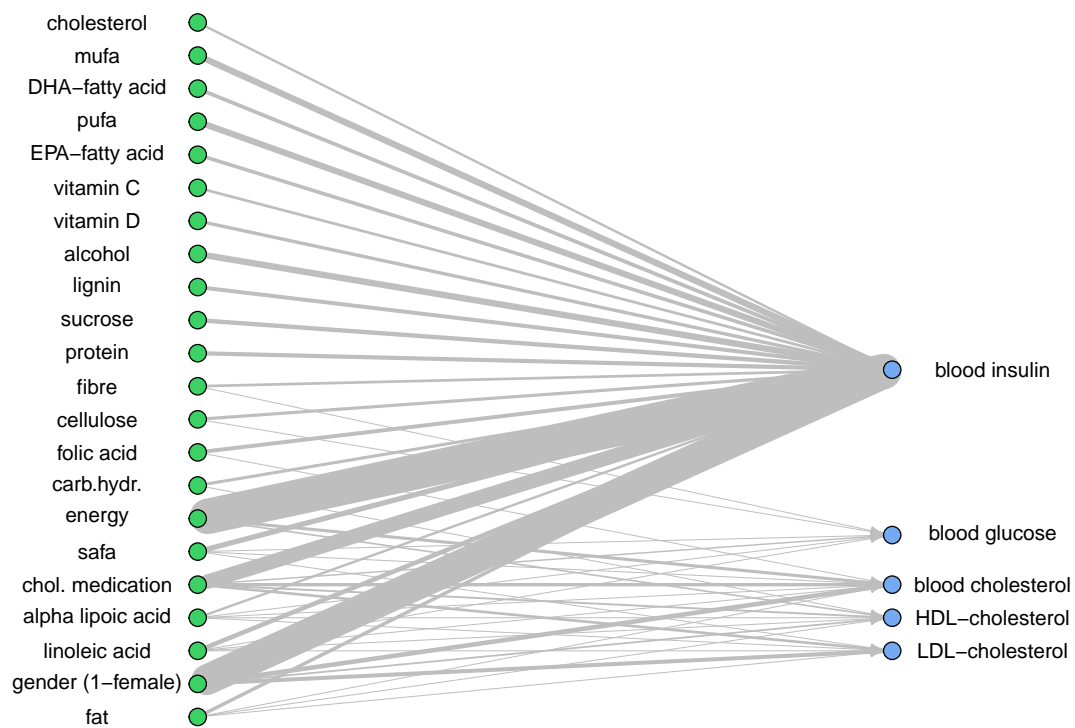
TODO: plot varcov-matrix Sigma_b (ggplot2 heatmap)

Clusters of personal graphs

Although there are personal differences, it is likely that not everyone behave uniquely but there might exist similar groups of behavior. We can analyze personal differences in two ways. We can look the absolute magnitudes of effects and we can also look how persons differ from typical behavior or mean of the effect.

We can now take the personal estimations of the effects and see, if they form clusters

Latent grouping
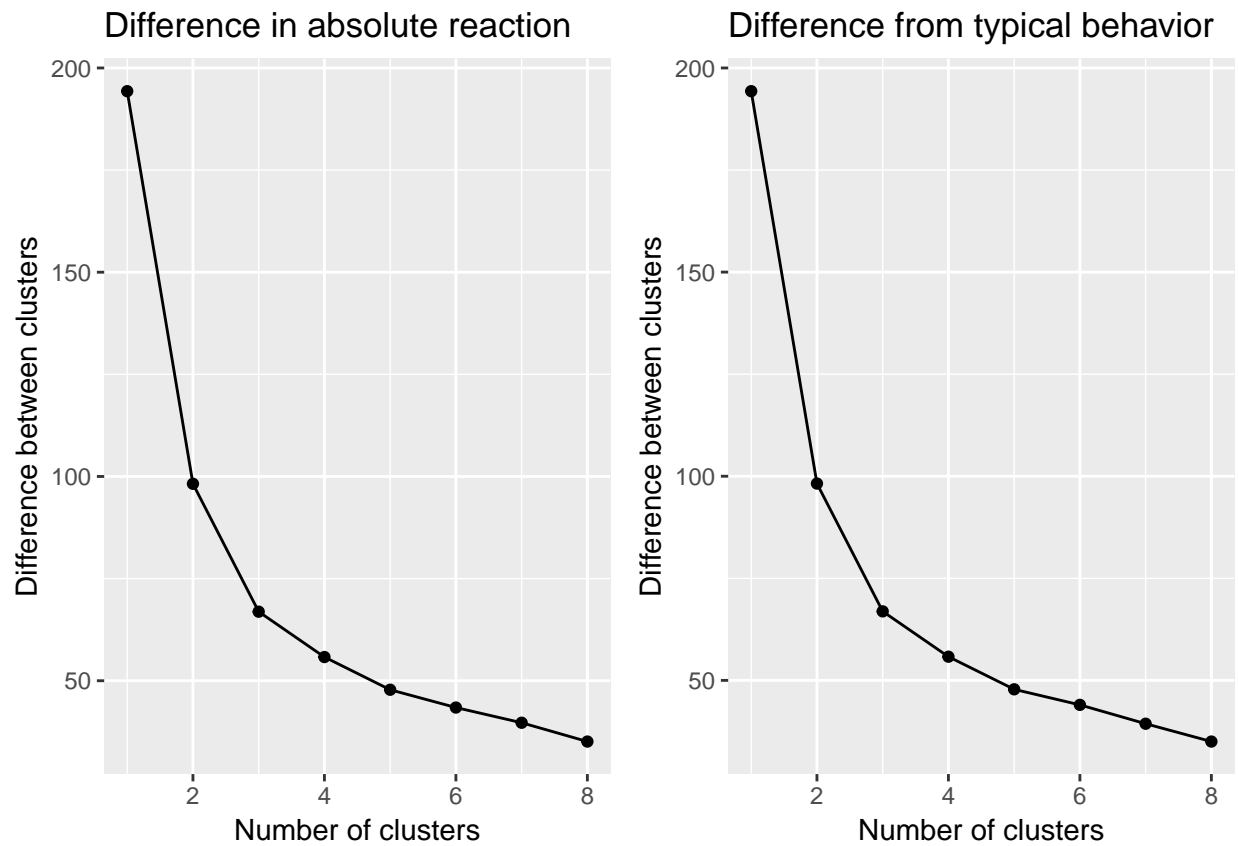
Difference from typical behavior

By looking previous diagram we see that there are four clearly identifiable groups between patients. At this plot, zero of X-axis denotes a typical behaviour of that particular reaction, and blue and red bars denote a deviation from this typical mean.
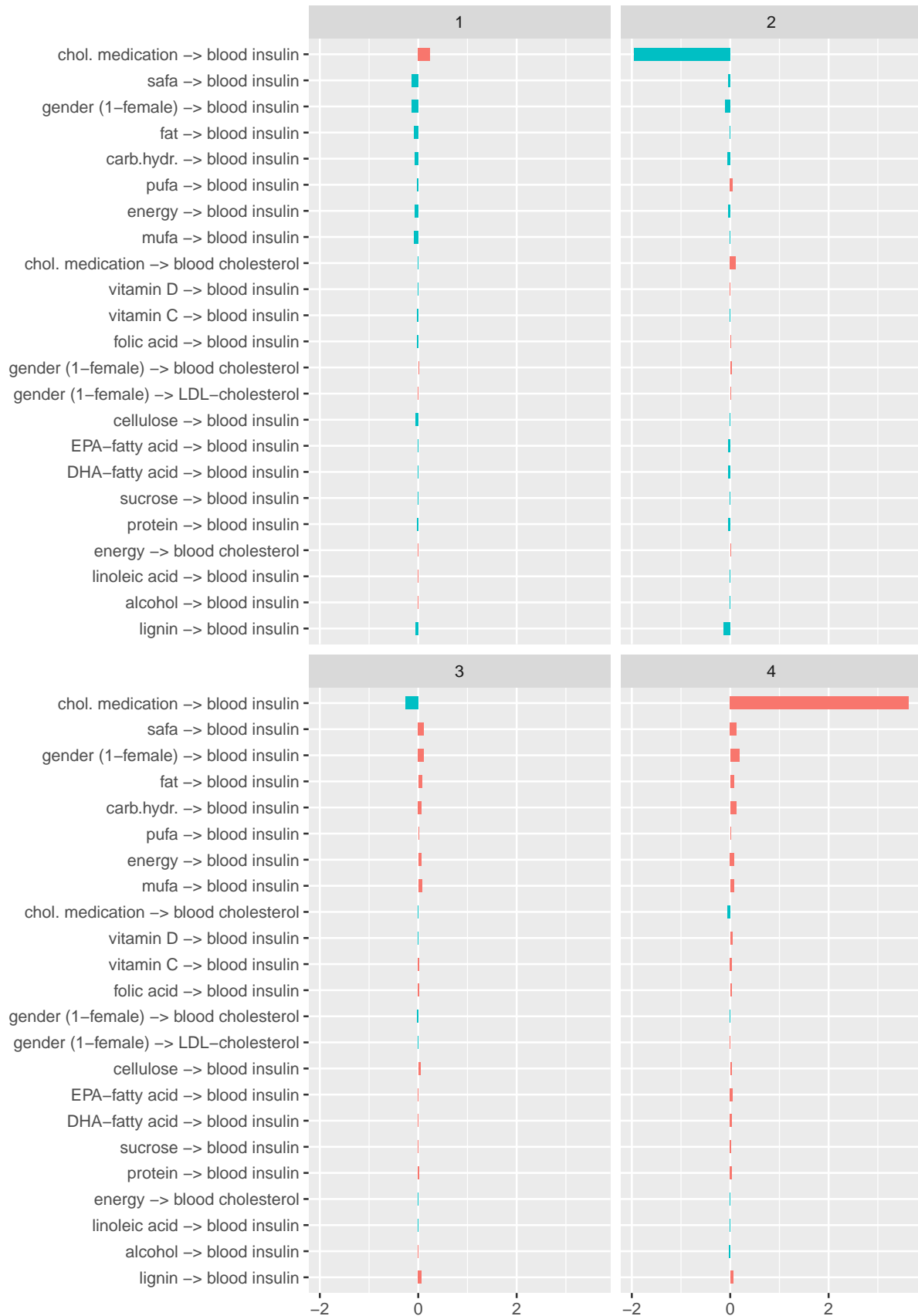
Cholesterol medication seems to dominate the clusters. There are patients for who cholesterol medication raises the blood insulin levels more than on average and for other group the raise is less than on average.

Let's see how the clustering shows with absolute effects rather than difference from mean. Note that these are not the greatest effects, but those with most personal variance.
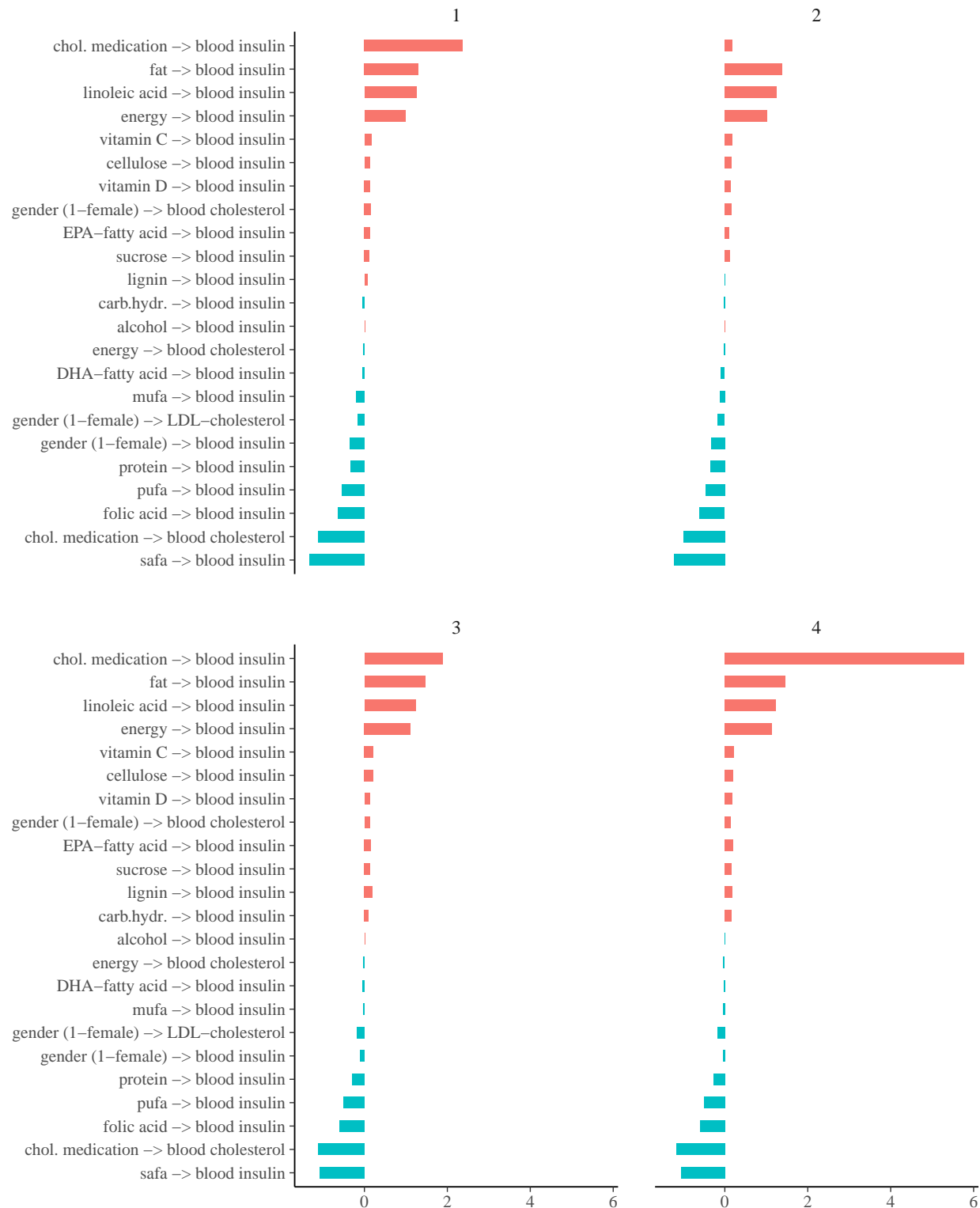
By looking at the absolute effect of the cholesterol medication it seems that there separetes two groups where one it is a significant factor in explaining blood insulin level, and other group where it does not play virtually any role. But does these patients even take cholesterol medications? Let's create a cross tabulation of observations to explore this effect more closely.

```
table(sysdimet$kolestrolilaakitys, sysdimet$effect_group)
```

```
##
##        1   2   3   4
##   0  156   0 152   0
##   1   20  36  32  28
```

Average blood insulin levels in these clusters:

```
for (i in seq(1,4)) print(paste0("cluster ", i, ": ", mean(sysdimet[sysdimet$effect_group == i,]$fsins)
```

```
## [1] "cluster 1: 8.41988636363636"
## [1] "cluster 2: 8.43055555555556"
## [1] "cluster 3: 14.7508152173913"
## [1] "cluster 4: 23.1107142857143"
```

There are 9 (=36/4) and 7 (=28/4) patients in clusters 2 and 4 who are taking cholesterol medication. For patients in the cluster 4 the average insulin level is the medication seems to rise the insulin level quite much, but for patients in the cluster 2 not much at all.

Let's try to confirm this finding by visualizing the components of insulin variance.
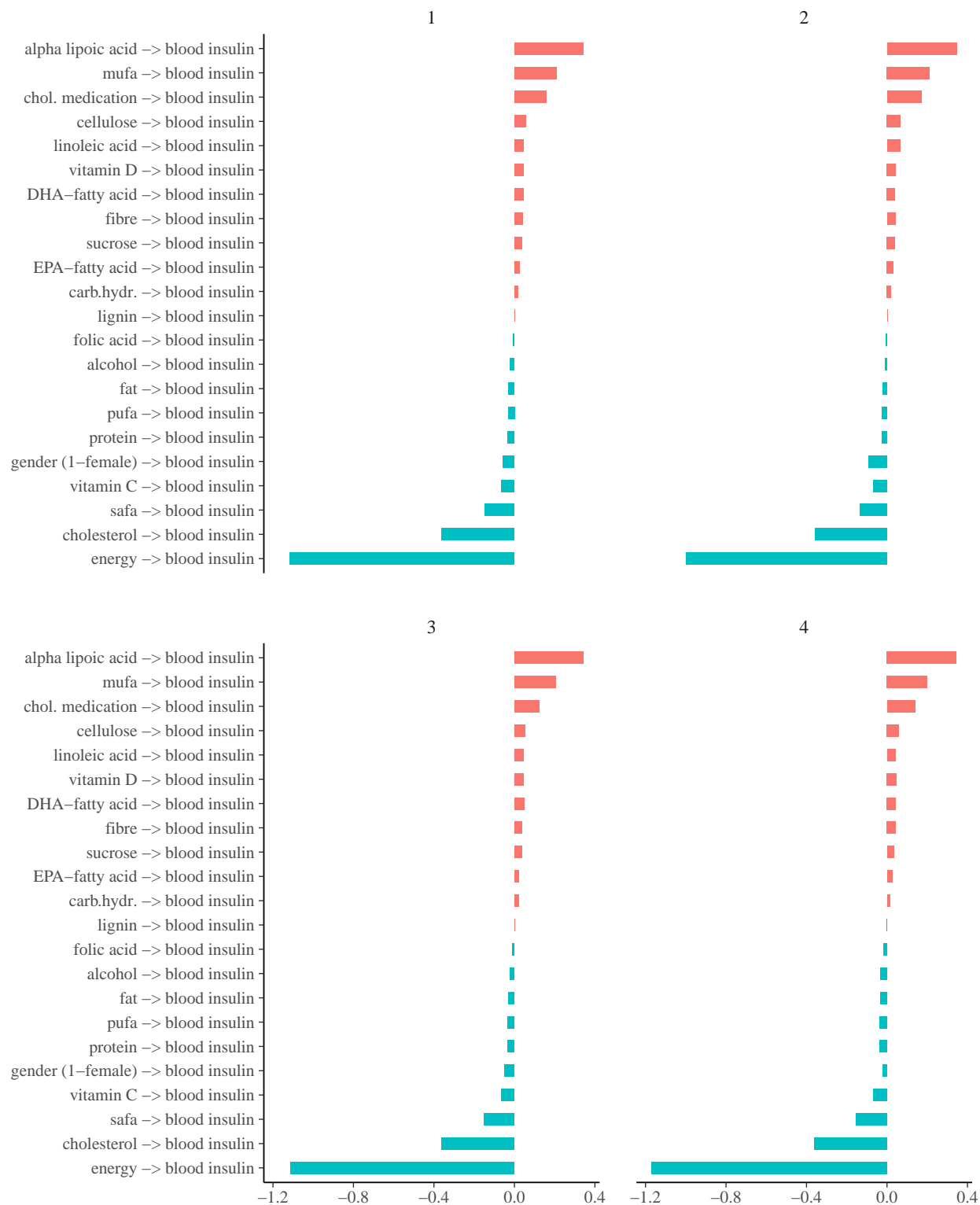
So, based on this we have an evidence that there is group on patients in this dataset who has significantly larger blood insulin values on average and a major component affecting the raise is cholesterol medication.

Both gender and cholesterol medication are unchanging factors at the dataset. Let us next remove those from clustering features and see how the clusters form based on nutritional effects only.

Number of clusters with nutritional effects only

In clusters 1 and 2 there are patients whose blood insulin levels react on lower and on higher than average. One interesting difference is lignin's effect to the blood insulin.

TODO: Can we take one of these groups and predict if new person belongs to these reaction groups?

Predicting personal reaction type

Finally, we can pick one the previously found reaction type clusters and hold aside the data for few of these patients. The goal would be then to use our model to predict the blood test values for this test group by their diet. This also reveals their personal reaction type for nutrition. We can compare this prediction to the true blood test values at dataset and also to the previous estimates from the whole sample population.

```
table(sysdimet$effect_group)
```

```
##
##   1   2   3   4
## 112  60 100 152
```

For testing, we sample a holdout set of patients overall the dataset. We then train a model without this holdout set and test how accurately we can place the predicted reaction types in previously found clusters. Note that we don't know the true reaction types for these patients. This makes absolute testing of prediction performance impossible, but in this way we can see at least if the prediction is consistent with analysis of the whole dataset. We know the previous cluster assiments of these holdout patients.

```
## [1]  9 45 50 52 63 95
```

The predictive ability of the model might be reduced by overfitting to small nuances of the whole dataset. To overcome the possible overfitting, we apply the Finnish horseshoe, a shrinkage prior, on regression coefficients. It allows specifying a prior knowledge, or at least an educated guess, about the number of significant predictors for a target. Here we guess that one third of the nutrients might be relevant for any given blood test, and apply following parameters for the shrinkage

```
shrinkage_parameters <- within(list(),
{
    scale_icept  <- 1          # prior std for the intercept
    scale_global <- 0.01825    # scale for the half-t prior for tau:
                               # ((p0=6) / (D=22-6)) * (sigma / sqrt(n=106*4))
    nu_global    <- 1          # degrees of freedom for the half-t priors for tau
    nu_local     <- 1          # degrees of freedom for the half-t priors for lambdas
    slab_scale   <- 1          # slab scale for the regularized horseshoe
    slab_df      <- 1          # slab degrees of freedom for the regularized horseshoe
})
```

Next we fit the previous model with the Finnish horseshoe added. Besides the shrinkage parameters, we provide now the data in two sets: input data for estimation and target data to hold out for prediction.

```
source("mebn/MEBN.r")

initial_graph <- mebn.fully_connected_bipartite_graph(datadesc)
sysdimet_gamma_ar1_rhs_pred <- mebn.bipartite_model(reaction_graph = initial_graph,
                                    inputdata = sysdimet,
                                    targetdata = holdout_index,
                                    predictor_columns = assumedpredictors,
                                    assumed_targets = assumedtargets,
                                    group_column = "SUBJECT_ID",
                                    local_estimation = mebn.sampling,
                                    local_model_cache = paste0("models/BLMM_gamma/ar1_rhs_effpred"),
                                    stan_model_file = "mebn/BLMM_gamma_ar1_rhs_effpred.stan",
                                    reg_params = shrinkage_parameters,
                                    normalize_values = TRUE)
```
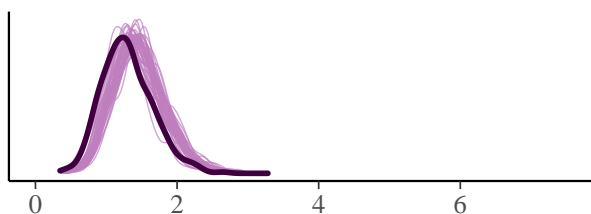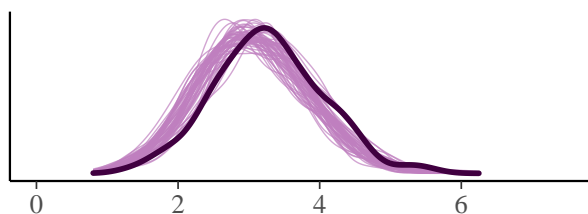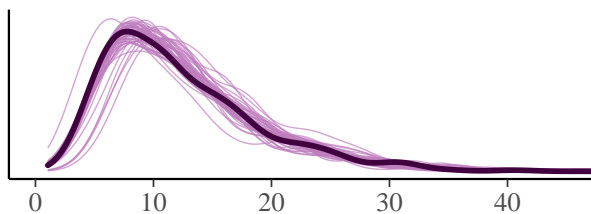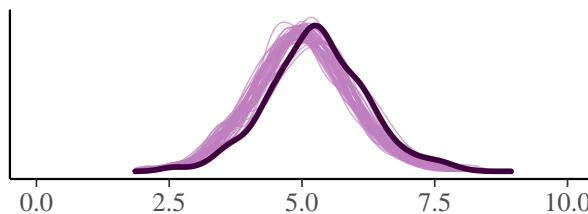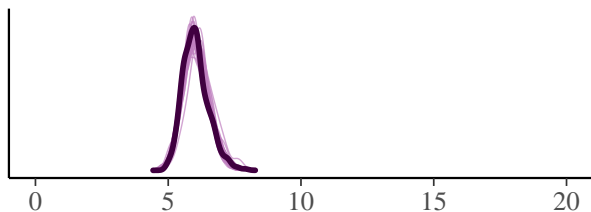
```
# get variation clusters of holdout persons

sysdimet$subject_idnum <- as.numeric(substr(sysdimet$SUBJECT_ID,2,5))
sysdimet[sysdimet$subject_idnum %in% holdout_persons,]$subject_idnum
```

```
## [1]  9  9  9  9 50 50 50 50 52 52 52 52 63 63 63 63
```

```
# TODO: näytä esim. ligniinin henkilökohtainen vaikutus insuliiniin ja onko se klusterin mukainen

# Näytä holdout-potilaan mittauksia ja miten reagointitapa ennuste sopii niihin (scatterplot?)

source("mebn/MEBN.r")

# This is a predicted personal reaction graph of patient in certain cluster

# This method adds (or should add) values of b and personal nodes to a typical graph

personal_graph9 <- mebn.personal_graph(person_id = 9,
                                       reaction_graph = sysdimet_gamma_ar1_rhs_pred,
                                       predictor_columns = assumedpredictors,
                                       assumed_targets = assumedtargets,
                                       local_model_cache = "BLMM_gamma/ar1_rhs_effpred/")
```
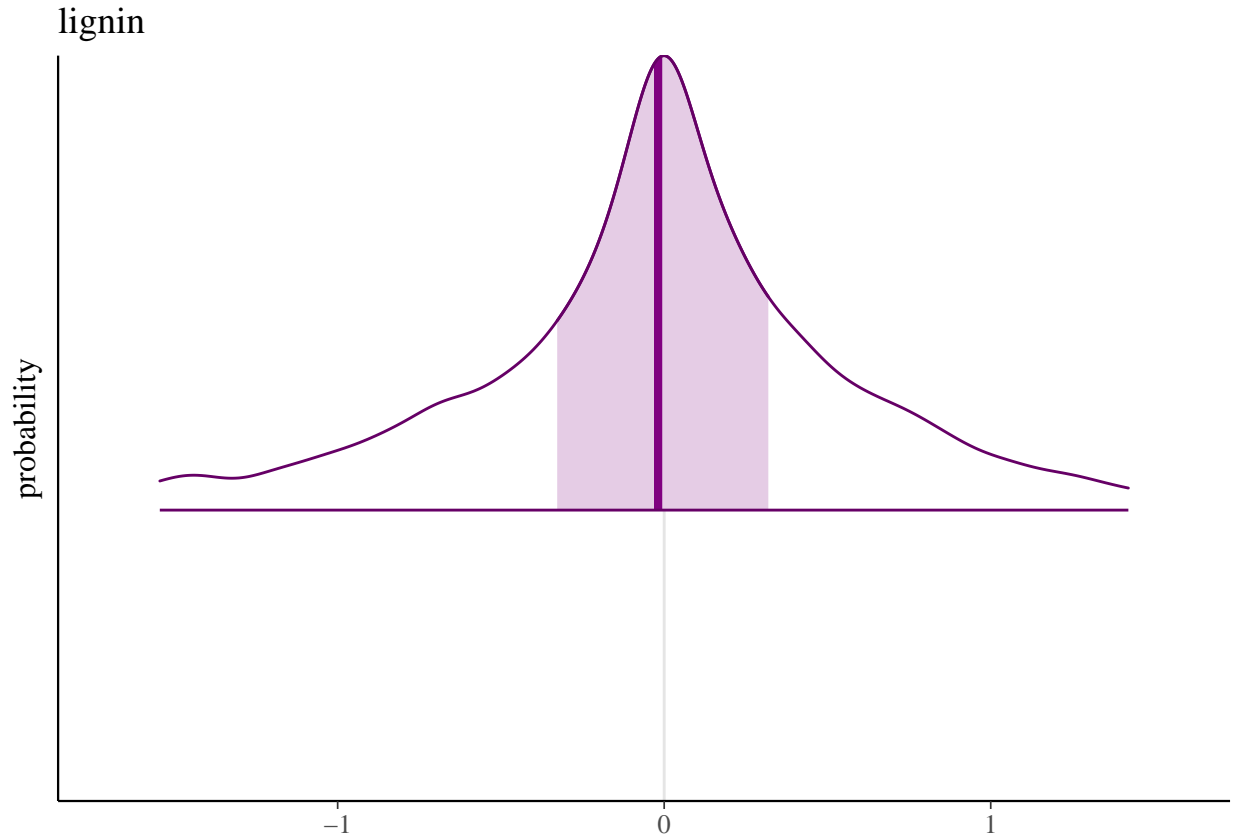
```
source("mebn/MEBN.r")
#mebn.plot_personal_effects(p9, 15)
mebn.plot_personal_effects(personal_graph9, 15)
```

Compare lignin to insulin effect of two patients who were previously found to be in different clusters

Bae, Harold, Stefano Monti, Monty Montano, Martin H. Steinberg, Thomas T. Perls, and Paola Sebastiani. 2016. "Learning Bayesian Networks from Correlated Data" 6 (May). The Author(s) SN -: 25156 EP. http://dx.doi.org/10.1038/srep25156.

Koller, Daphne, and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning.* The MIT Press.

Lankinen, M., U. Schwab, M. Kolehmainen, J. Paananen, K. Poutanen, H. Mykkanen, T. Seppanen-Laakso, H. Gylling, M. Uusitupa, and M. Ore?i? 2011. "Whole grain products, fish and bilberries alter glucose and lipid metabolism in a randomized, controlled trial: the Sysdimet study." *PLoS ONE* 6 (8): e22646.