GluGo2.0

Assessing the Merits of the current GluGo Algorithm

New T1D Paradigm on the Horizon

Dexcom G6 Sensor

NIGHTSCOUT

#WeAreNotWaiting

GluGo's Algorithm

- Structural model (no machine learning involved)
- Jingxian Liu (17') and Jihong Jin (16')

$$UpwardSlope(/min) = \frac{\frac{Sens}{CarbRatio} * Carb}{4 * 60}$$

$$DownwardSlope(/min) = -\frac{Insulin * Sens - Basal * 4 * Sens}{4 * 60}$$

Initial Hypotheses, Questions, and Concerns

- **Hypotheses**
- Algorithm will be fairly accurate overall
- substantially less accurate at predicting high glucose

Initial Hypotheses, Questions, and Concerns

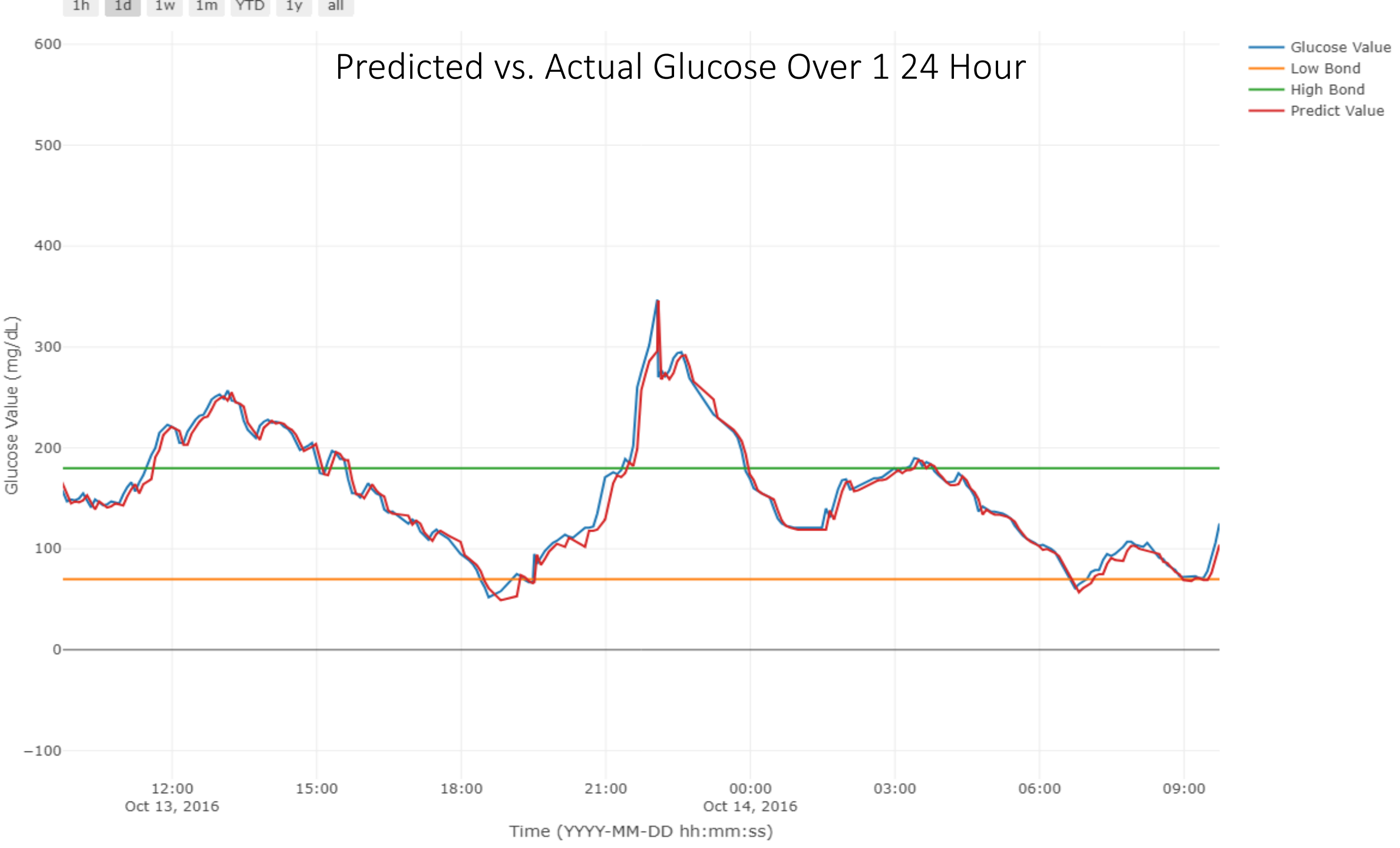
- **Hypotheses**
- Algorithm will be fairly accurate overall
- substantially less accurate at predicting high glucose

- **Questions**
- Are the predictions biased?
- Why is it less accurate for high glucose

Initial Hypotheses, Questions, and Concerns

- **Concerns**

- The algorithm is based solely on theory, and makes ***many*** simplifying assumptions
- Doesn’t “learn” about the patient (no ML) – not personalized


1h
1d
1w
1m
YTD
1y
all
Predicted vs. Actual Glucose Over 1 24 Hour
Glucose Value
Low Bond
High Bond
Predict Value
600
500
400
300
200
100
0
−100
Glucose Value (mg/dL)
12:00
Oct 13, 2016
15:00
18:00
21:00
00:00
Oct 14, 2016
03:00
06:00
09:00
Time (YYYY-MM-DD hh:mm:ss)

Summary Statistics on Residuals and Absolute Residuals

Residuals:

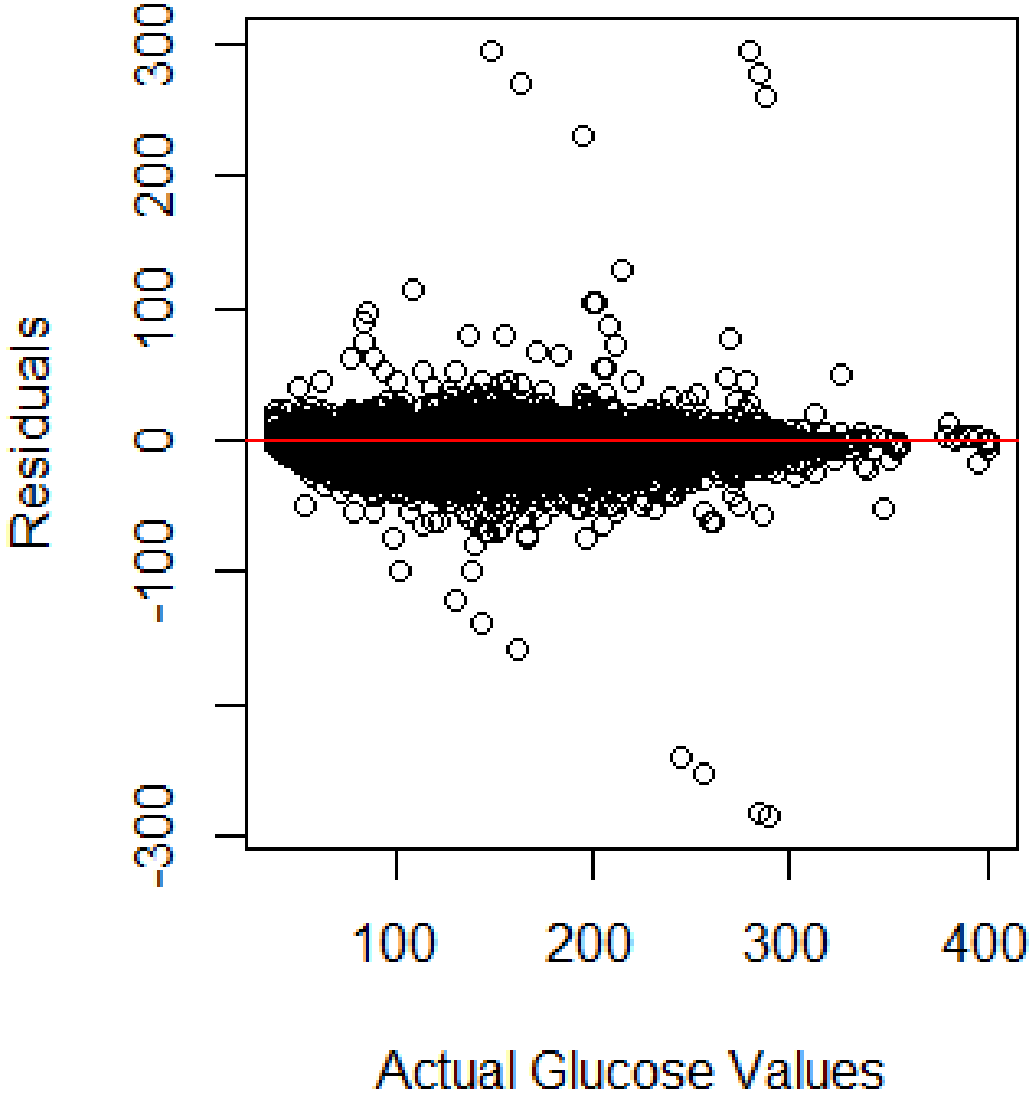
Mean	Median	Standard Deviation	Minimum	Maximum
-2.92	-2	9.11	-285	296

Sample Size
n = 28,078

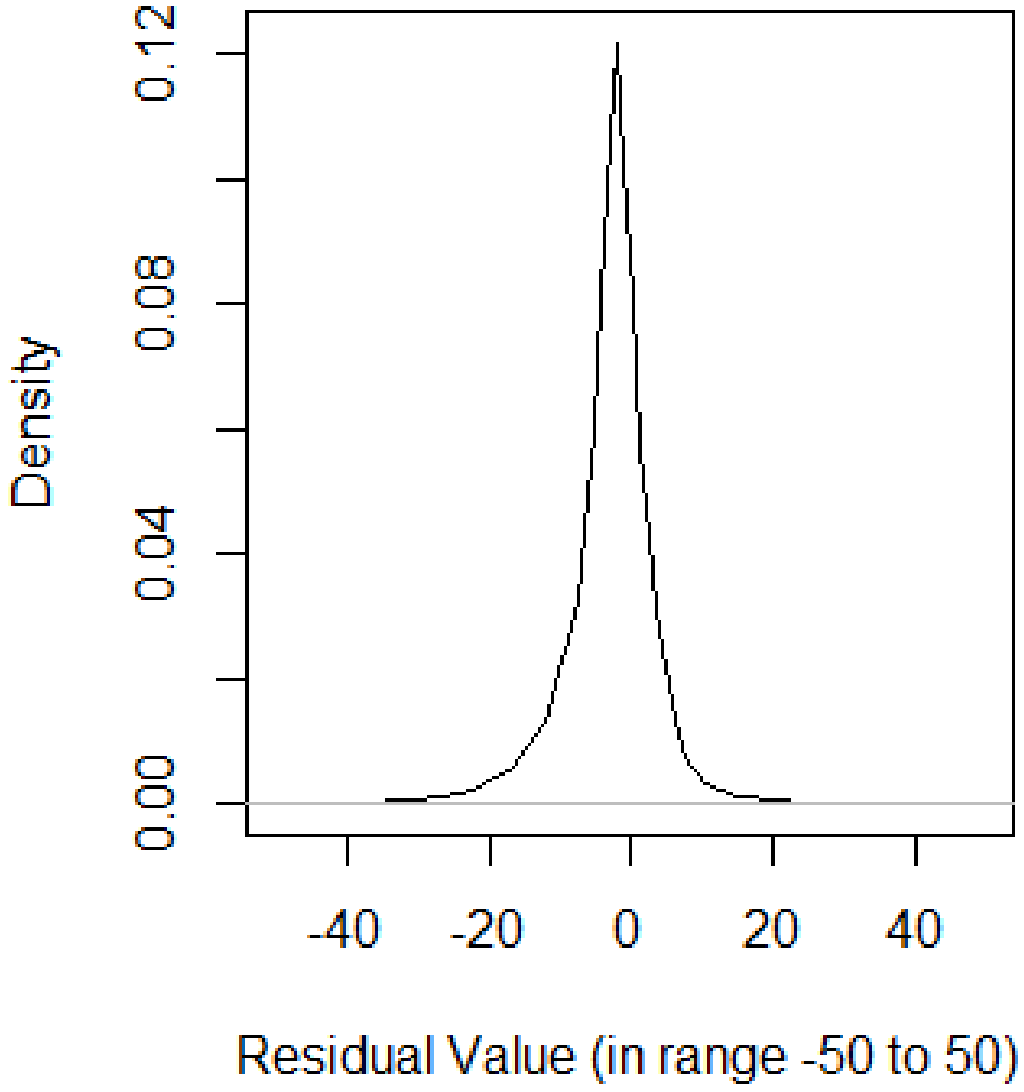
Absolute Residuals:

Mean	Median	Standard Deviation	Minimum	Maximum	Average % Residual
5.16	3	8.05	0	296	4.01%

Residuals (Predicted – Actual) by Actual Glucose Value



Residual Density Plot



Upon Closer Inspection…

- The algorithm predicts at most ***15 minutes*** into the future
- Usually only ***5 minutes***

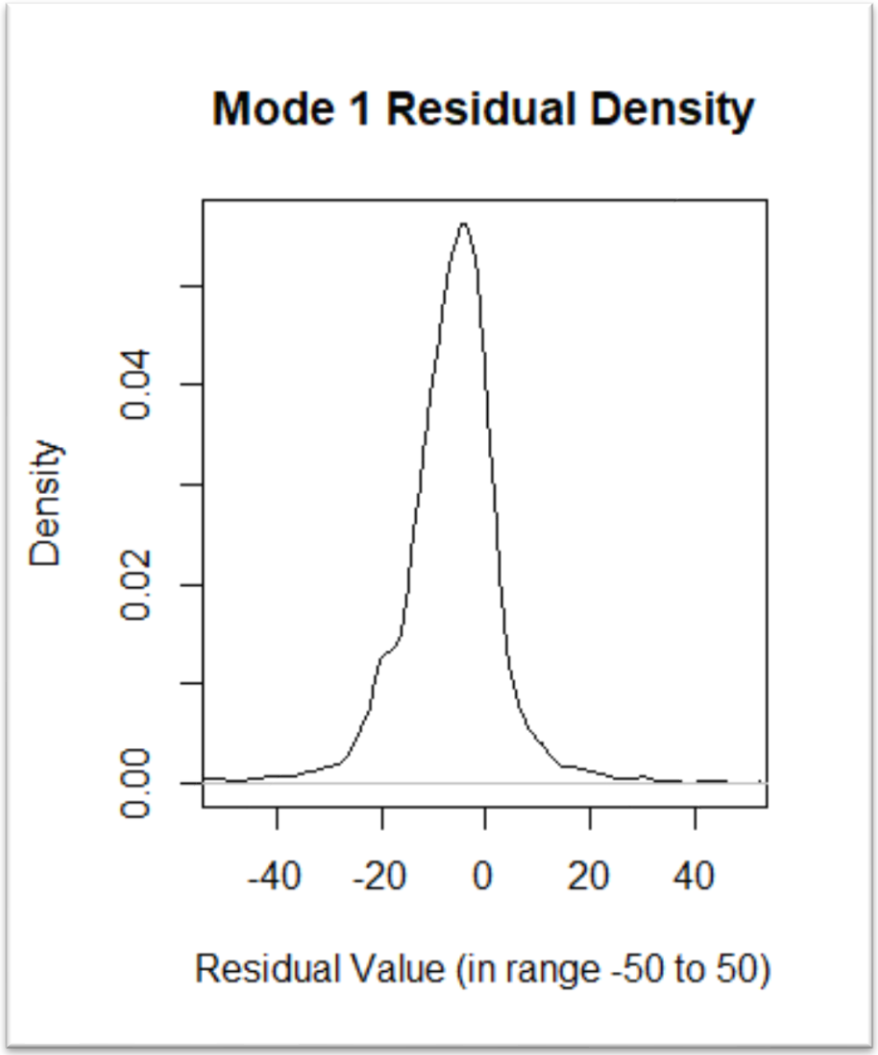
- The test data used by the previous group is ***not a random sample***
- Only a fraction of the total sample includes carb intake

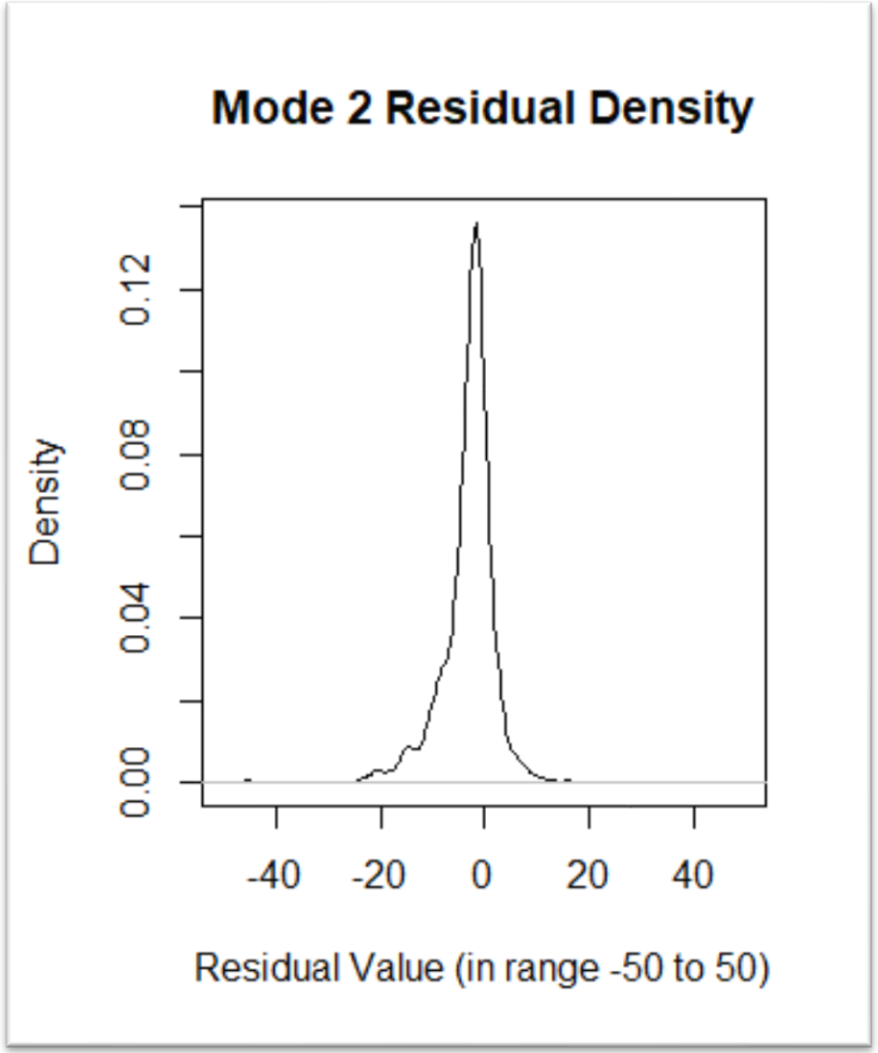
- Algorithm functions in two “modes”
- ***Mode 1***: After Carbohydrate Intake
- ***Mode 2***: No Carbohydrate Intake

Mode 1 (Carbs) vs. Mode 2 (No Carbs)

	Mode 1	Mode 2				
Mean Residual	-5.53	-3.25				
Mean Abs. Residual	9.74	4.23				
Average % Difference	7.37%	3.73%				
Sample Size	n = 3,016	n = 1,714				

Mode 1 (Carbs) vs. Mode 2 (No Carbs)





Conclusions about Algorithm

1. Old algorithm in it’s current form is not optimal

1. Prior Assessments of the model’s accuracy are inflated due to a biased sampling procedure

Looking Forward: Building New Test Datasets

- Compiling a new dataset

A	B	C	D	E	F	G
Timestamp	Glucose (ml/dL)	Basal Insulin (U/hr)	Insulin Sensitivity (mg/dL/U)	Carb Ratio (g/U)	Bolus Insulin (U)	Carb Intake (g)

Looking Forward: Building New Test Datasets

- Compiling a new dataset

A	B	C	D	E	F	G
Timestamp	Glucose (ml/dL)	Basal Insulin (U/hr)	Insulin Sensitivity (mg/dL/U)	Carb Ratio (g/U)	Bolus Insulin (U)	Carb Intake (g)

- Testing the new dataset

Moving from a database to directly reading a csv and using pandas library

```python
from dataAccess import dataAccess as DB
import datetime as DT
import csv

#path to filled out DB
dbpath = "lab.db"
#Connect to the Database by creating an instance of dataAccess

katDB = DB("Kate",dbpath)

for patient in [katDB]:
    dat = patient.getData()

gluPredict = []

i = 0
while i < len(dat):

    # if dat[i][3] == None:

    #   gluPredict.append()
    if dat[i][9] == "INRANGE":
        if dat[i][3] != None:

            bolis = dat[i][3]
            carb = dat[i][4]
            carbRate = dat[i][15]
            basalRate = dat[i][16]
            sens = dat[i][17]
            gluPredict.append(dat[i-1][2])

            currentTime = dat[i][1]          Evan Laferriere (a year ago) · Add files via upload

            upwardSlope = ((sens/carbRate) * carb) / (4*60.0)
            downwardSlope = (- bolis * sens - basalRate * 4 * sens) / (4*60.0)
            i  = i+1

            if dat[i][3] == None:
                upValue1 = round(dat[i-1][2] + ((dat[i][1] - dat[i-1][1]).total_seconds() / 60.0) * upwardSlope)
                upValue2 = round(upValue1 + ((dat[i+1][1] - dat[i][1]).total_seconds() / 60.0) * upwardSlope)
                upValue3 = round(upValue2 + ((dat[i+2][1] - dat[i+1][1]).total_seconds() / 60.0) * upwardSlope)
                i = i+3
                gluPredict.append(upValue1)
                gluPredict.append(upValue2)
                gluPredict.append(upValue3)


                while dat[i][3] == None and i < len(dat) -4 and dat[i+1][9] == "INRANGE" and (dat[i][1] - currentTime).total_seconds() < 15300 :
                    newValue = round(dat[i-1][2] + ((dat[i][1] - dat[i-1][1]).total_seconds() / 60.0) * downwardSlope)
                    gluPredict.append(newValue)
                    i = i+1
```

```python
import csv
import pandas
import numpy

data = pandas.read_csv('mlk.csv')
#for i in range (0,5):          Testing if pandas work.
#    print(data["Timestamp"][i])

length_csv = len(data)
#print(length_csv)
gluPredict = []

i = 0
while i < length_csv:
    #print(data["Bolus Insulin (U)"][i]) just to check the values.
    # if dat[i][3] == None: LEGACY CODE
    #   gluPredict.append() LEGACY CODE
    if data["Bolus Insulin (U)"][i] != None : # assuming that if Bolus insulin has a value then, it will be an
        bolis = data["Bolus Insulin (U)"][i]
        carb = data["Carb Intake (g)"][i]
        carbRate = data["Carb Ratio (g/U)"][i]
        basalRate = data["Basal Insulin (mg/dL)"][i]
        sens = data["Insulin Sensitivity (mg/dL/U)"][i]
        gluPredict.append(data["Glucose (ml/dL)"][i])

        print("glucose level: ")
        print(data["Glucose (ml/dL)"][i]) # print the glucose level
        currentTime = data["Timestamp"][i] # put it as a time stamp but its not really time stamp. Not sure what the value means in their database.

        upwardSlope = ((sens/carbRate) * carb) / (4*60.0)
        downwardSlope = (- bolis * sens - basalRate * 4 * sens) / (4*60.0)
        i  = i+1
        if data["Bolus Insulin (U)"][i].isnull():
                upValue1 = round(data["Glucose (ml/dL)"][i-1] + ((data["Timestamp"][i] - data["Timestamp"][i-1]).total_seconds() / 60.0) * upwardSlope)
                upValue2 = round(upValue1 + ((data["Timestamp"][i+1] - data["Timestamp"][i]).total_seconds() / 60.0) * upwardSlope)
                upValue3 = round(upValue2 + ((data["Timestamp"][i+2] - data["Timestamp"][i+1]).total_seconds() / 60.0) * upwardSlope)
                i = i+3
                gluPredict.append(upValue1)
                gluPredict.append(upValue2)
                gluPredict.append(upValue3)


                while data["Bolus Insulin (U)"][i] == None and i < length_csv -4 and (data["Timestamp"][i] - currentTime).total_seconds() < 15300 :
                    newValue = round (data["Glucose (ml/dL)"][i-1] + ((data["Timestamp"][i] - (data["Timestamp"][i-1]).total_seconds() / 60.0) * downwardSlope)
                    print("i am here")
                    gluPredict.append(newValue)
                    i = i+1


    else:

        newValue = round(data["Glucose (ml/dL)"][i-1] - ((data["Timestamp"][i] - data["Timestamp"][i-1]).total_seconds() / 60.0) * (data["Basal Insulin"][i] *
```

Looking Forward: Better Algorithms

- SVR Model with Physiological Features: