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

Supplementary Materials

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

Abstract

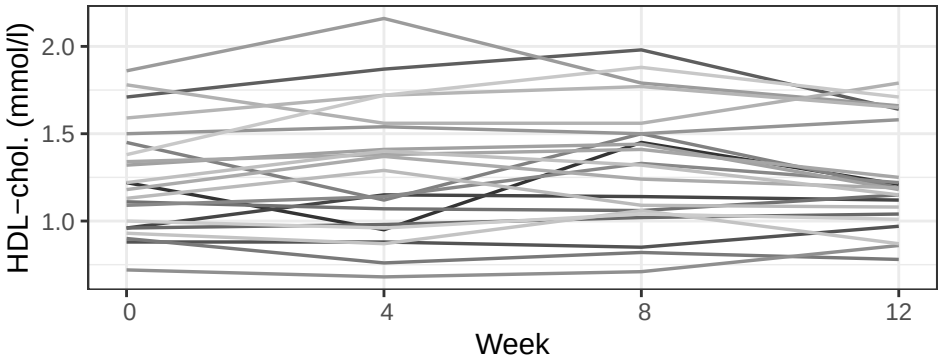
This notebook is a supplementary material for the article "Revealing the Personal Effects of Nutrition with Mixed-Effect Bayesian Network". The notebook shows in detail how mixed-effect Bayesian network is constructed and how it is used to model the effects of nutrition. Comparison of The hierarchical structure of the graphical model allows us to study the effects in both typical and personal levels. In addition we show that personal effect variations form clusters of similarly behaving patients. Finally, we show how personal graphical models can be predicted for patients that are held out from the model estimation. This shows that personal differences in nutrional effects do exist, and that they can be detected from repeated observations of food diaries and blood tests.

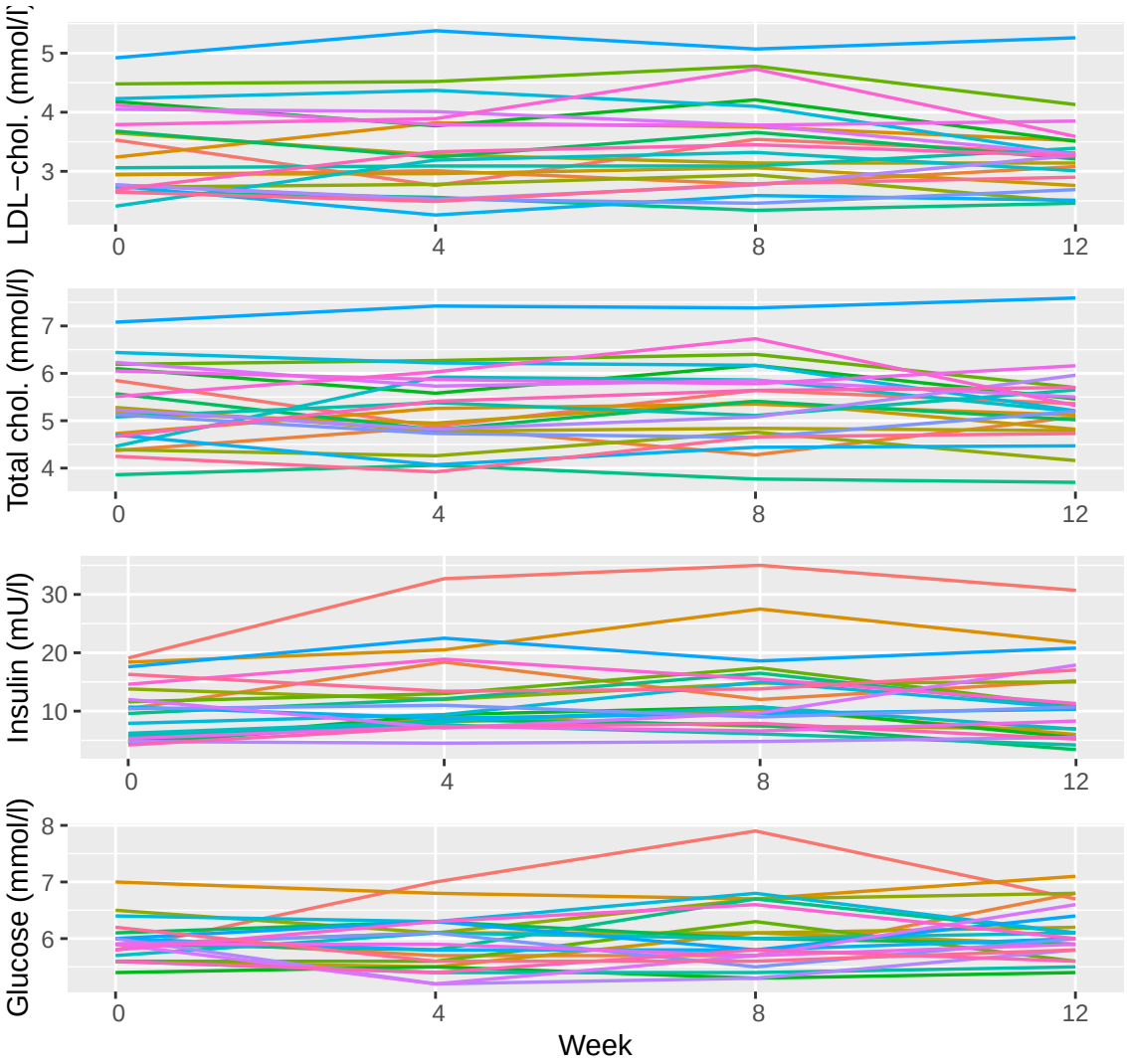
In this work we propose a Bayesian network as an appealing way to model and predict the effects of nutrition. The Bayesian network are directed graphical models where nodes of the graph are random variables and the edges between the nodes indicate the effects between variables. In the nutritional modelling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

Motivating example of personal nutrition data

As an example, we analyze a dataset from the Sysdimet study (Lankinen et al. 2011) that contains four repeated measurements of 17 nutrients, some basic information about patients (gender, medication) and their corresponding blood test results, from 106 patients.

Following spagettiplots show how the blood test measurements progress during the 12-week study. Measurements of only ten of 106 patients are shown at the plot here for clarity. Our goal is to predict next personal blood test result when patient's diet from past weeks is known. It is also likely that the successive blood test measurements are correlated to each other.





When the future direction of the blood test levels can be predicted, the typical and personal importance of nutrients contributing this change be estimated from the model parameters.

Mixed-effect Bayesian network

We focus only on two-level, bipartite, graphs where nutrients and personal details are assumed to affect blood tests, and only the magnitude of the 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. Thorough explanation of the model theory can be found at the main article.

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

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

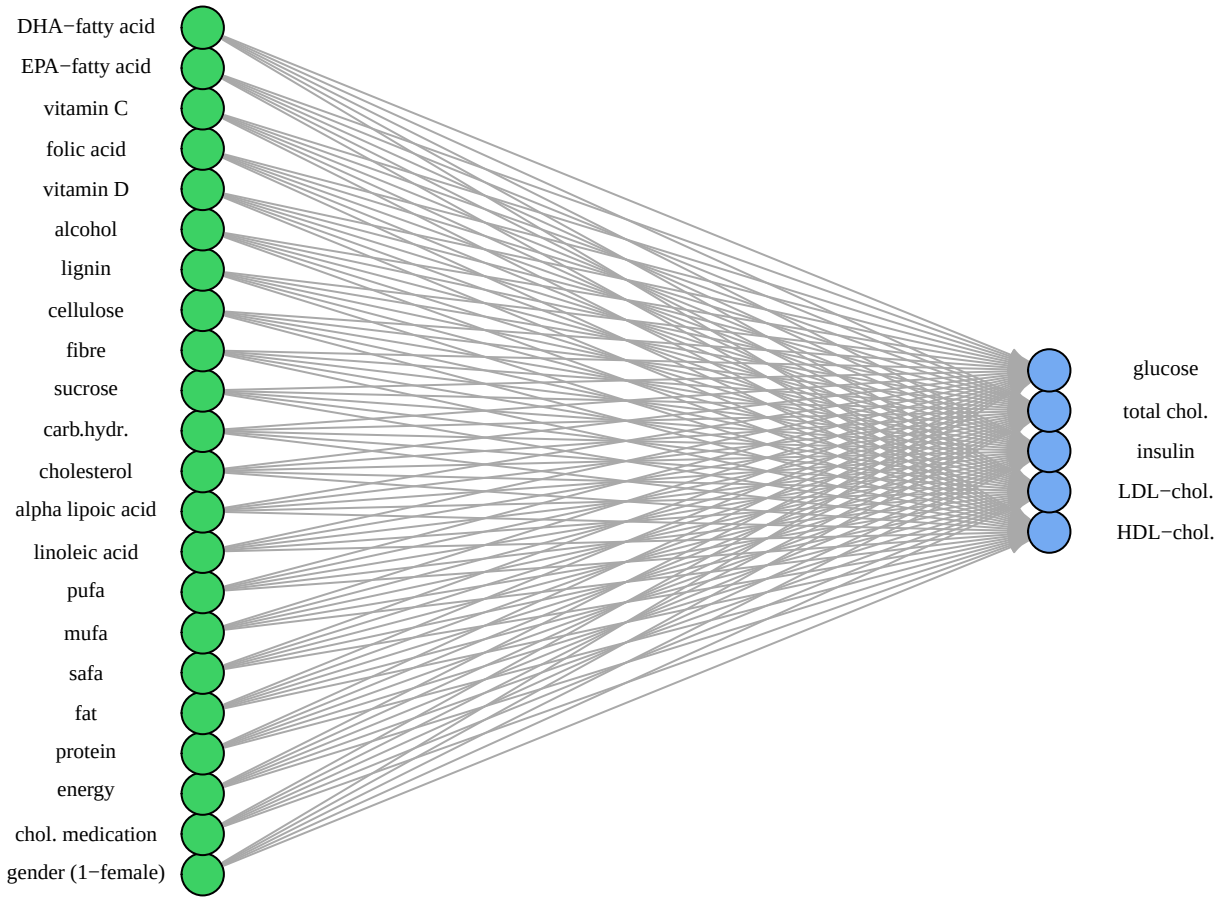
This turns the problem to finding a graph that is most probable given data. To enforce our assumption of biologically plausible graphs, we set the graph prior $P(G)$ to zero for all other graphs than the previously descibed bipartite graphs with nutrients affecting the bodily responses.

The joint probability of the graph $P(G|D)$ can furthermore factorized into separate local probability distributions (Bae et al. 2016, @Koller:2009:PGM:1795555) by their *Markov blankets*. Hierarchical mixed-effect estimation allows us to study the distributions in both typical and personal levels.

Hierarchical estimation of the local distributions

The joint probability distribution of the graph is factorized into separate distributions. We assume that the blood test random variables Y_i are affected by a linear combination of the nutrient variables $pa(Y_i)$. The set of parameters describing this effect, ϕ_i, can be split into typical β and personal b parts. We assume that the actual data denoted by the random variables follows distributions from the exponential family and we can use some link function to use this linear predictor.

$$\begin{aligned} P(D|G)P(G) &= \prod_{i=1}^{v} P(D|G_i)P(G_i) \\ &= \prod_{i=1}^{v} P(Y_i|pa(Y_i), \phi_i)P(G_i) \\ &= \prod_{i=1}^{v} EXPFAM(Y_i|pa(Y_i)\beta_i + pa_Z(Y_i)b_i)P(G_i) \end{aligned} \tag{3}$$



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 describing the connections.

Model development

Our starting assumption is that both nutrients and the blood test values are normally distributed. This assumption may be 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.

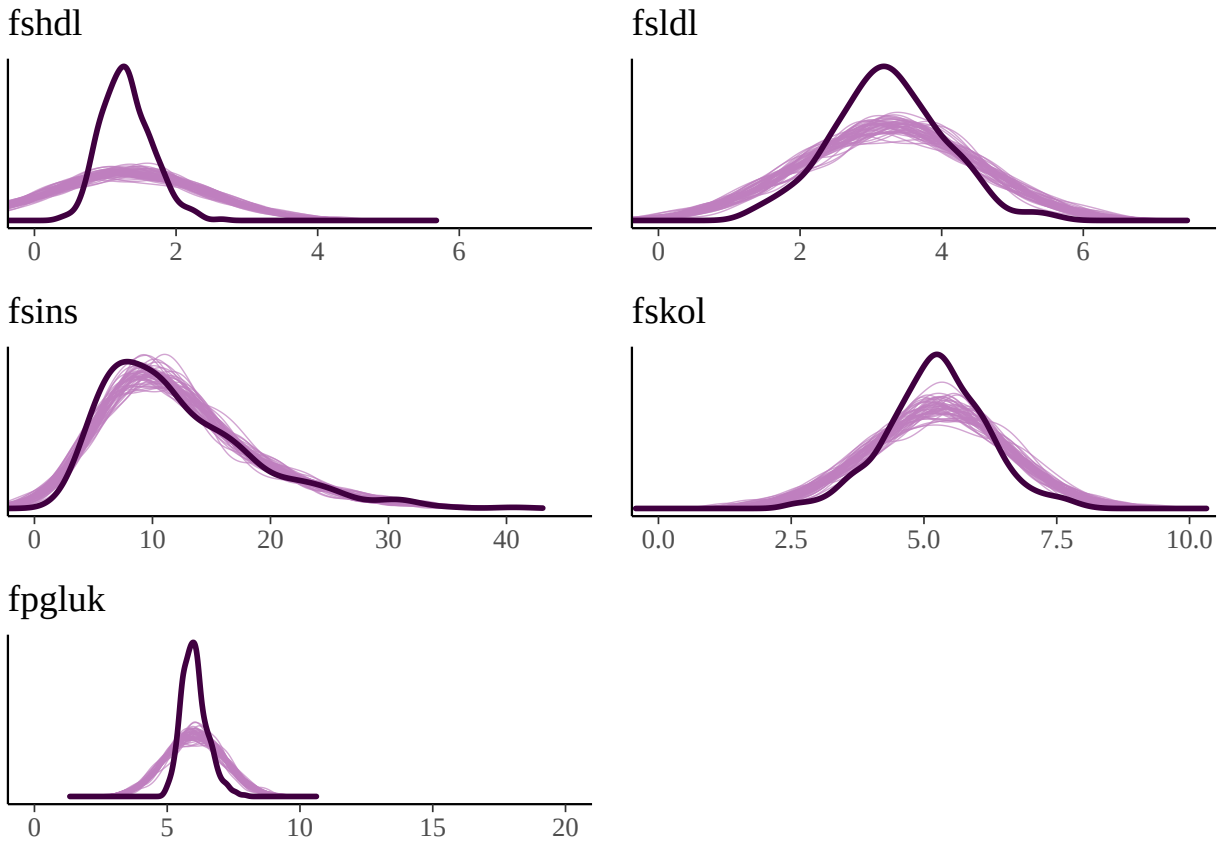
```
sysdimet_normal <- mebn.bipartite_model(reaction_graph = initial_graph,
                                        inputdata = sysdimet,
                                        predictor_columns = assumedpredictors,
                                        assumed_targets = assumedtargets,
         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

```
## [1] "fshdl"
```

```
## [1] "fsldl"
## [1] "fsins"
## [1] "fskol"
## [1] "fpgluk"
```

```
write.graph(sysdimet_normal, "graphs/sysdimet_normal.graphml", "graphml")
```

We evaluate the fit of model candidates visually by plotting the posterior predictive distributions (PPC) over the distributions of the true blood test values. As expected, the normally distributed random variables are not fitting well. Normal distribution allows probability for negative blood test values that resulting the probability to leak for impossible values. Normal distribution is also symmetric while the distributions of blood test values are more skewed to right.



Developing the model beyond normal distributions

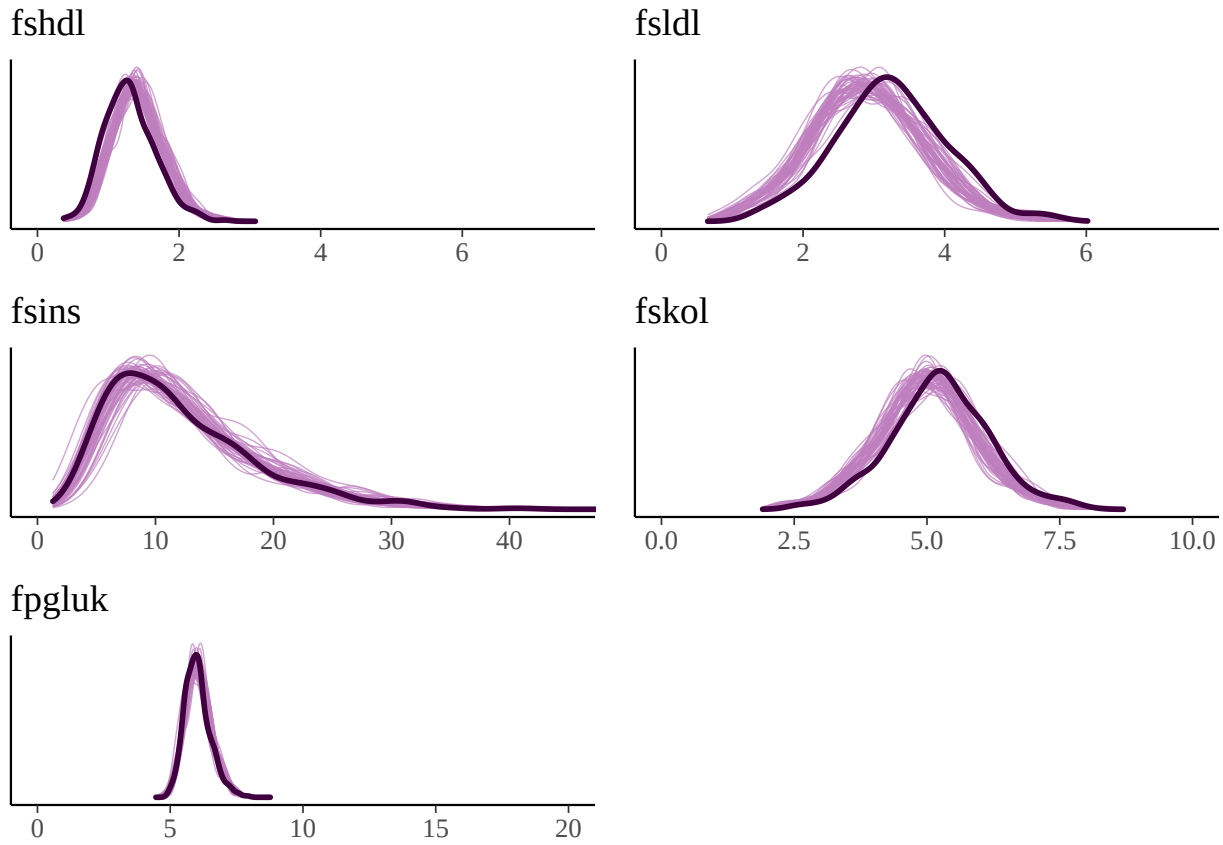
More realistic probability distribution for blood test values would be Log-Normal or Gamma distributions (Limpert 2011). They both allow only positive values and model better the right tail of individual larger values. For further development, we choose Gamma distribution with identity link function. This is important as it keeps the regression coefficients of the nutrients on the same absolute scale than with Normal models. The regression coefficients are used as weights of the edges at the Bayesian network and they all should be on the same scale regardless of the random variables' distribution.

```
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",
                                       stan_model_file = "mebn/BLMM_gamma.stan",
                                       normalize_values = TRUE)
```

```
## [1] "fshdl"
## [1] "fsldl"
## [1] "fsins"
## [1] "fskol"
## [1] "fpgluk"
```

```
write.graph(sysdimet_gamma, "graphs/sysdimet_gamma.graphml", "graphml")
```

The visual comparison of the true and estimated distributions of blood test variables shows that Gamma distribution follows the true distributions quite well.



The visual PPC inspection shows that the Gamma distribution fit generally better to responses than Normal

distribution, but cholesterol values are still systematically estimated too low (fsldl and fskol) or too high (fshdl).

Modeling correlated observations with an autocorrelation structure

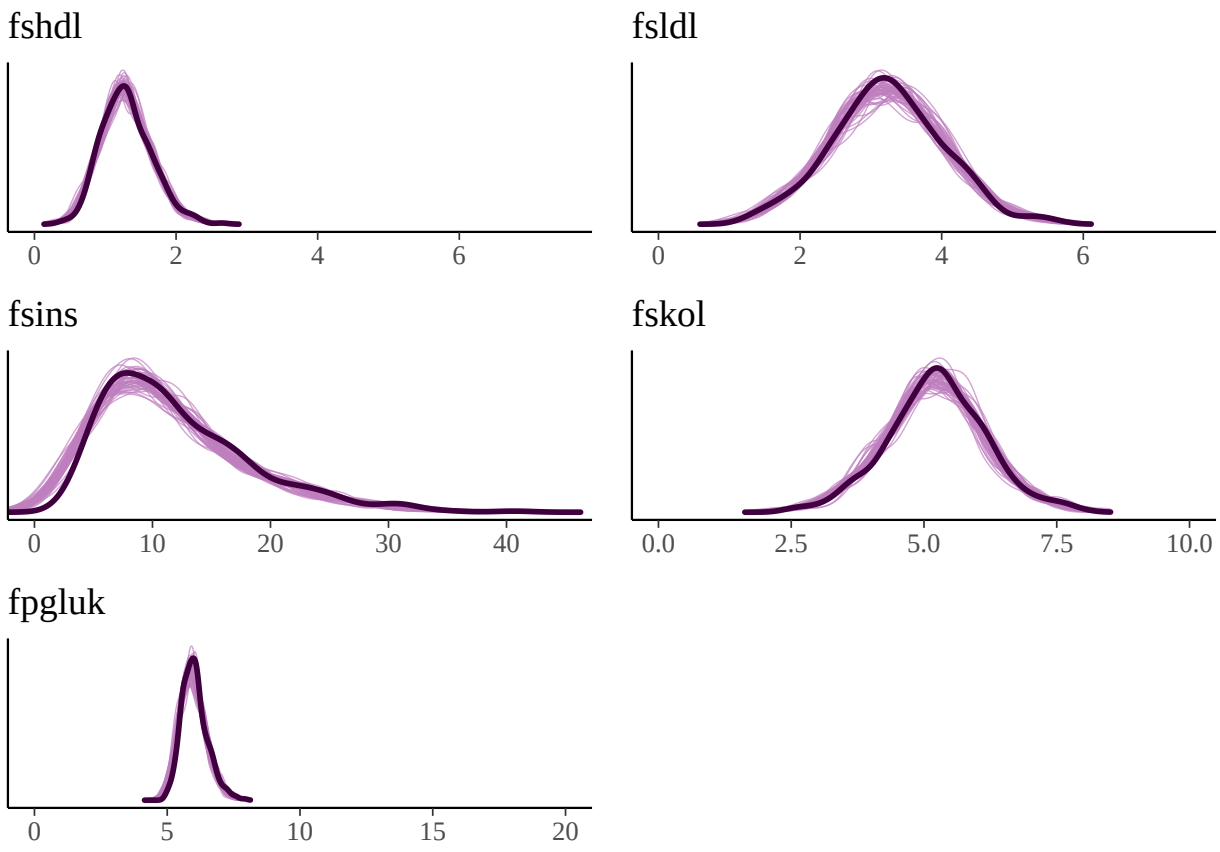
In the dataset the observations from patients' food diaries and blood tests have one week response time. This might not be optimal time for all the responses and the successive observations might be correlated. For modeling the correlated observations, we add an autocorrelation process to the model by adding an estimated portion of previous measurement to the linear predictor

```
mu[n] += Y[n-1] * ar1
```

This coefficient 'ar1' is estimated separately for each response

```
## [1] "fshdl"
## [1] "fsldl"
## [1] "fsins"
## [1] "fskol"
## [1] "fpgluk"
```

The autocorrelation structure seems to decrease the variance of the model and correcting the systematical estimation errors from the previous model candidate



It is interesting also to compare the AR(1) coefficients of different local distributions at the graph. Blood insulin (fsins) has largest AR(1) coefficient indicating that successive measurements are most correlated and the values are not changing as rapidly as with other measurements. Credible interval for cholesterol measurements includes zero indicating that autocorrelation is very low or non-existent between two measurements.

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

distribution	AR(1)	CI-10%	CI-90%
fshdl	0.00066	-0.00052	0.00179
fsldl	0.00064	-0.00180	0.00315
fsins	0.04553	0.02990	0.06127
fskol	0.00177	-0.00118	0.00472
fpgluk	0.00380	0.00157	0.00600

Skrinkage prior model

In our analysis, we use two models: 1) typical effects are better shown with previous model that has vague prior on beta coefficients, and 2) personal effects became clearer in a model that uses shrinkage prior on beta coefficients. This shrinkaged model is also faster to evaluate in k-fold setting and is possibly a better predictor.

The predictive ability of the model might be reduced by overfitting to small nuances of the whole dataset. To overcome the possible overfitting of non-shrinked model, 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 <- 0.02428   # scale for the half-t prior for tau:
                              # ((p0=11) / (D=22-11)) * (sigma = 1 / 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.

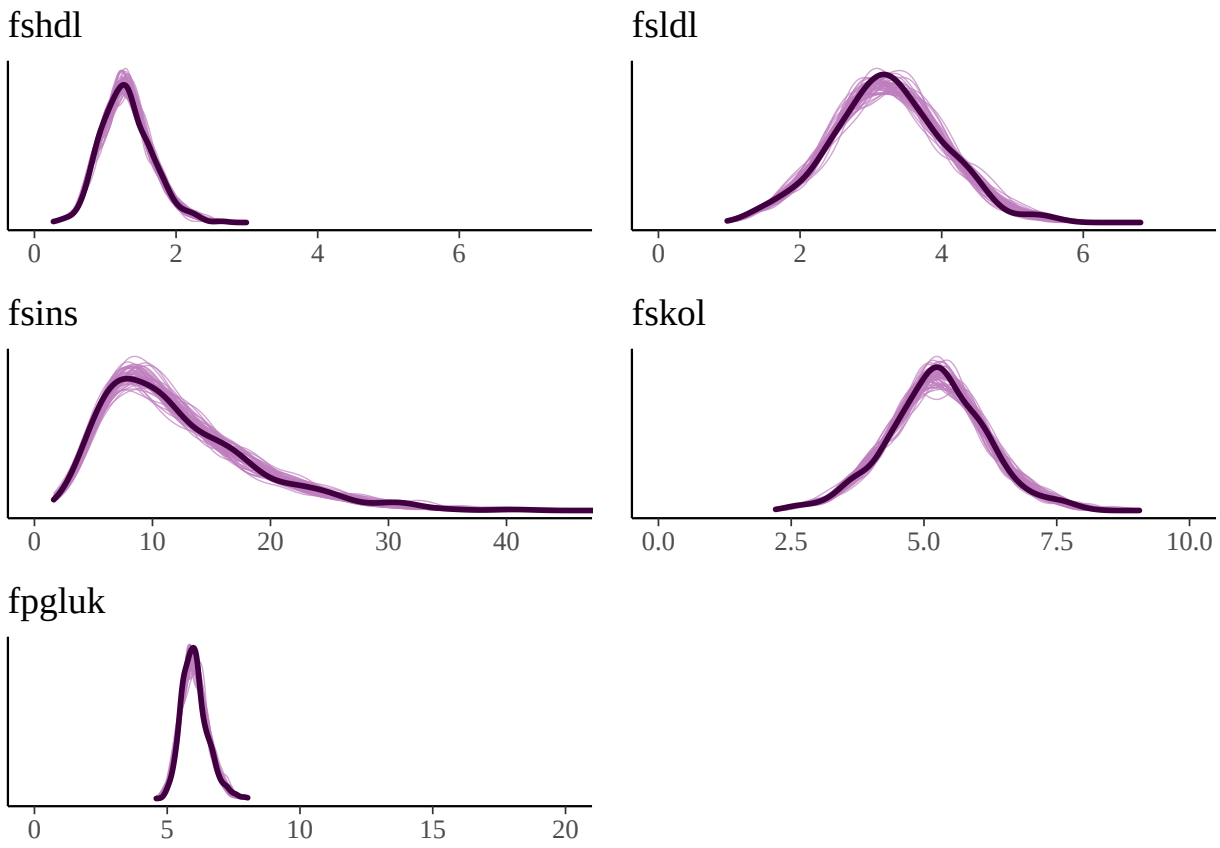
```
# One RHS model with all data (no_holdout)
holdout_index <- rep(0, nrow(sysdimet))
```

```
initial_graph <- mebn.new_graph_with_randomvariables(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/no_holdout"),
                                    stan_model_file = "mebn/BLMM_gamma_ar1_rhs_cv_ppc.stan",
                                    reg_params = shrinkage_parameters,
                                    normalize_values = TRUE)
```

```
## [1] "fshdl"
## [1] "fsldl"
## [1] "fsins"
## [1] "fskol"
## [1] "fpgluk"
```

```
write.graph(sysdimet_gamma_ar1_rhs_pred, "graphs/sysdimet_gamma_ar1_rhs.graphml", "graphml")
```



Visual comparison with posterior predictive checks seem to favor this model. Let us confirm the evaluation with numerical metrics.

Evaluation of the models' fit and predictive performance

Diffent model versions are compared with a normalized root mean squared error (NRMSE) and also with a classification test that shows better the usefulness of the method. The classification test allows also comparison with other classification methods.

Table 1: Predictive accuracy percents for correct direction of change

model	fshdl	fsldl	fsins	fskol	fpgluk	mean
Normal	0.039	0.06	0.222	0.062	0.055	0.088
Gamma	0.024	0.062	0.279	0.048	0.028	0.088
Gamma AR(1)	0.011	0.026	0.271	0.036	0.029	0.075
Gamma AR(1) RHS	0.013	0.03	0.291	0.041	0.032	0.081

Comparison with in-sample data

Following compares different model candidates posterior predictive distributions with their true values using normalized root mean squared error (NRMSE) metrics. It allows comparing the prediction accuracy of responses with different scales.

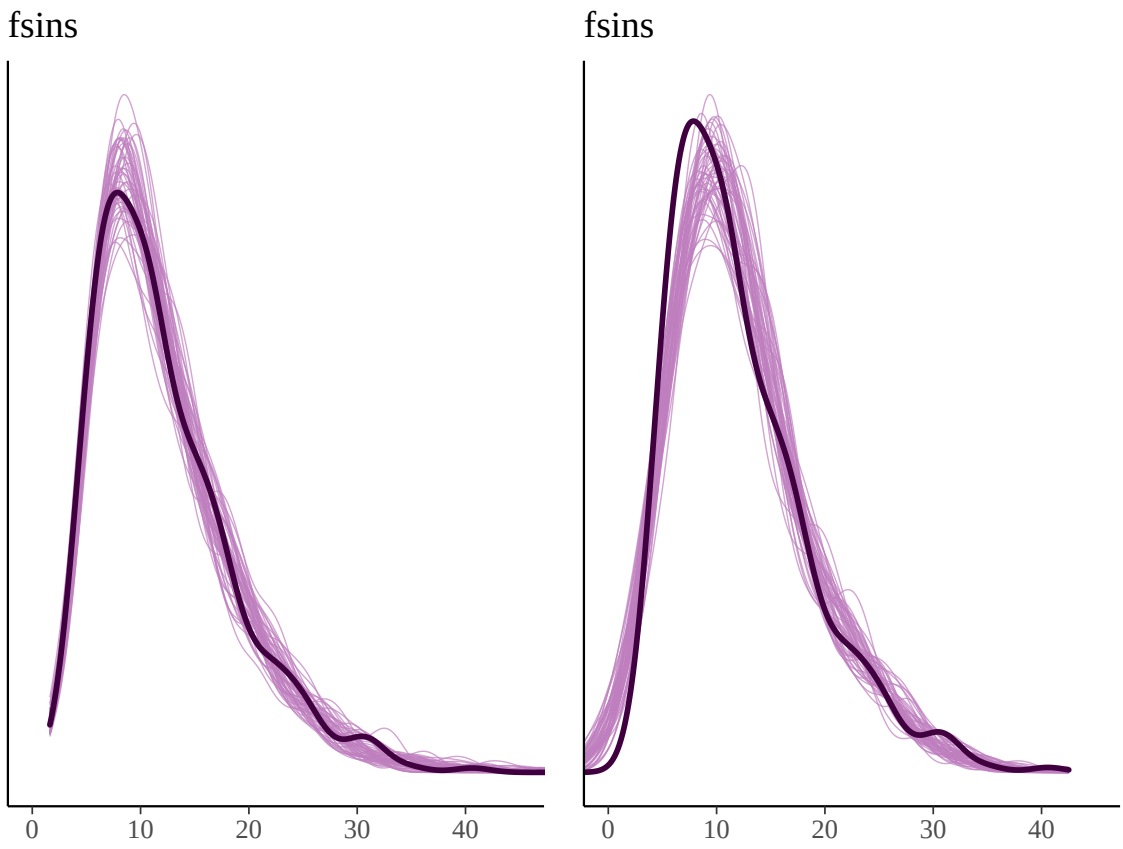
It shows that Gamma distribution with AR(1) autocorralation structure has best overall fit to data. Horseshoe prior on typical effects has shrinked some coefficients close to zero and it has increased mean NRMSE just a little. An interesting finding is that the blood insulin (fsins) predictions are better modeled with Normal distribution. For the final MEBN we replace Gamma AR(1) with Normal AR(1) model for fsins.

```
## [1] "fsins"
```

```
initial_graph <- mebn.new_graph_with_randomvariables(datadesc)
only_fsins <- assumedtargets[assumedtargets$Name=="fsins",]

fsins_normal_ar1_rhs <- mebn.bipartite_model(reaction_graph = initial_graph,
                                    inputdata = sysdimet,
                                    targetdata = holdout_index,
                                    predictor_columns = assumedpredictors,
                                    assumed_targets = only_fsins,
                                    group_column = "SUBJECT_ID",
                                    local_estimation = mebn.sampling,
                                    local_model_cache = paste0("models/BLMM_normal/ar1_rhs/no_holdout"),
                                    stan_model_file = "mebn/BLMM_normal_ar1_rhs_cv_ppc.stan",
                                    reg_params = shrinkage_parameters,
                                    normalize_values = TRUE)
```

```
## [1] "fsins"
```

Plot on the left is a PPC of Gamma distributed fsins-response and the right plot is a normally distributed response that fits better with lower mean squared error



Next, we contruct a EXPFAM AR(1) model that has this normally distributed fsins-response and otherwise Gamma distributed blood test responses.

```
## [1] "fshdl"
## [1] "models/BLMM_gamma/ar1"
## [1] "fsldl"
## [1] "models/BLMM_gamma/ar1"
## [1] "fsins"
## [1] "models/BLMM_normal/ar1"
## [1] "fskol"
## [1] "models/BLMM_gamma/ar1"
## [1] "fpgluk"
## [1] "models/BLMM_gamma/ar1"
```

```
## [1] "fshdl"
## [1] "models/BLMM_gamma/ar1_rhs/no_holdout"
## [1] "fsldl"
## [1] "models/BLMM_gamma/ar1_rhs/no_holdout"
## [1] "fsins"
## [1] "models/BLMM_normal/ar1_rhs/no_holdout"
## [1] "fskol"
## [1] "models/BLMM_gamma/ar1_rhs/no_holdout"
## [1] "fpgluk"
## [1] "models/BLMM_gamma/ar1_rhs/no_holdout"
```

Table 2:

model	fshdl	fsldl	fsins	fskol	fpgluk	mean
Normal	0.039	0.06	0.222	0.062	0.055	0.088
Gamma	0.024	0.062	0.279	0.048	0.028	0.088
Gamma AR(1)	0.011	0.026	0.271	0.036	0.029	0.075
Gamma AR(1) RHS	0.013	0.03	0.291	0.041	0.032	0.081
EXPFAM AR(1)	0.011	0.026	0.227	0.036	0.029	0.066
EXPFAM AR(1) RHS	0.013	0.03	0.234	0.041	0.032	0.07
Gamma AR(1) RHS CV	0.021	0.048	0.443	0.064	0.047	0.125
EXPFAM AR(1) RHS CV	0.021	0.048	0.452	0.064	0.047	0.126

Predictive accuracy cross-validated MEBN

Execution of the k-fold cross-validation is done in a separated notebook (cross-validation.Rmd) as it takes substantial amount of time to run. The results are stored and load for analysis here. For the cross-validation we partitioned the observations in data to 12 partitions for weeks 4,8 and 12. Then each of the data partitions in turn is left out from model estimation. This allows us to make predictions for unseen data.

Each of the model candidates are compared here with a normalized root mean squared error (NRMSE) and last final versions (Gamma AR(1) RHS CV and Expfam AR(1) RHS CV) are evaluated with the cross-validation.

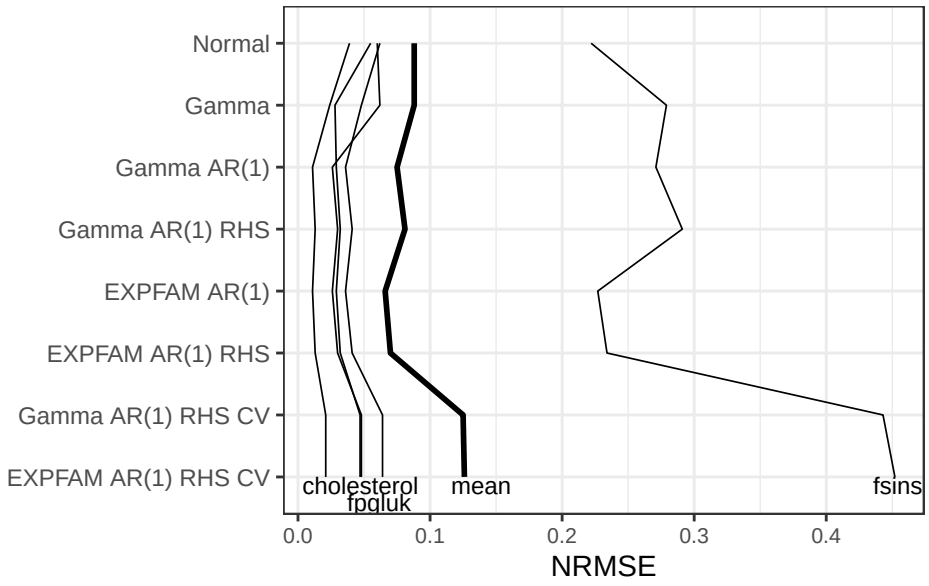


Table 3: Predictive accuracy percents for correct direction of change

	0 to 4 weeks	**4 to 8 weeks**	**8 to 12 weeks**	**mean accuracy**
fshdl	67%	61%	67%	65%
fsldl	76%	69%	63%	69%
fsins	76%	67%	68%	70%
fskol	67%	69%	63%	66%
fpgluk	62%	56%	65%	61%

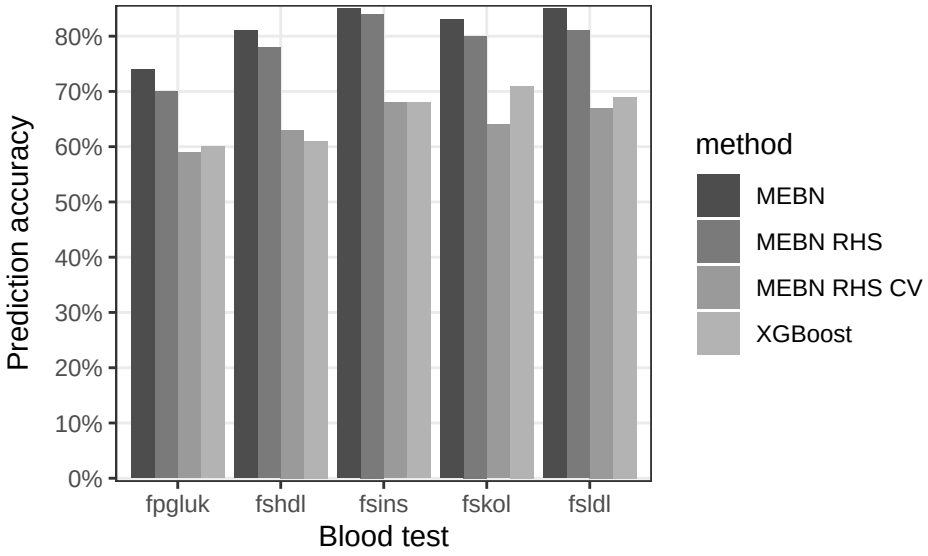
Evaluation with classification metrics

Normalized root mean squared error (NRMSE) is good for comparing model candidates, but it does not say much about the actual usefulness of the model. For personal nutrition guiding, it is important to predict how given diet will affect the future blood test values. In the standard NRMSE comparison, the error between true and predicted values were compared, but a custom classification test can be also used. It tells if the given diet raises or lowers the future blood test values compared to past measurement.

Classification test allows also to compare MEBN with other classification methods. XGBoost is a well-performing decision tree method. It is a non-parametric machine learning method that learns personal reactions only by example. Following compares the cross-validated predictive accuracy of XGBoost with MEBN predictions (MEBN CV) that uses EXPFAM AR(1) RHS-model. Also in-sample MEBN fit with non-shrinked EXPFAM AR(1) is given for comparison. This best predicting model is used for dataset analysis at rest of this notebook.

Summary of the model comparison

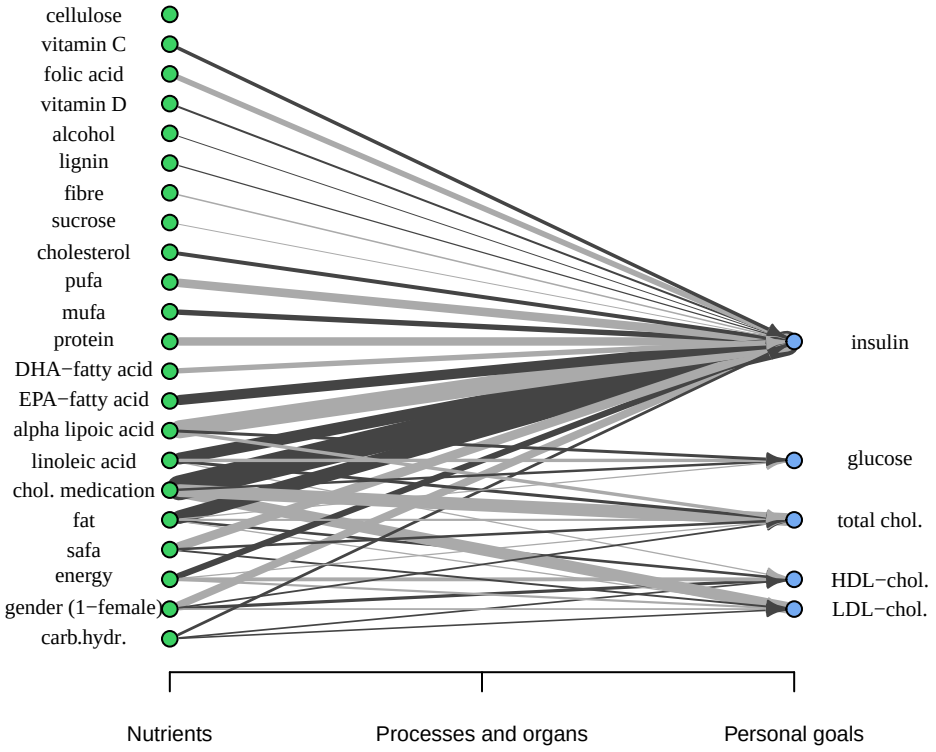
This table summarizes how accurately we can predict if blood test values raise or lower when the current level and a diet are known. This evaluates the model that uses shrinkage.



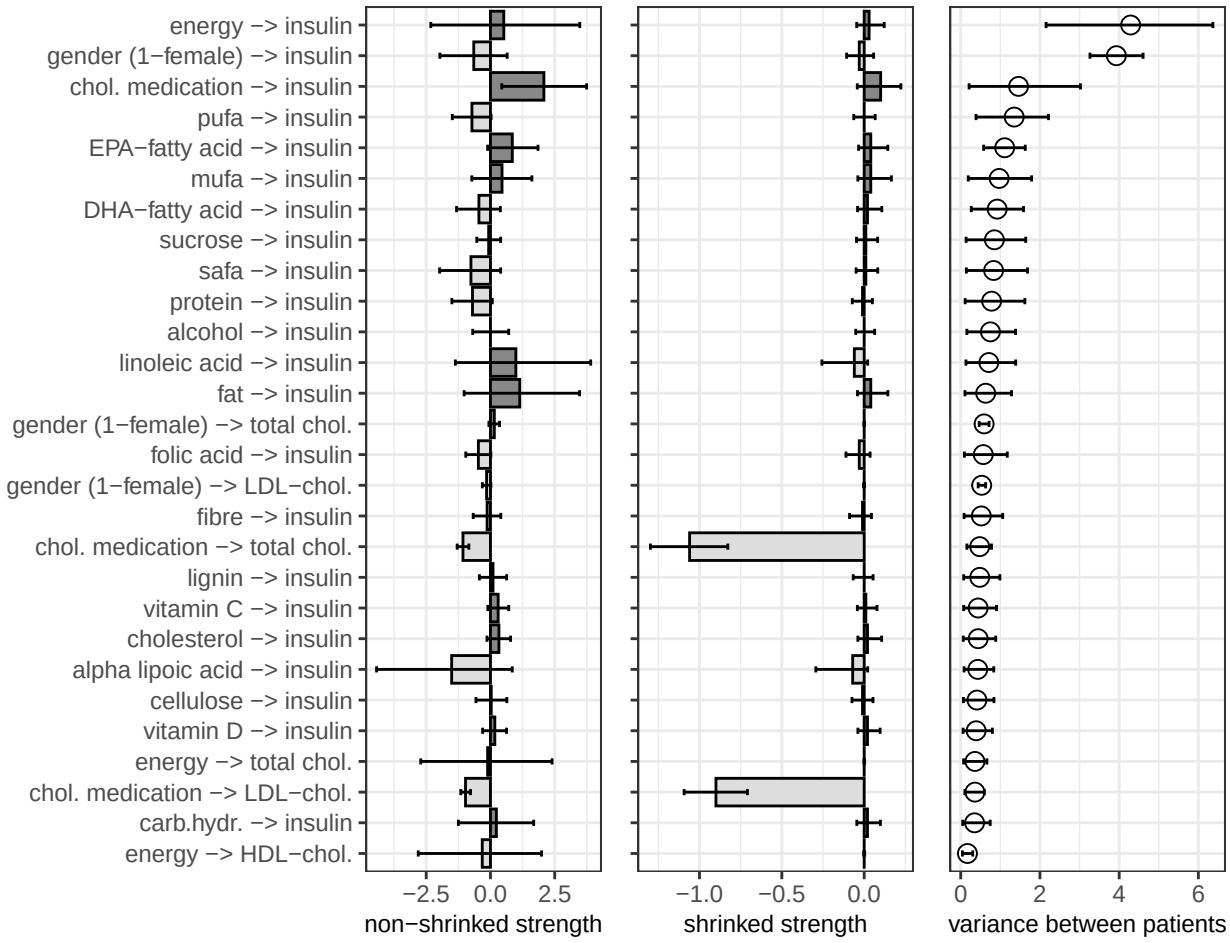
Typical effects of nutrition, and variance from the typical effects

We have now reached a reasonably good fit for the local distributions of the Bayesian network. Our code has extracted a graphical model from these 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 the effects for different nutrients. For better overview, let us plot the graph of typical nutritional effects. This visualization shows mean of posterior for 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

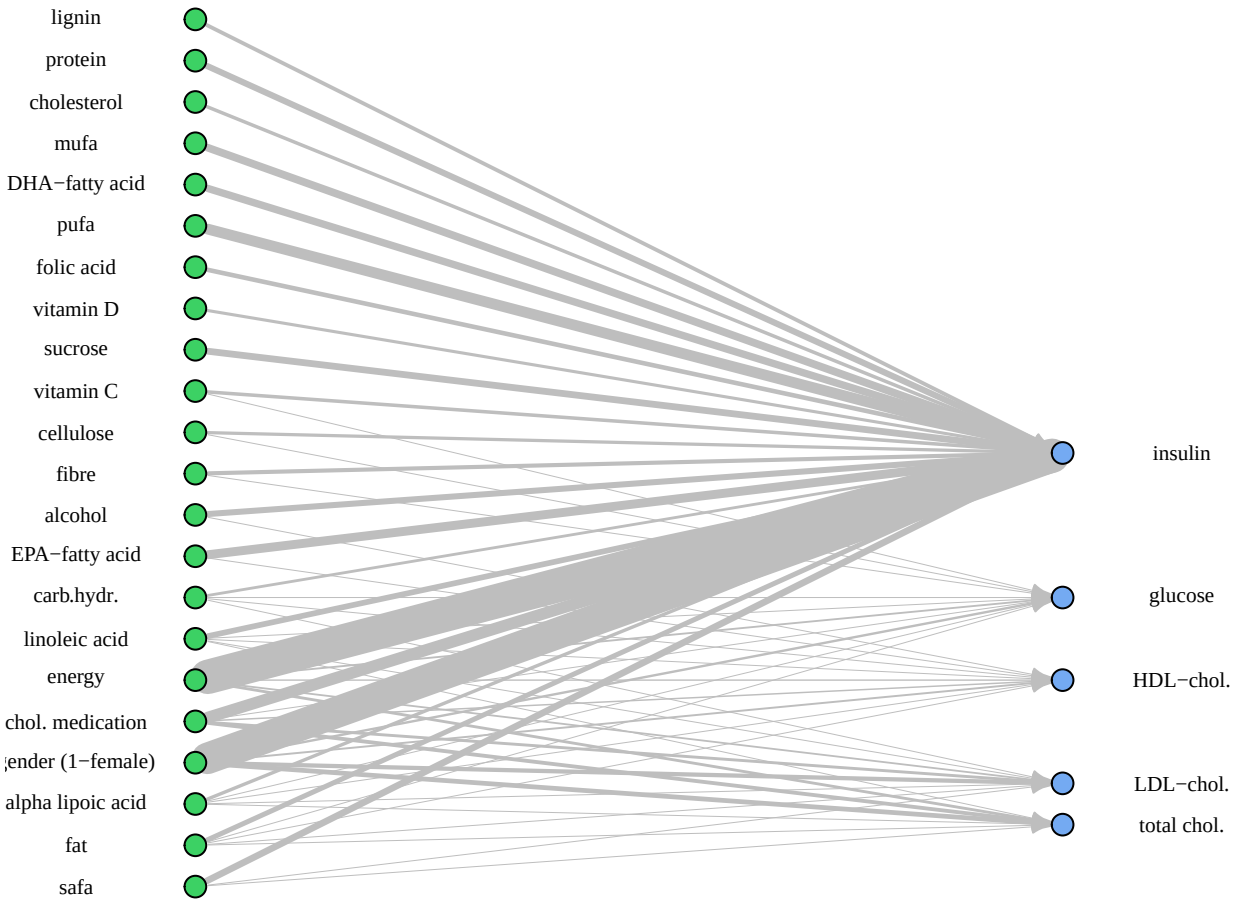


We can examine closer the effects in the previous graph. In following, we show 90% credible interval for each effect and sort them by variance between persons. These effects with high variance between persons are intresting to us.



These are the effect with most difference between patients. Let us the examine closer those of the effects that have most probable variance over 0.30 and plot them as a graph

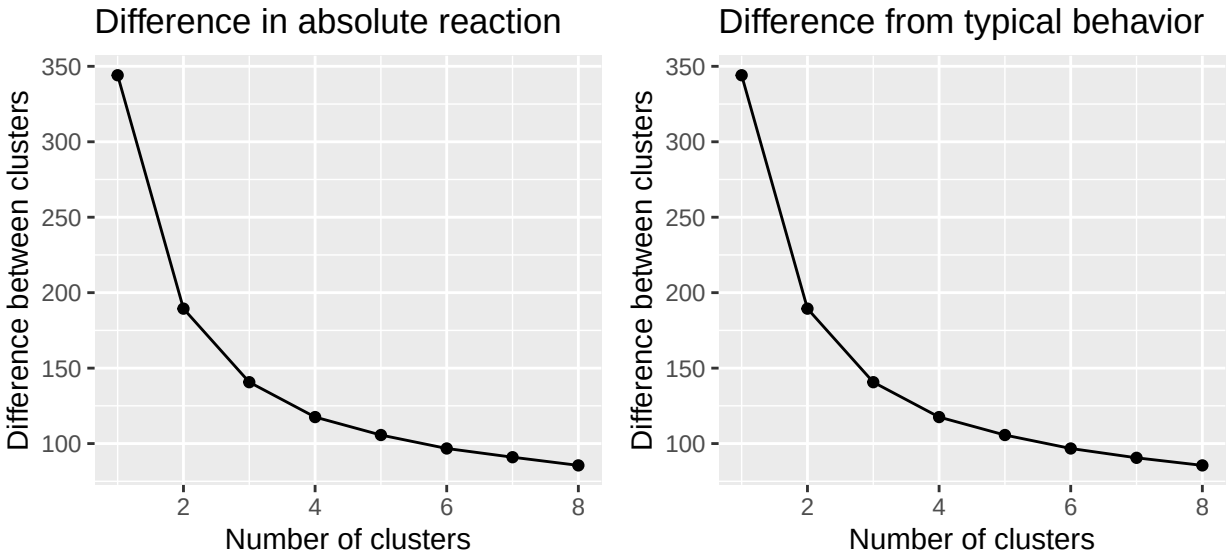
```
source("mebn/MEBN.r")
effects_with_most_variance <- ordered_typical_effects[ordered_typical_effects$variance >= 0.30,]
effects_with_most_variance$effect <- effects_with_most_variance$effect_text
number_of_varying_effects <- nrow(effects_with_most_variance)

mebn.plot_personal_variations(sysdimet_expfam_ar1, number_of_varying_effects)
```



Finding the clusters of personal reaction types

Although there are personal differences, it is likely that everyone is not behaving 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.

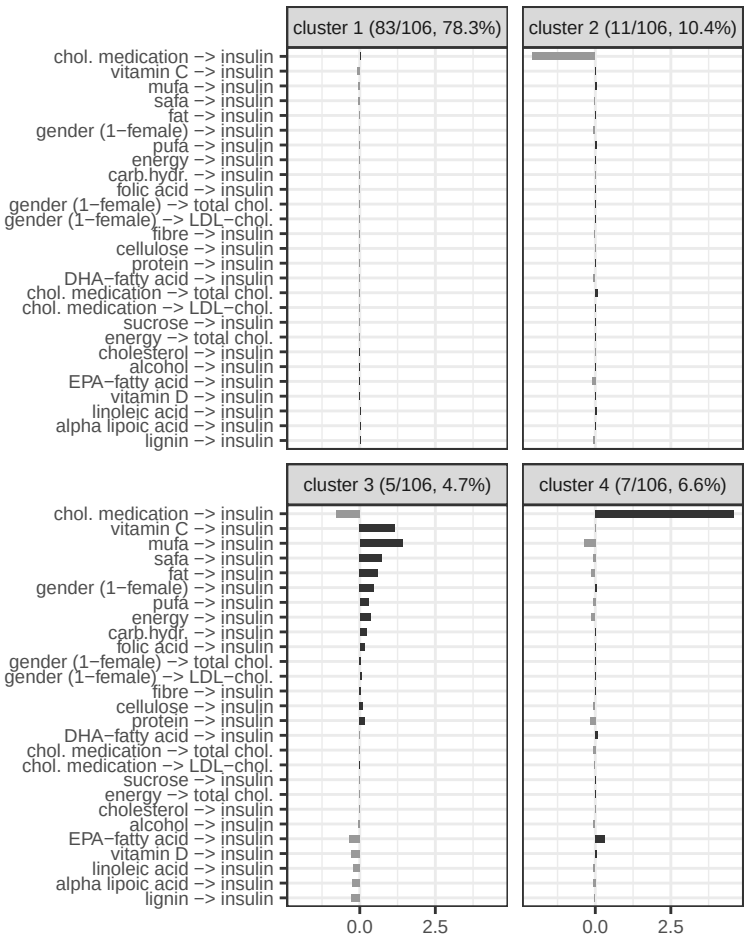
We can now take the personal estimations of the effects and see, if they form clusters



By looking previous diagram we see that there are four clearly identifiable groups between patients.

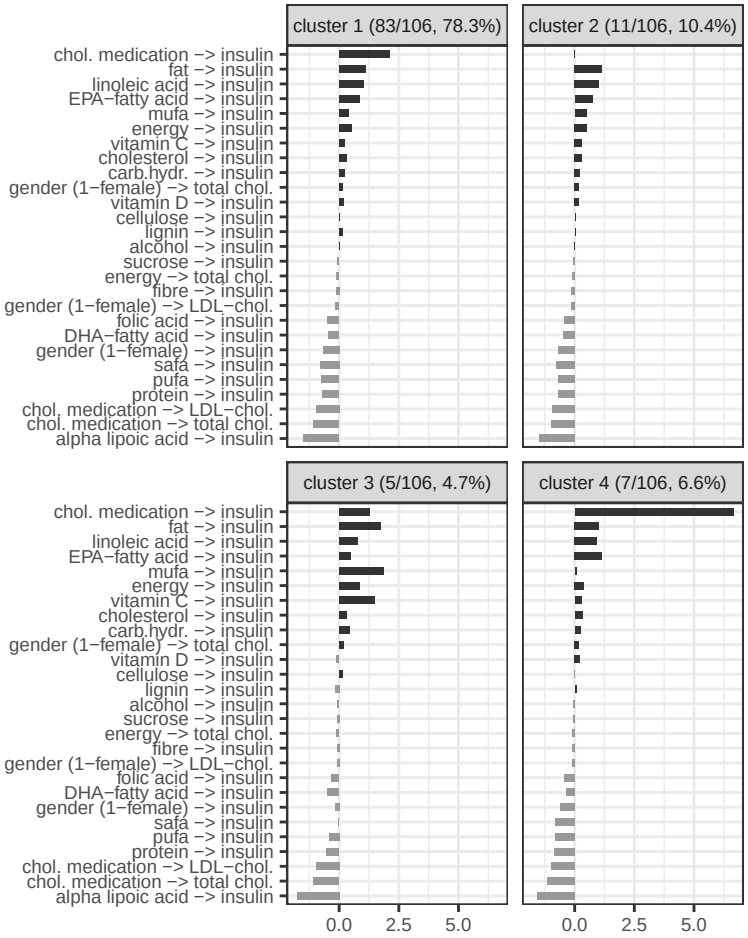
The effect of cholesterol medication

At this plot, zero of X-axis denotes a typical behaviour of that particular reaction, and 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.

	1	**2**	**3**	**4**
No medication	288	0	20	0
Chol. med.	44	44	0	28

This indicates that patients in clusters 2 and 4 are indeed taking cholesterol medication.

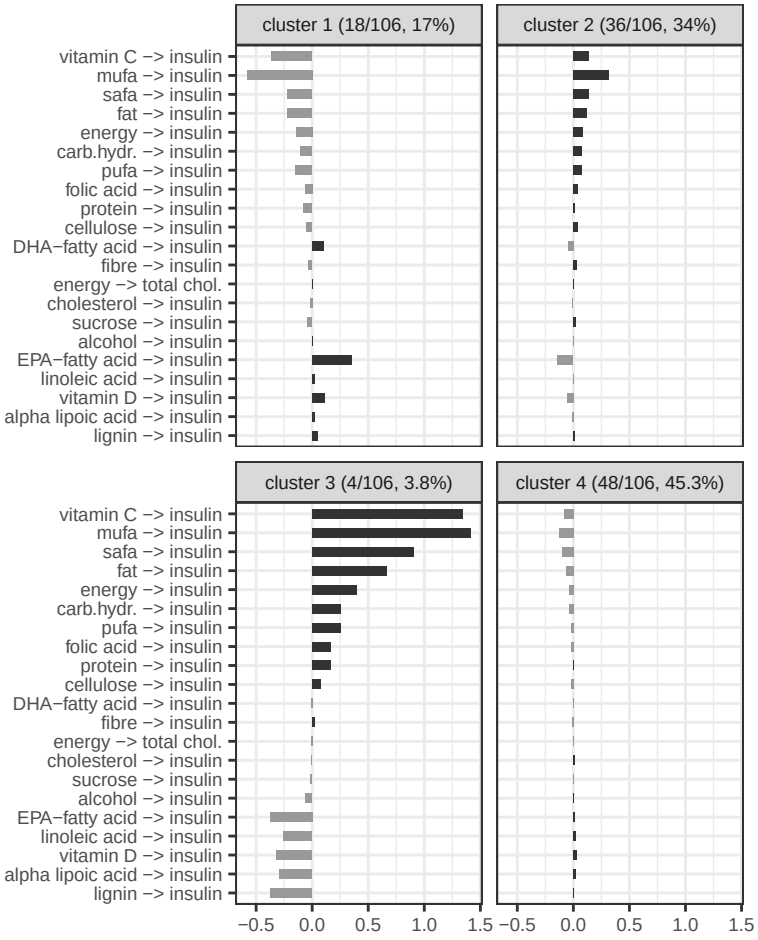
The average blood insulin levels in these clusters are

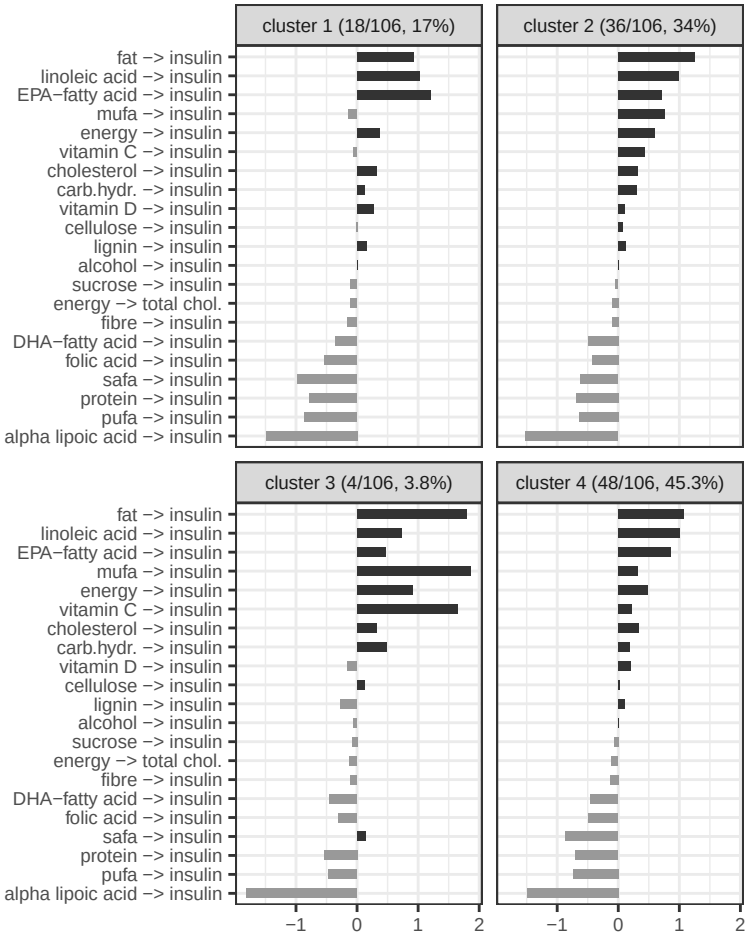
cluster	**average blood insulin**
1	10.948193
2	8.070455
3	25.482500
4	23.110714

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.

Clustering with nutritional effects only

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.



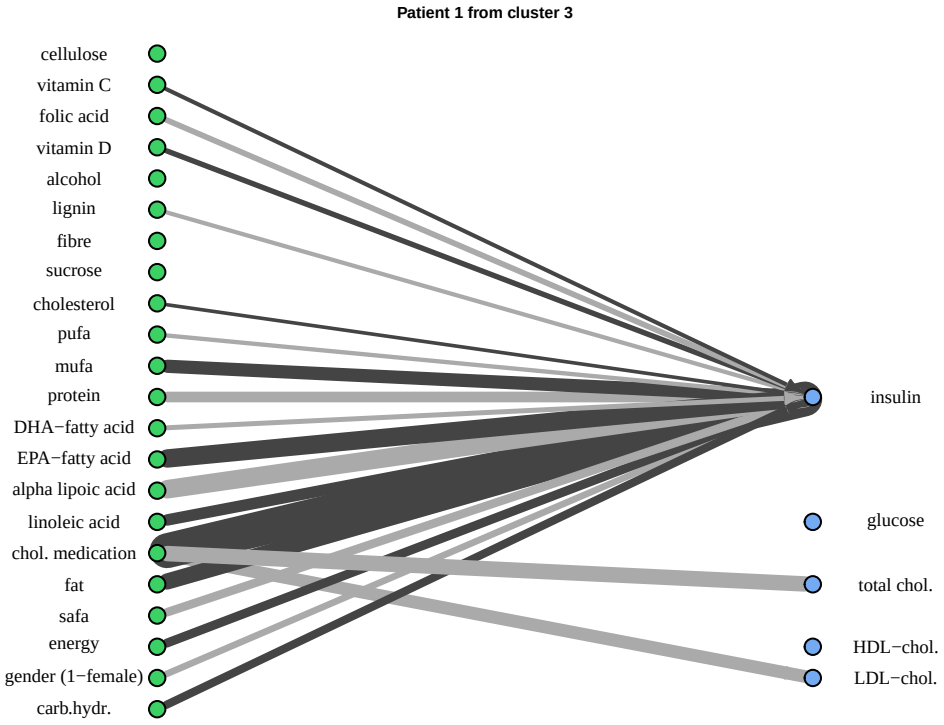


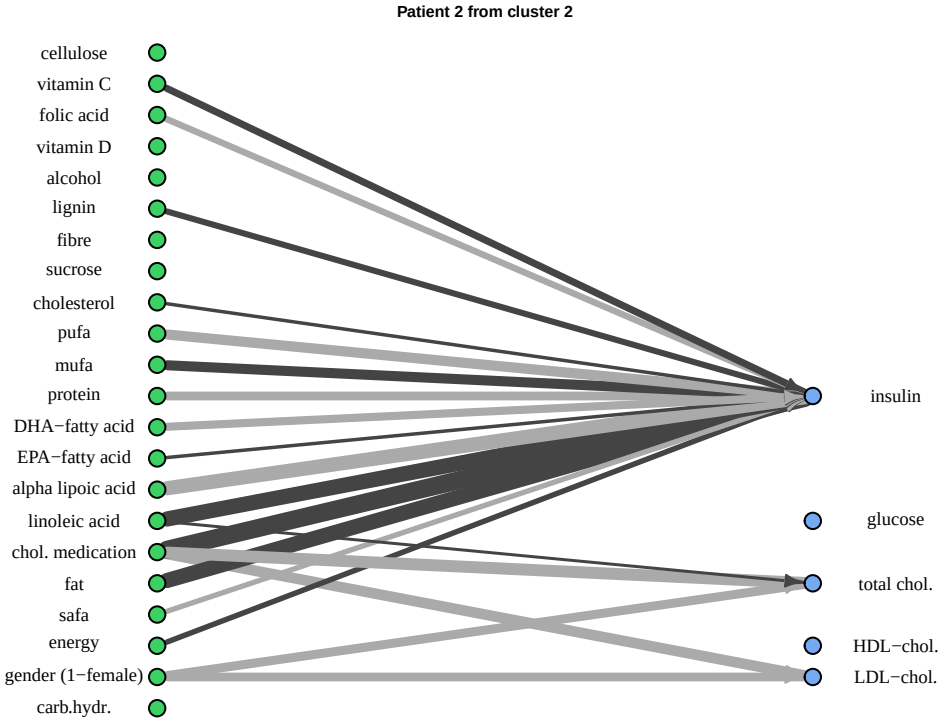
In clusters 1 and 2 there are patients whose blood insulin levels react on lower and on higher than average.

Personally predicted reaction types

During the cross-validation we estimated personal models for each patient. Next we illustrate the differences within personal reaction types by picking few most distant ones.

We can then compare the visualizations of the personal reaction graphs. The effect of cholesterol medication is most significant difference, but other differences exist as well.


Patient 1 from cluster 3
cellulose
vitamin C
folic acid
vitamin D
alcohol
lignin
fibre
sucrose
cholesterol
pufa
mufa
protein
DHA−fatty acid
EPA−fatty acid
alpha lipoic acid
linoleic acid
chol. medication
fat
safa
energy
gender (1−female)
carb.hydr.
insulin
glucose
total chol.
HDL−chol.
LDL−chol.


Patient 2 from cluster 2
cellulose
vitamin C
folic acid
vitamin D
alcohol
lignin
fibre
sucrose
cholesterol
pufa
mufa
protein
DHA−fatty acid
EPA−fatty acid
alpha lipoic acid
linoleic acid
chol. medication
fat
safa
energy
gender (1−female)
carb.hydr.
insulin
glucose
total chol.
HDL−chol.
LDL−chol.

References

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.

Limpert, Werner A., Eckhard AND Stahel. 2011. "Problems with Using the Normal Distribution – and Ways to Improve Quality and Efficiency of Data Analysis." *PLOS ONE* 6 (7). Public Library of Science: 1–8. https://doi.org/10.1371/journal.pone.0021403.