Personal recommendation of potassium and phosphorous intakes for end-stage renal patients

Supplementary Materials

JARI TURKIA [1,2] URSULA SCHWAB [3,4] VILLE HAUTAMÄKI [1,5]
[1] School of Computing, University of Eastern Finland, 80101 Joensuu, Finland
[2] CGI Suomi, Joensuu, Finland
[3] School of Medicine, Institute of Public Health and Clinical Nutrition, University of Eastern Finland, Kuopio, Finland
[4] Department of Medicine, Endocrinology and Clinical Nutrition, Kuopio University Hospital, Kuopio, Finland
[5] Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Abstract

This notebook is a supplementary material for the article

This notebook repeats fully the analysis of personal diet recommendations described in the main article. The notebook starts from the collected raw data, prepares it for analysis and estimates personal reaction models with these data. The personal reaction models are combined with the estimation of the current personal diets for constructing personal graphical models. These personal models generate the levels of blood concentrations when a diet is given, they are used in simulating recommended personal diets for reaching predefined normal concentrations. Finally, these personal diet recommendations are compared for showing the divergence among the studied patients.

The notebook execution generates all the figures and tables that are included in the article, and produces also the referenced supplementary figures. The article is accompanied with a PDF rendetion of the notebook that shows all the supplementary figures and important parts of the program code so that the analysis can be followed in detail. The executable RMarkdown notebook with data can be found in a public Github repository of the corresponding author.

Dialysis patient data

The analyzed dataset consists of food records and laboratory measurements from end-stage renal patients in dialysis. Following nutrients are possible considered as predictors of concentrations. Note that we use energy-% of fats and protein as unit in our analysis. This is referred as Table 1 in the main article.

Supplementary Table S 1: Nutrient predictors of the model.

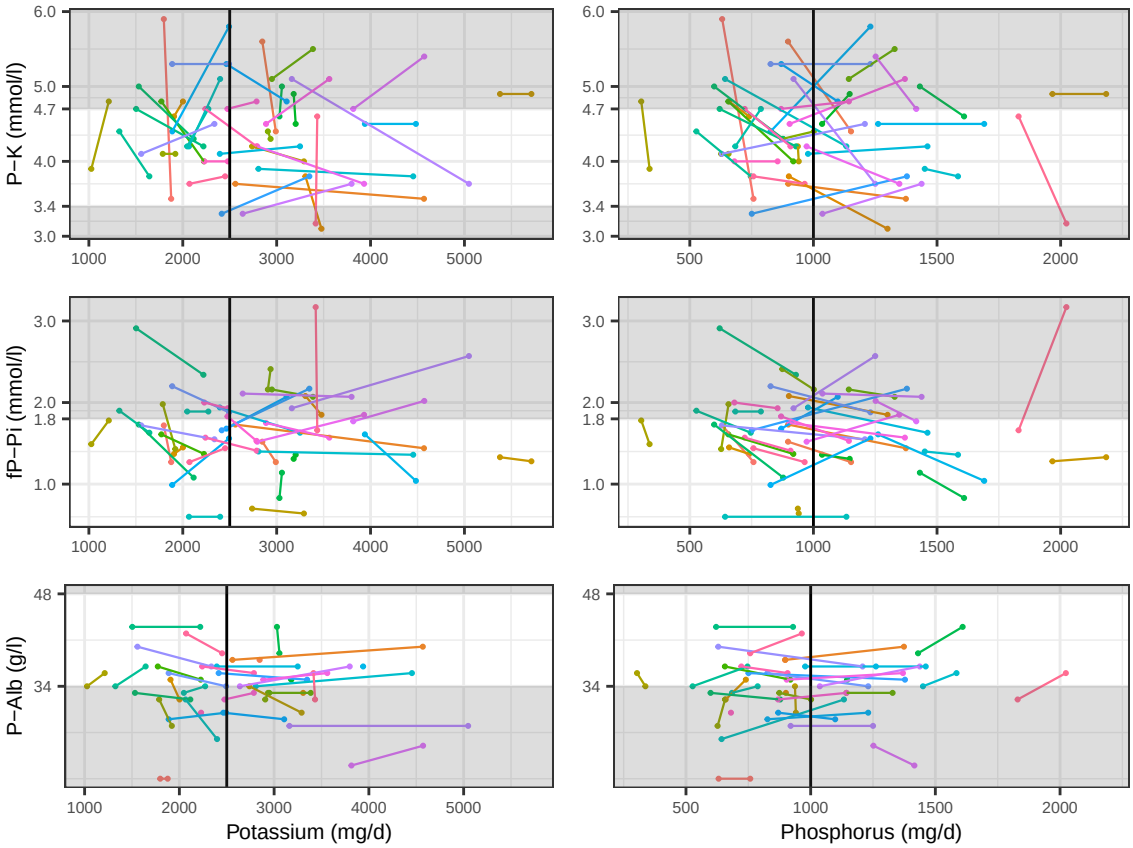
Nutrient	**Sample avg. (min-max)**
Carbonhydrates, E%	43.6 (27.1 - 63.6) E%
Fat E%	38.9 (23.4 - 54.1) E%
Monounsaturated Fatty Acids, E%	14.7 (5.6 - 25.1) E%
Polyunsaturated Fatty Acids, E%	7.1 (2.2 - 15.8) E%
Protein, E%	15.1 (9.2 - 22.4) E%
Saturated Fatty Acids, E%	13.7 (5.9 - 24.5) E%
Fiber	17 (5 - 42) g/d
Protein, g/kg	0.8 (0.2 - 2.1) g/kg/d
Energy, kcal/kg	21.8 (5.6 - 58.6) kcal/kg
Calsium	570 (123 - 1741) mg/d
Natrium	2588 (813 - 5487) mg/d
Phosphorous	**1042 (304 - 2184) mg/d**
Potassium	**2785 (1026 - 5713) mg/d**
Salt	6560 (201 - 13863) mg/d
Water	1804 (601 - 3613) ml/d
Vitamin D	8 (0 - 31) ug/d

and also following personal details and medication are considered as predictors. This is referenced as Table 2 in the article.

Supplementary Table S 2: Personal details that are used as predictors.

Personal detail	**Percentage of patients**
Blood fat medication	68%
Diabetes medication	51%
Hydroxycholecalciferol	43%
Phosphate binder med.	22%
Renavit	97%
Gender	41% female
Home hemodialysis	24%
Hospital hemodialysis	57%
Peritoneal dialysis	19%

Let us explore then how the intake of dietary potassium and phosphorus corresponds in data to plasma concentrations of potassium, phosphate and albumin



Supplementary Figure S 1: Figure shows the progress between two observations of plasma potassium (P-K), fasting plasma phosphate (fP-Pi) and plasma albumin (P-Alb) concentrations of the studied patients. White regions in the panels show the recommended concentration levels, P-K 3.4 - 4.7 mmol/l, fP-Pi < 1.8 mmol/l and P-Alb 34 - 48 g/l. Vertical black lines denote the commonly recommended maximum intakes of these nutrients. The goal is to find personal intake levels that keep the concentrations in recommended levels, if possible. The figure is plotted with ggplot2 package for R language (v 3.3.5, https://ggplot2.tidyverse.org).

Aim of this analysis is to find such personal levels of potassium and phosphorous intakes that keep all these concentrations in their normal levels marked with white regions, if possible.

Development of nutrition reaction model

In this work we construct personal generative models for concentrations that allow conditioning personal diet recommendations. Essential part of these models is the personal reactions to nutrients and other predictors.

Simultaneous reactions on all considered concentrations are modeled as multivariate model that has all the concentrations are response variables.

For comparison, we estimate multivariate systems with and without cross-model covariance. With cross-model covariance estimated, the model corresponds to seemingly unrelated model system, and without the model is Bayesian network with separated local distributions. This simpler model is estimated first with only potassium and phosphorous concentrations as responses.

```
initial_graph <- mebn.fully_connected_bipartite_graph(datadesc_fat_epros)

pk_fppi_targets <- datadesc_fat_epros[datadesc_fat_epros$Name %in% c('pk','fppi'),]

dialdiet_gamma <- mebn.bipartite_model(reaction_graph = initial_graph,
                                       inputdata = dialysis,
                                       predictor_columns = assumedpredictors_fat_epros,
                                       assumed_targets = pk_fppi_targets,
                                       group_column = "potilas",
                                       local_estimation = mebn.sampling,
                                       local_model_cache = "models/BLMM_gamma_separate",
                                       stan_model_file = "mebn/v2/BLMM_gamma.stan",
                                       normalize_values = TRUE)

write.graph(dialdiet_gamma, "graphs/dialysis_gamma_separate.graphml", "graphml")
```

In Bayesian network both responses were estimated separately. Next, we estimate a multivariate model where both distribution are estimated during single sampling. It does not factorize into separate distributions, but allows using more data.

```
initial_graph <- mebn.fully_connected_bipartite_graph(datadesc_fat_epros)

pk_fppi_targets <- datadesc_fat_epros[datadesc_fat_epros$Name %in% c('pk','fppi'),]

dialdiet_gamma_mv2_epros <- mebn.bipartite_multivariate(reaction_graph = initial_graph,
                                       inputdata = dialysis,
                                       predictor_columns = assumedpredictors_fat_epros,
                                       assumed_targets = pk_fppi_targets,
                                       group_column = "potilas",
                                       local_estimation = mebn.multivariate_sampling,
                                       local_model_cache =
                                         "models/BLMM_gamma_qr_multivariate2/fat_epros",
                                       stan_model_file = "mebn/v2/BLMM_gamma_qr_mv.stan",
                                       normalize_values = TRUE)

write.graph(dialdiet_gamma_mv2_epros,
            "graphs/dialysis_gamma_multivariate2_epros.graphml", "graphml")
```

Next we like to add plasma albumin concentration as a third constraint in the model, but unfortunately 8 of 37 patients have missing albumin measurements. From such a small dataset we don't want to remove any patients and so we predict the missing albumin levels and impute them to dataset.

For prediction, estimate the model without missing values in P-Alb. The rows with missing values are held out from the density estimation, but patients are kept in the model for estimating their parameters.

```
pk_fppi_palb_targets <-
  datadesc_fat_epros[datadesc_fat_epros$Name %in% c('pk','fppi','palb'),]

# 0/1-index for palb = NA
holdout_index <- as.vector(as.numeric(is.na(dialysis$palb)))

# Stan does not support NA in data (in Y), so let's change NA to magic number
if (anyNA(dialysis$palb)) dialysis[is.na(dialysis$palb),]$palb <- -1

initial_graph <- mebn.fully_connected_bipartite_graph(datadesc_fat_epros)

dialdiet_gamma_mv3_missing_palb <- mebn.bipartite_multivariate(
                                   reaction_graph = initial_graph,
                                   inputdata = dialysis,
                                   targetdata = holdout_index,
                                   predictor_columns = assumedpredictors_fat_epros,
                                   assumed_targets = pk_fppi_palb_targets,
                                   group_column = "potilas",
                                   local_estimation = mebn.multivariate_sampling,
                                   local_model_cache =
                                     "models/BLMM_gamma_qr_multivariate3/fat_epros_missing_palb",
                                   stan_model_file = "mebn/v2/BLMM_gamma_qr_mv_cv.stan",
                                   normalize_values = TRUE)
```

Personal data imputation models

Now we can create personal models for patients that have missing plasma albumin measurements. These models are then used for making personal predictions for replacing missing values.

```
# Extract personal generative models for patients who have missing P-Alb values

patients_with_missing_palb <- unique(as.vector(dialysis[is.na(dialysis$palb),]$potilas))

for (person_id in 1:length(patients_with_missing_palb)) {

  # - initial graph structure
  initial_graph <- mebn.fully_connected_bipartite_graph(datadesc_fat_epros)

  # - previously estimated graphical model with all the persons
  local_distributions <- pk_fppi_palb_targets
  local_distributions$modelcache <-
    "models/BLMM_gamma_qr_multivariate3/fat_epros_missing_palb"

  # - get personal data, normalized and original

  # - statistics for vertex levels are calculated from normalized data
  predictors <- nrow(assumedpredictors_fat_epros)
  normalized_input <- sapply(1:predictors, mebn.scale_gaussians,
                             data = dialysis, datadesc = assumedpredictors_fat_epros)
  normalized_input_df <- as.data.frame(normalized_input)

  # - pick rows for the selected person
```

```
  subject_code <- patients_with_missing_palb[person_id]
  personal_data_df <- cbind(dialysis$potilas, normalized_input_df)
  personal_data_df <- personal_data_df[personal_data_df$`dialysis$potilas` == subject_code,]
  personal_data <- as.matrix(subset(personal_data_df, select = -c(`dialysis$potilas`)))

  # - store also these original stats in graph
  personal_data_org <- subset(dialysis[dialysis$potilas == subject_code,],
                              select = as.vector(assumedpredictors_fat_epros$Name))

  # function parameters

  personal_model_dir <- paste0("data_imputation_models/", subject_code)

  # Generate a personal graph in directory

  # - This function extracts the personal model from a single level multivariate model

  personal_graph <- mebn.extract_personal_graph_from_mv(person_id, initial_graph,
                    personal_model_dir, assumedpredictors_fat_epros, pk_fppi_palb_targets,
                    "models/BLMM_gamma_qr_multivariate3/fat_epros_missing_palb",
                    personal_data, personal_data_org, datadesc_fat_epros)
}

datadesc <- datadesc_fat_epros
```

Then we predict missing values with these data imputing models

```
library(igraph)
source("mebn/v2/MEBNv2.r")

# this vector is imputed with predicted values
original_palb <- dialysis$palb
imput_length <- length(original_palb)

# Normalize input for predictions (as it was in the likelihood estimation)
predictors <- nrow(assumedpredictors_fat_epros)
normalized_input <- sapply(1:predictors, mebn.scale_gaussians,
                           data = dialysis, datadesc = assumedpredictors_fat_epros)
normalized_input_df <- as.data.frame(normalized_input)

normalized_input_df <- cbind(dialysis$potilas, normalized_input_df)
normalized_input_df <- cbind(dialysis$palb, normalized_input_df)

colnames(normalized_input_df) <- c("palb", "potilas", assumedpredictors_fat_epros$Name)
rows_with_missing_palb <- normalized_input_df[is.na(normalized_input_df$palb),]

for (i in 1:nrow(rows_with_missing_palb)) {

  datarow <- rows_with_missing_palb[i,]

  # get model for patient in this datarow

  personal_model_dir <- paste0("data_imputation_models/",datarow$potilas)
```

```
  print(paste0("Reading personal data imputation model '", personal_model_dir,
               "/personal_graph.graphml'"))

  personal_graph <- read.graph(paste0(personal_model_dir, "/personal_graph.graphml"), "graphml")

  # and use the data in this row to predict palb

  evidence <- rows_with_missing_palb[i, assumedpredictors_fat_epros$Name]

  posterior_prediction <- mebn.personal_prediction(reaction_graph = personal_graph,
                                graph_dir = personal_model_dir,
                                evidence = evidence,
                                stan_model_file = "diet/posterior_prediction.stan")

  # personal_predictions contains predictions multivariate predictions
  # P-K and fP-Pi are known, but P-Alb is missing

  posterior <- rstan::extract(posterior_prediction, par= "posterior[3]")

  # use predicted posterior mean for missing P-Alb value
  # - this is i:th NA value in dialysis
  predicted_palb <- mean(posterior$`posterior[3]`)

  # rows in 'rows_with_missing_palb' and 'original_palb' are in same order
  # so we can imput in the NA values at 'original_palb'

  for (m in 1:imput_length)
  {
      if (is.na(original_palb[m]))
      {
          original_palb[m] <- predicted_palb
          break;
      }
  }

}

dialysis2 <- dialysis

dialysis2$palb <- original_palb

saveRDS(dialysis2, "data/DIALYSIS_imputed_palb.rds")
write.csv2(dialysis2, file="data/DIALYSIS_imputed_palb.csv", row.names = FALSE, sep = ";")
```

With this data imputation we can estimate a cross-covariance model with three responses

```
dialysis_imputed <- readRDS("data/DIALYSIS_imputed_palb.rds")

pk_fppi_palb_targets <- datadesc_fat_epros[datadesc_fat_epros$Name %in% c('pk','fppi','palb'),]

no_holdout <- rep(0, nrow(dialysis_imputed))

initial_graph <- mebn.fully_connected_bipartite_graph(datadesc_fat_epros)
```

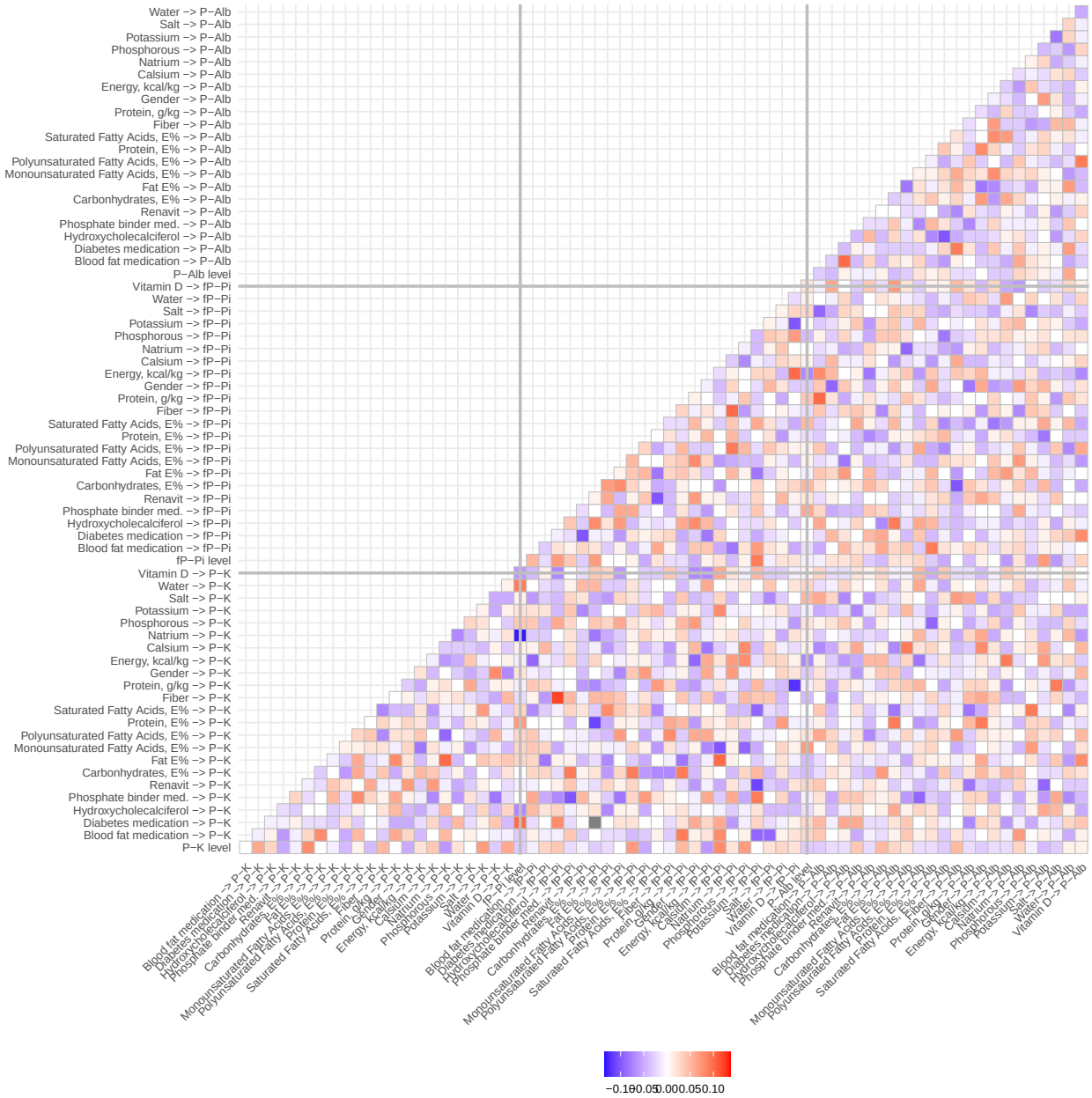
```
dialdiet_gamma_mv3 <- mebn.bipartite_multivariate(reaction_graph = initial_graph,
                                  inputdata = dialysis_imputed,
                                  targetdata = no_holdout,
                                  predictor_columns = assumedpredictors_fat_epros,
                                  assumed_targets = pk_fppi_palb_targets,
                                  group_column = "potilas",
                                  local_estimation = mebn.multivariate_sampling,
                                  local_model_cache =
                                    "models/BLMM_gamma_qr_multivariate3/imputed_palb",
                                  stan_model_file = "mebn/v2/BLMM_gamma_qr_mv_cv.stan",
                                  normalize_values = TRUE)

write.graph(dialdiet_gamma_mv3,
            "graphs/dialysis_gamma_multivariate3_imputed.graphml", "graphml")
```

Finally, we consider also the effect of dialysis treatment type as new level of grouping in data. This allows us to estimate average effects of nutrition for each treatment type and then also personal effects within those treatments.

```
dialysis_imputed <- readRDS("data/DIALYSIS_imputed_palb.rds")

# add the dialysis treatment type as a grouping factor
dialysis_imputed$hoitoryhma <- as.factor(dialysis_imputed$hoitomuoto)

# and sort the data by treatment/patient/observation
dialysis_imputed <- dialysis_imputed[order(dialysis_imputed$hoitoryhma, dialysis_imputed$potilas, dialys

pk_fppi_palb_targets <- datadesc_fat_epros[datadesc_fat_epros$Name %in% c('pk','fppi','palb'),]

no_holdout <- rep(0, nrow(dialysis_imputed))

initial_graph <- mebn.fully_connected_bipartite_graph(datadesc_fat_epros)

dialdiet_gamma_mv3_two_level <- mebn.bipartite_two_level_multivariate(reaction_graph = initial_graph,
                                  inputdata = dialysis_imputed,
                                  targetdata = no_holdout,
                                  predictor_columns = assumedpredictors_fat_epros,
                                  assumed_targets = pk_fppi_palb_targets,
                                  group_column = "hoitoryhma",
                                  subject_column = "potilas",
                                  local_estimation = mebn.two_level_multivariate_sampling,
                                  local_model_cache = "models/BLMM_gamma_mv_cross/two_levels",
                                  stan_model_file = "mebn/v2/BLMM_gamma_two_level_grouping.stan",
                                  normalize_values = TRUE)

write.graph(dialdiet_gamma_mv3_two_level, "graphs/dialysis_gamma_two_level_grouping.graphml", "graphml"
```

Model analysis

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```



Supplementary Figure S 2: Correlation plot of matrix C that includes the estimated within-model and cross-model correlations of personal effects

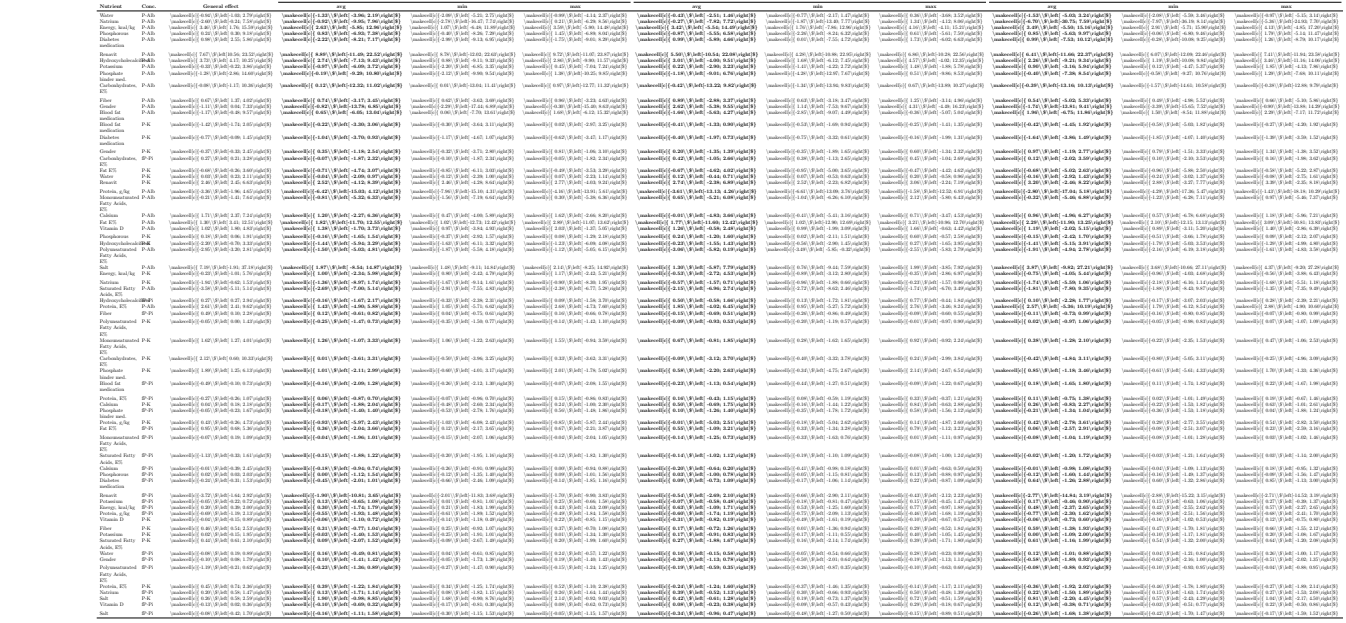
Supplementary Table S 3: Table shows 30 highest positive or negative correlations between personal effects of nutrients or other considered predictors. This structure of correlations is used in estimating the personal effects based on personal intake and matching concentrations.

effect1	effect2	corr
Diabetes medication -> P-K	Carbonhydrates, E% -> fP-Pi	0.136
Fiber -> P-K	Hydroxycholecalciferol -> fP-Pi	0.116
Fat E% -> P-K	Natrium -> P-K	0.105
Fat E% -> P-K	Calsium -> fP-Pi	0.102
Diabetes medication -> P-K	fP-Pi level	0.100
Fiber -> fP-Pi	Natrium -> fP-Pi	0.097
Natrium -> fP-Pi	Fiber -> fP-Pi	0.097
Blood fat medication -> P-Alb	Hydroxycholecalciferol -> P-Alb	0.096
Protein, g/kg -> fP-Pi	Blood fat medication -> P-Alb	0.095
Phosphate binder med. -> P-K	Potassium -> fP-Pi	0.094
Carbonhydrates, E% -> P-K	Phosphate binder med. -> fP-Pi	0.093
Phosphate binder med. -> fP-Pi	Carbonhydrates, E% -> P-K	0.093
Blood fat medication -> P-K	Protein, g/kg -> fP-Pi	0.092
Energy, kcal/kg -> P-K	Calsium -> P-Alb	0.092
Gender -> P-Alb	Protein, E% -> P-K	0.092
Energy, kcal/kg -> P-K	Gender -> fP-Pi	-0.092
Gender -> fP-Pi	Energy, kcal/kg -> P-K	-0.092
Blood fat medication -> P-K	Salt -> fP-Pi	-0.094
Natrium -> P-K	Polyunsaturated Fatty Acids, E% -> P-K	-0.094
Phosphate binder med. -> P-K	Phosphate binder med. -> fP-Pi	-0.095
Carbonhydrates, E% -> fP-Pi	Fiber -> P-Alb	-0.100
Saturated Fatty Acids, E% -> P-Alb	Hydroxycholecalciferol -> P-Alb	-0.100
Monounsaturated Fatty Acids, E% -> P-K	Calsium -> fP-Pi	-0.102
Renavit -> fP-Pi	Saturated Fatty Acids, E% -> fP-Pi	-0.103
Potassium -> fP-Pi	Vitamin D -> fP-Pi	-0.104
Diabetes medication -> fP-Pi	Renavit -> fP-Pi	-0.105
Renavit -> P-K	Potassium -> fP-Pi	-0.108
Carbonhydrates, E% -> fP-Pi	Protein, E% -> P-K	-0.108
Protein, g/kg -> P-K	Vitamin D -> fP-Pi	-0.120
Natrium -> P-K	fP-Pi level	-0.133

Effects of nutrition in both treatment and personal levels

First, we create personal models that include also information of personal treatment

Supplementary Table S 5: Variation of nutrition effects between different dialysis treatments and between patients within the same treatment.



Supplementary Table S 4: Personal graphical models with different reaction model candidates are evaluated with NRMSE for each modelled concentration and average model error.

	NRMSE			
Reaction model	**P-K**	**fP-Pi**	**P-Alb**	**average error**
mv3_cross_two_levels	0.004	0.007	0.002	0.004
mv3_cross_single_level	0.036	0.13	0.059	0.036

Then, we make a summary from effects of treatment and personal variations within the treatment

Two level effect summary

This large table in includes all estimated nutrition effects

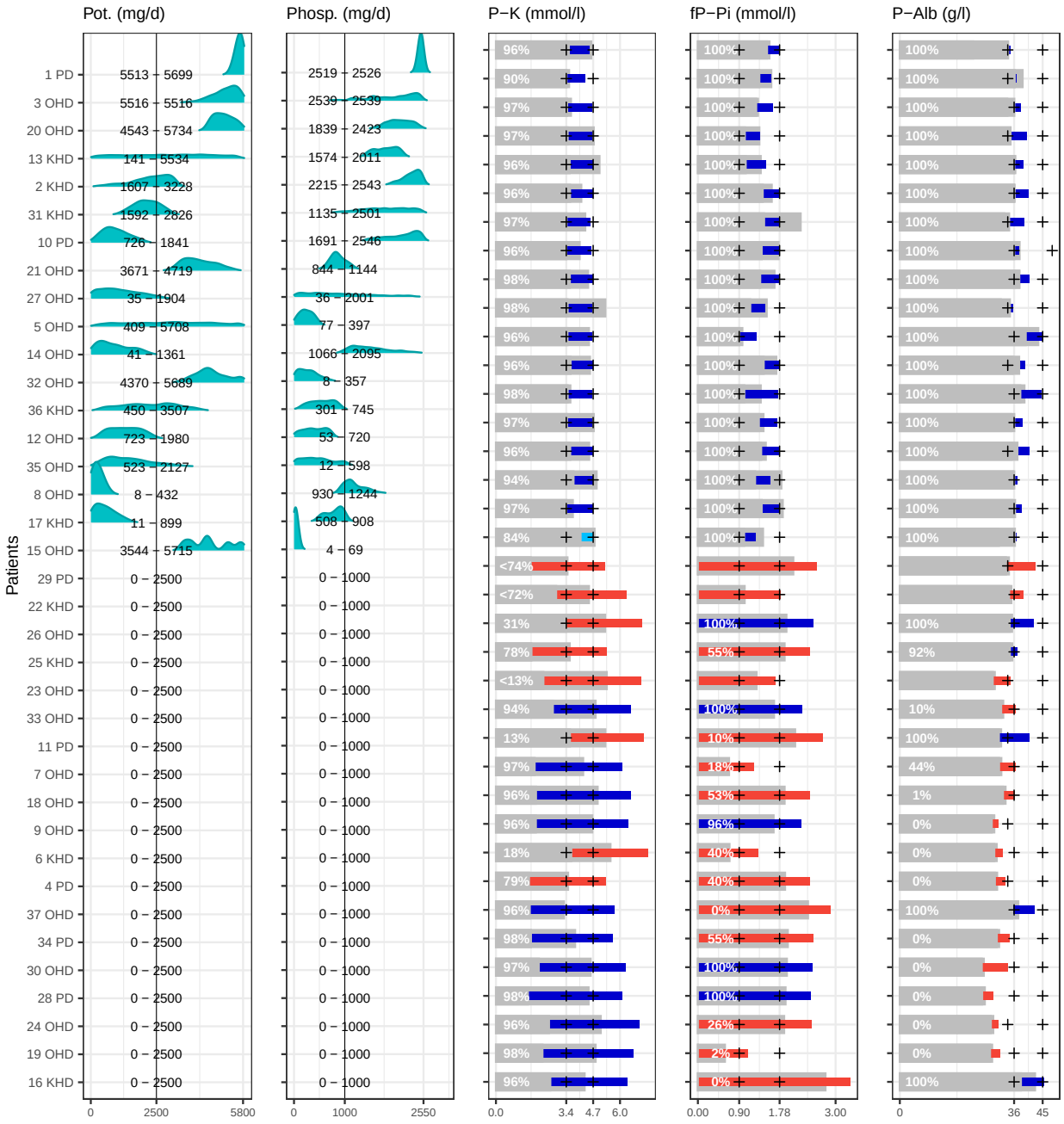
##Personal recommandations

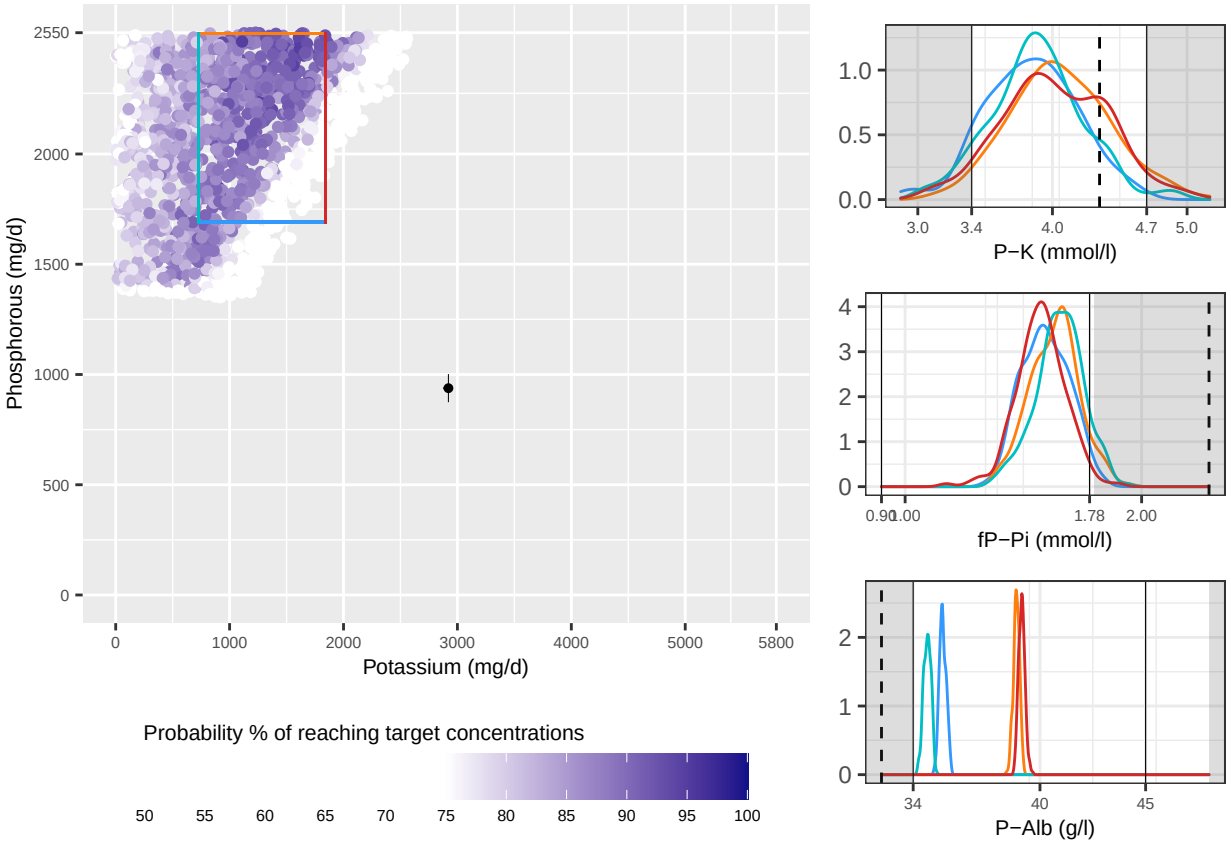
Here we query recommendations with graphical models. In recommendation, we let potassium and phosphorous intakes have values uniformly from full generally recommended range.

Recommendation plot

```
## Warning: Ignoring unknown parameters: width
```

```
## Warning: Ignoring unknown parameters: width
```


Pot. (mg/d)
Phosp. (mg/d)
P−K (mmol/l)
fP−Pi (mmol/l)
P−Alb (g/l)
Patients
1 PD
3 OHD
20 OHD
13 KHD
2 KHD
31 KHD
10 PD
21 OHD
27 OHD
5 OHD
14 OHD
32 OHD
36 KHD
12 OHD
35 OHD
8 OHD
17 KHD
15 OHD
29 PD
22 KHD
26 OHD
25 KHD
23 OHD
33 OHD
11 PD
7 OHD
18 OHD
9 OHD
6 KHD
4 PD
37 OHD
34 PD
30 OHD
28 PD
24 OHD
19 OHD
16 KHD
0
2500
5800
0
1000
2550
0.0
3.4
4.7
6.0
0.00
0.90
1.78
3.00
0
36
45


Phosphorous (mg/d)
2550
2000
1500
1000
500
0
0
1000
2000
3000
4000
5000
5800
Potassium (mg/d)
Probability % of reaching target concentrations
50
55
60
65
70
75
80
85
90
95
100
1.0
0.5
0.0
3.0
3.4
4.0
4.7
5.0
P−K (mmol/l)
4
3
2
1
0
0.90
1.00
1.78
2.00
fP−Pi (mmol/l)
2
1
0
34
40
45
P−Alb (g/l)