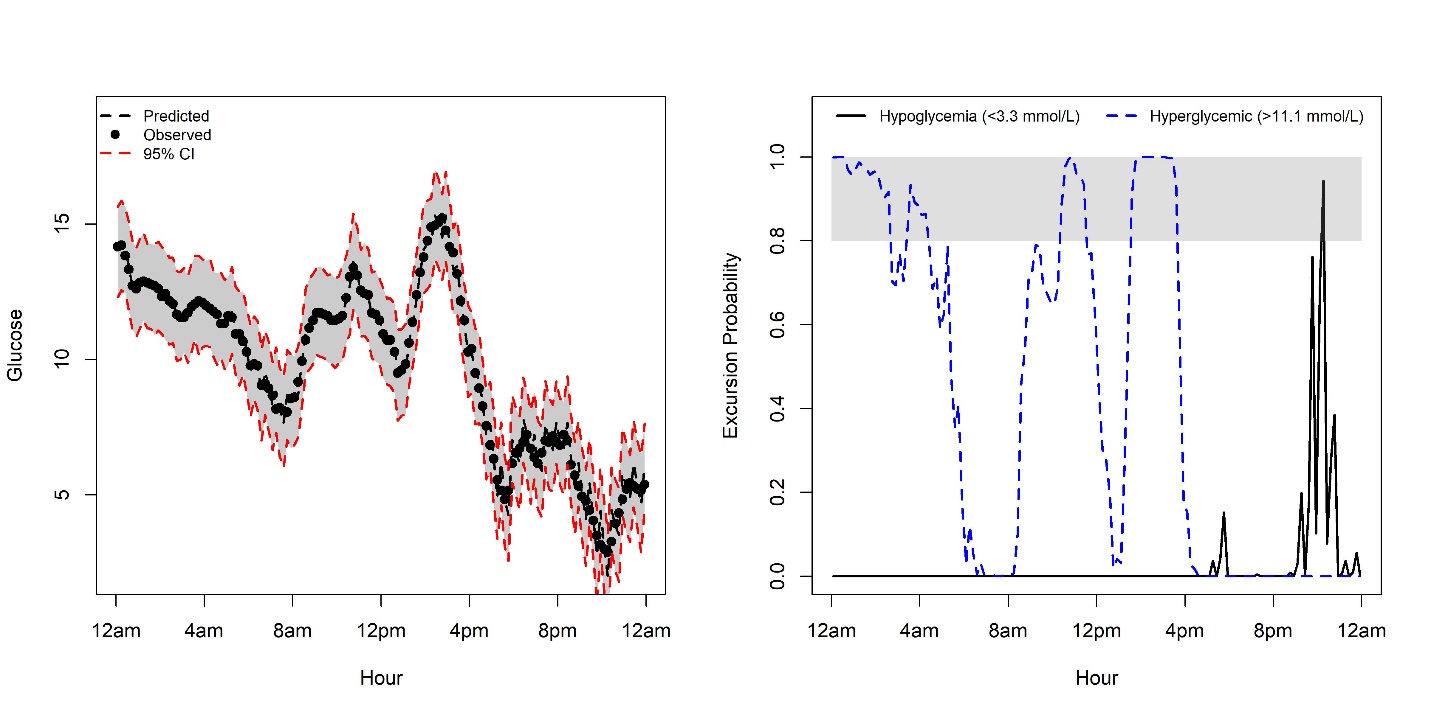
**Supplemental File**

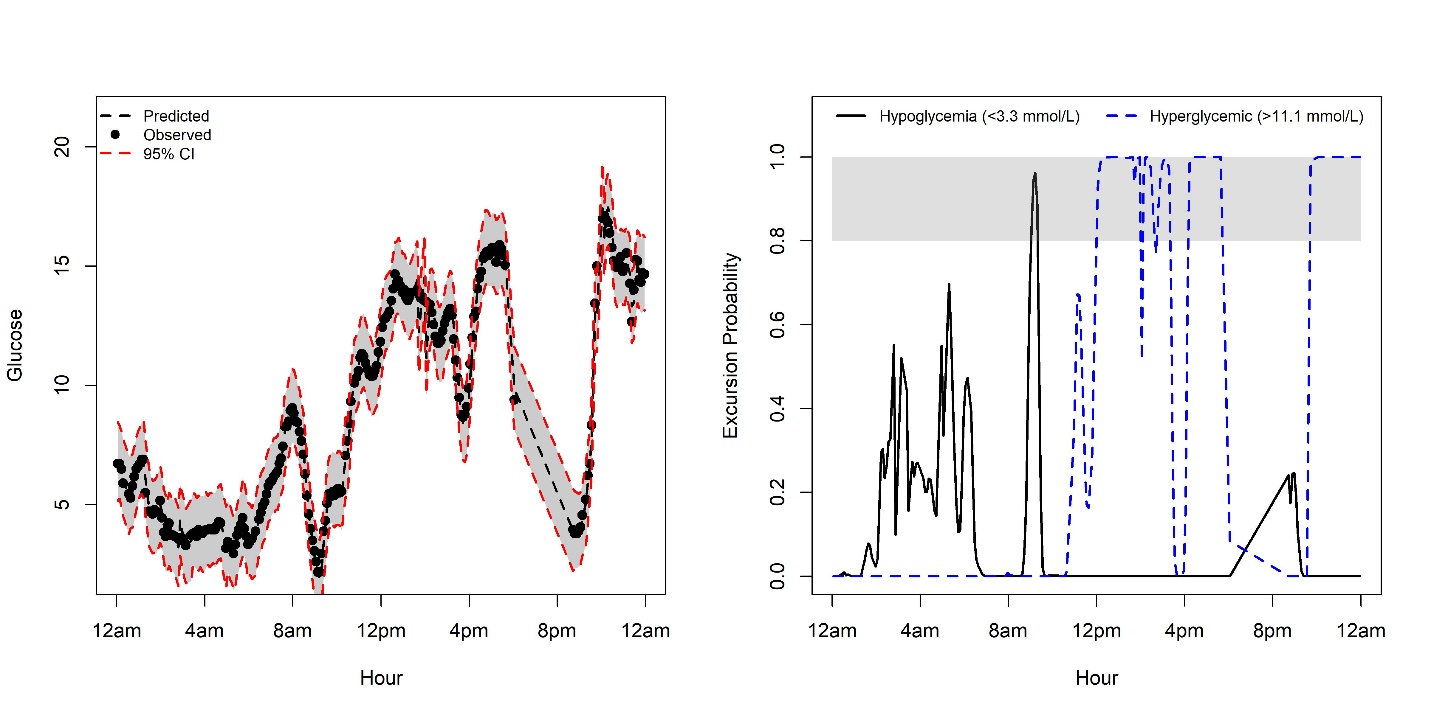
**1. Figure 4 (mmol/L)**

The graphs on the right panel of Figure 4 present the predicted risk of hypo- and hyper-glycemic excursions for the same three subjects mentioned above. We used and for identifying hypoglycemia and hyperglycemia, respectively. The first patient (A and B) was at high risk of hyperglycemia for the whole day except 6-8am and 4pm-12am. Her risk of hypoglycemia was relatively low except for around 9pm-12am in the evening. The second patient (C and D) was at high risk of hyperglycemia in the afternoon, followed by decreased risk during between 6-10pm. Her risk of hypoglycemia was quite high around 9am in the morning. The third patient (E and F) was at high risk of hyperglycemia in the afternoon during 1-5.30pm and right before midnight from 11pm-12am. His risk of hypoglycemia was quite low except between 9-10pm in the evening.

****

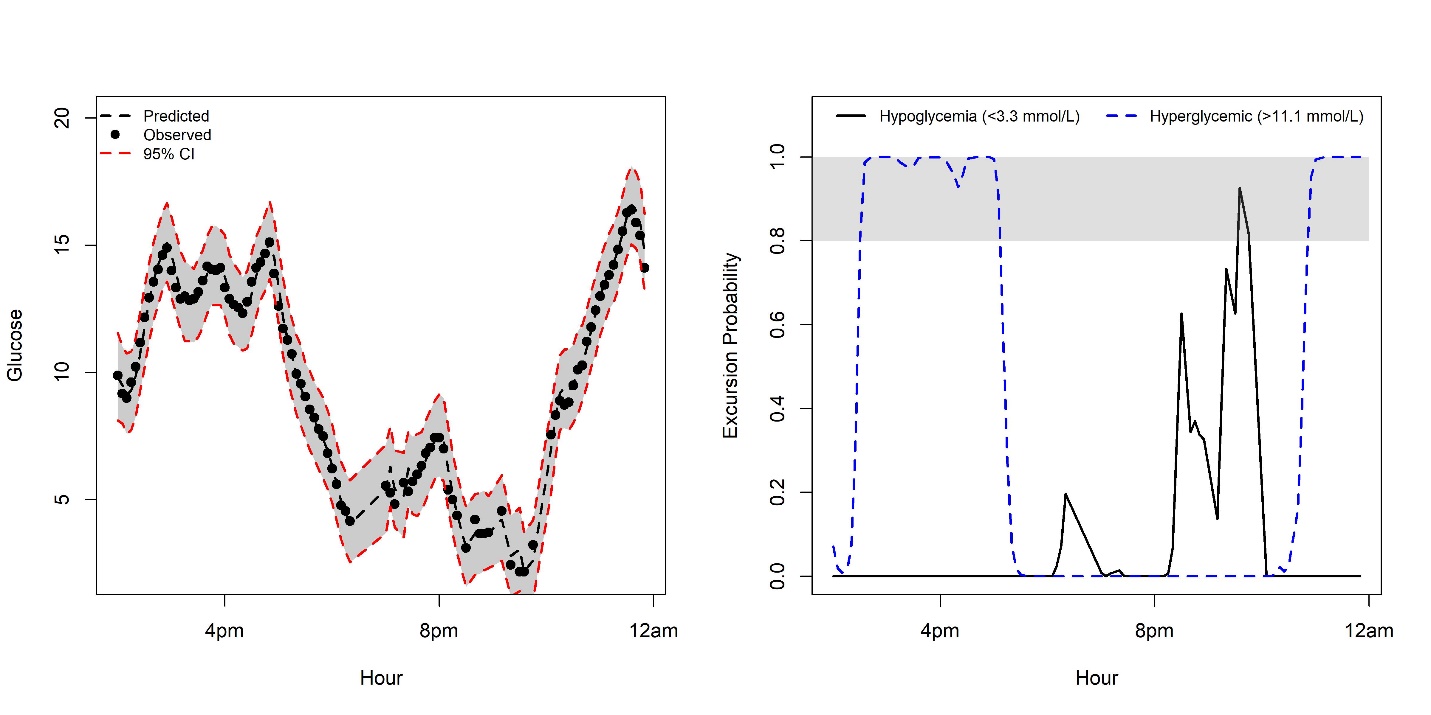
**B)**

**A)**

****

**D)**

**C)**

**Figure 4.** **Left panel**: Observed glucose readings (y-axis) from CGM (black dots) over clock time (x-axis) are shown with FD prediction (dashed line) and 95% CI (gray band with red dashed lines); **Right panel**: real-time risk for glycemic excursions (black line is the probability of hypoglycemia; blue line is the probability of hyperglycemic: gray band is the area where probabilities > 0.80 or 80%). The 1st row is for a 62-year-old white female from the control group; height: 160 cm; weight: 68 kg. The 2nd row is for an 8-year-old white female from the control group; height: 140 cm; weight: 32.8 kg. The 3rd row is for a 41-year-old white male from the RT-CGM group; height: 168 cm; weight: 79 kg.

**F)**

**E)**

**2. Computer code**

The R and Matlab script below is based on the analysis of the CGM data and variables from the dataset are described in the manuscript.

2.1) R script for FPCA results presented in section 4.1 in manuscript.

#### R script for FPCA on CGM data ####

#load packages

library(fpca)

library(scales)

#load dataset

load("data.RData") #dataset

#hrweek is in terms of hr

data<-subset(data,select=c(PtID, Glucose,hr)) #subset the data

data1<-data

e\_min <-min(data$hr)

e\_max <- max(data$hr)

#scale hr to improve the convergence and efficiency of the algorithm

#to be scaled between 0 and 1 (i.e., the unit interval)

data$hr<- rescale(data$hr,to=c(0.0001,0.999))

data\_use <- as.matrix(data)

## parameters for fpca.mle ##

ini.method = "EM"

basis.method = "bs"

sl.v = rep(0.5, 10)

max.step = 80

grid.l = seq(0, 1, 0.05)

grids = seq(0, 1, length = 125)

## fit candidate models by fpca.mle ##

result <- fpca.mle(data\_use, M.set, r.set, ini.method, basis.method, sl.v, max.step, grid.l, grids)

## after getting the results ##

grids.new <- result$grid

M<-result$selected\_model[1] #the selected M (number of basis functions)

r<-result$selected\_model[2] # the selected r (dimension of the process)

muest<-result$fitted\_mean # the estimated mean curve by local linear fitting evaluated at grid

evalest<-result$eigenvalues # the estimated eigenvalues under the selected model

sig2est<-result$error\_var # the estimated error variance under the selected model

eigenfest<-result$eigenfunctions # the estimated eigenfunctions under the selected model evaluated at grid

## derive fpc scores and look at the predicted curve ## # fpc scores

fpcs <- fpca.score(data\_use, grids.new, muest, evalest, eigenfest, sig2est, r) # get predicted trajectories on a fine grid: the same grid for which mean and eigenfunctions are evaluated

pred <- fpca.pred(fpcs, muest, eigenfest)

2.2) Matlab code for two-stage FPCA (section 4.2), the detail of this Matlab package can be found at <http://anson.ucdavis.edu/~mueller/data/pace.html> .

#Set directory

addpath(genpath('…'));

p = path;

isExist = regexp(p, 'PACE');

if isempty(isExist) == 1

addpath(genpath('../PACE/'));

end

#load data 'dataweek1.csv'

data\_week=csvread('dataweek1.csv',1,1);

data\_day =csvread('dataday1.csv',1,1);

nw=length(unique(data\_week(:,1)));

wy\_str=cell(1,nw);

wt\_str=cell(1,nw);

nd=length(unique(data\_day(:,1)));

dy\_str=cell(1,nd);

dt\_str=cell(1,nd);

for i=1:nw

wt\_str{i}=( data\_use(data\_week(:,1)==uni(i),3))';

wy\_str{i} =( data\_use(data\_week(:,1)==uni(i),2))';

end

for i=1:nd

dt\_str{i}=( data\_use(data\_week(:,1)==uni(i),3))';

dy\_str{i} =( data\_use(data\_week(:,1)==uni(i),2))';

end

% p = setOptions('yname','x', 'regular',regular,'method','IN','selection\_k', 'FVE','FVE\_threshold', 0.9,'screePlot',1, 'designPlot',1,'corrPlot',1,'numBins',0, 'verbose','on');

p = setOptions('yname','x', 'regular', regular, 'selection\_k', 'FVE','FVE\_threshold', 0.9,'screePlot',1, 'designPlot',1,'corrPlot',1,'numBins',0, 'verbose','on');

[week\_result]= FPCA(wy\_str,wt\_str,p);

[day\_result] = FPCA(dy\_str,dt\_str,p);

y1=day\_result{19}(:,(1:nd))';

y1 = cellfun(@transpose,y1,'UniformOutput',false);

%t1=day\_result{21}';

%t1 = cellfun(@transpose,t1,'UniformOutput',false);

%tval = [t1{:}];

yval = [y1{:}];

plot(yval) ;

day\_pc=day\_result{6};

median(day\_pc(:,1));

median(day\_pc(:,2));

2.3) R script for predicting real-time risk of glycemic events by using CGM data (section 3.3 and 4.3 in manuscript.)

#### R script for predicting real-time risk of glycemic events ####

#load packages

library(ngme)

library(tidyverse)

#load data

data<- readRDS("dataweek1.rds")

#subset the data; hrweek is in terms of hr

data<-subset(data,select=c(PtID, Glucose,hr,group))

fixed <- Glucose ~ hr+group

#fit model ##################################################

set.seed(123)

init\_fit\_normal <- ngme(fixed = fixed,

random = ~ 1|PtID,

data = data,

reffects = "Normal",

process = c("Normal", "fd2"),

error = "Normal",

timeVar = "hr",

nIter = 20000,

use.process = TRUE,

silent = TRUE,

mesh = list(cutoff = 3/365.25,

max.dist = 2/12,

extend = 0.01),

controls = list(pSubsample = 0.01,

subsample.type = 1,

step0 = 0.8,

estimate.fisher = FALSE,

polyak.rate = -1,

alpha = 0.01)

)

#check the convergence of the fixed effects and apply some polyak averaging to reduce variance ####

fit\_normal <- polyak.ngme(init\_fit\_normal, polyak.rate = 0.0001, plot = TRUE, param = "fixed")

#same for random effects

fit\_normal <- polyak.ngme(fit\_normal, polyak.rate = 0.00015,

plot = TRUE, param = "random")

#for process

fit\_normal <- polyak.ngme(fit\_normal, polyak.rate = 0.0002,

plot = TRUE, param = "process")

#for error term

fit\_normal <- polyak.ngme(fit\_normal, polyak.rate = 0.0001,

plot = TRUE, param = "error")

summary(fit\_normal)

######### ######## ######## Prediction and getting excursions ##############################

excursion\_ids <- deid\_data %>% select(PtID) %>% unique %>% slice(c(12)) %>% unlist

delta <- 0.01

Bf\_pred <- Br\_pred <- list()

for(i in 1:length(excursion\_ids)){

data\_i <- deid\_data[deid\_data$PtID == excursion\_ids[i], ]

n\_i <- nrow(data\_i)

Br\_pred\_i <- matrix(1, nrow = n\_i, ncol = 1)

Bf\_pred\_i <- cbind(data\_i$hr+ delta,data\_i$group

)

Br\_pred[[i]] <- Br\_pred\_i

Bf\_pred[[i]] <- Bf\_pred\_i

}

derivative\_list <- list(Bfixed = Bf\_pred,

Brandom = Br\_pred,

delta = delta)

######### excursion for Y< 60 (to get predictive probabilities for hypo-glycaemia) ############

excursions\_normal3 <-

predict(fit\_normal,

id = excursion\_ids,

controls = list(nSim = 1000,

nBurnin = 500,

silent = FALSE,

predict.derivatives = derivative\_list,

excursions = list(list(type = '<',

level =60,

process = 'Y'))

),

type = "Nowcast"

)

pred\_res3 <- excursions\_normal3$pred.data[, c("id", "time", "observed")]

pred\_res3$normal\_excursions3 <- unlist(lapply(1:length(excursion\_ids),

function(i) excursions\_normal3$predictions$Y.summary[[i]]$excursions$P))

######### excursion for Y>200 (to get predictive probabilities for hyper-glycaemia) ############

excursions\_normal4 <-

predict(fit\_normal,

id = excursion\_ids,

controls = list(nSim = 1000,

nBurnin = 500,

silent = FALSE,

predict.derivatives = derivative\_list,

excursions = list(list(type = '>',

level =200,

process = 'Y'))

),

type = "Nowcast"

)

pred\_res4 <- excursions\_normal4$pred.data[, c("id", "time", "observed")]

pred\_res4$normal\_excursions4 <- unlist(lapply(1:length(excursion\_ids),

function(i) excursions\_normal4$predictions$Y.summary[[i]]$excursions$P))

################### getting prediction errors RMSE and MAE- see manuscript ###################

MAE<- c(excursions\_normal[6],excursions\_normal[8])

RMSE<- c(excursions\_normal[10],excursions\_normal[12])

#creating prediction and excursion plots

#plot for observed and predicted trajectory with confidence interval

par(mfrow = c(1, 2))

for(i in 1:length(excursion\_ids)){

plot(pred\_res[pred\_res$id == excursion\_ids[i], "time"],

pred\_res[pred\_res$id == excursion\_ids[i], "observed"],

xlab = "Hour", ylab = "Glucose",xaxt = "n",

pch = 19, cex = 0.6, ylim=c(min(pred\_res[pred\_res$id == excursion\_ids[i], "normal.lb"]),max(pred\_res[pred\_res$id == excursion\_ids[i], "normal.ub"])))

polygon(c(pred\_res[pred\_res$id == excursion\_ids[i], "time"],rev(pred\_res[pred\_res$id == excursion\_ids[i], "time"])),c(pred\_res[pred\_res$id == excursion\_ids[i], "normal.lb"],rev(pred\_res[pred\_res$id == excursion\_ids[i], "normal.ub"])), col = 'grey80', border = NA)

points(pred\_res[pred\_res$id == excursion\_ids[i], "time"], pred\_res[pred\_res$id == excursion\_ids[i], "observed"], pch = 19,lwd =1)

lines(pred\_res[pred\_res$id == excursion\_ids[i], "time"],

pred\_res[pred\_res$id == excursion\_ids[i], "normal.Ymean"], lty = 2,lwd=2,xaxt = "n")

lines(pred\_res[pred\_res$id == excursion\_ids[i], "time"], pred\_res[pred\_res$id == excursion\_ids[i], "normal.ub"], lty = 'dashed', col = 'red',lwd=2)

lines(pred\_res[pred\_res$id == excursion\_ids[i], "time"], pred\_res[pred\_res$id == excursion\_ids[i], "normal.lb"], lty = 'dashed', col = 'red',lwd=2)

legend("topleft",

legend = c("Predicted", "Observed","95% CI"),

lty = c(2, NA,2),col=c("black","black","red"),pch = c(NA,19,NA), lwd=2 ,cex=0.8,bty = "n" )

axis(1, at=c(0,4,8,12,16,20,24),cex=0.3, labels=(c("12am","4am","8am","12pm","4pm","8pm","12am")))

legend("top",

legend = c("Mean predictor MAE= 7.4 (0.89)", "Mean predictor RMSE=11.2 (6.67)"),

cex=0.7, bty = "n" )

}

#plot for excursions

for(i in 1:length(excursion\_ids)){

########## fitfh graph for W<60 vs hour ##################

plot(pred\_res3[pred\_res3$id == excursion\_ids[i], "time"],

pred\_res3[pred\_res3$id == excursion\_ids[i], "normal\_excursions3"],

xlab = "Hour", ylab = "Excursion Probability",

type = "l", lty = 1, ylim = c(0, 1.1),lwd=2,xaxt = "n")

#

rect(0,0.8,24,1,col = rgb(0.5,0.5,0.5,1/4),border = NA)

legend("top", legend =c("Hypoglycemia (<60 mg/dL)","Hyperglycemic (>200 mg/dL)"),horiz=TRUE,col=c("black","blue"), lty = c(1,2),cex=0.8,lwd=2,bty = "n")

lines(pred\_res4[pred\_res4$id == excursion\_ids[i], "time"],pred\_res4[pred\_res4$id == excursion\_ids[i], "normal\_excursions4"],lty = 'dashed', col = 'blue',lwd=2)

axis(1, at=c(0,4,8,12,16,20,24),cex=0.3, labels=(c("12am","4am","8am","12pm","4pm","8pm","12am")))

}