UNIVERSITY OF CALIFORNIA SAN DIEGO

Statistical Modeling of Sensors and Dials in Metabolic Networks

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science and Engineering

by

Swathi Vishwanath Hoysala

Committee in charge:

Professor Bernhard O. Palsson, Chair
Professor Nuno Bandeira
Professor Debashis Sahoo

2018

The thesis of Swathi Vishwanath Hoysala is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2018

DEDICATION

To my Mom, Dad, Grandparents and Sister.

I am what I am because of you!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

My advisor Dr. Bernhard O. Palsson for taking me under his wing, for seeing the potential in this project and for his valuable feedback.

My co-advisor and mentor Dr. Zachary King for being my neighbor who I happened to bump into in the hallway and decided to collaborate. For the countless hours he spent explaining the biological concepts, for the brainstorming sessions and most importantly for his research acumen.

My committee members Dr. Nuno Bandeira and Dr. Debashis Sahoo for their valuable and insightful feedback.

Researchers at System's Biology Research Group for all their help and support.

The Novo Nordisk foundation for funding my research.

Isaac Shamie for laying the initial groundwork for this project.

The CSE MS advisors for helping with signatures for thesis and c My friend Alok for all the caffeine and entertainment he supplied and my roommate Asmitha for feeding me

ABSTRACT OF THE THESIS

Statistical Modeling of Sensors and Dials in Metabolic Networks

by

Swathi Vishwanath Hoysala

Master of Science in Computer Science and Engineering

University of California San Diego, 2018

Professor Bernhard O. Palsson, Chair

Knowing how a microbe senses environmental inputs and regulates metabolic changes is important for metabolic engineers trying to direct microbial resources and reactions to specific pathways. Prediction of metabolic changes that result from genetic or environmental perturbations has several important applications, including diagnosing metabolic disorders and discovering novel drug targets. Most of the research in the field of modeling transcriptional regulatory networks (TRNs) and their metabolic effects focuses on integrating metabolic networks with additional data like transcriptional or genomic data. However, these existing methods are limited by the availability of datasets and the huge parameter space associated with TRN models. Thus, there is a need for alternative approaches to modeling regulation of metabolic networks.

It was recently established that microbial cells contain flux sensors which measure the rate at which enzymatic reactions take place, and then adjust, or dial, certain reactions and pathway fluxes. We hypothesize that these flux **sensors provide enough information to predict the change in metabolic "dials"**, i.e flux splits between different pathways. This project aims to prove the above-mentioned hypothesis using statistical modeling of sensors and dials data in metabolic network simulations.

Using Markov Chain Monte Carlo sampling methods, we sample the flux states of the *Escherichia coli* K-12 MG1655 strain under varying nutrient sources. We sample from 34 conditions to create a dataset with 340000 datapoints, each representing a unique feasible metabolic flux. We then apply statistical modeling techniques including linear regression, decision trees and ensemble learning methods to predict metabolic dial values using sensor values as input. The results from the statistical modeling techniques show that sensors can effectively predict the dial values without the need for additional data like transcriptional or genomic data.

Chapter 1

Introduction

To survive, microbial cells depend on precise regulation of metabolic operation. Cellular metabolism is comprised of:

- catabolic pathways which produce energy by breaking down molecules, and
- anabolic reaction routes, which synthesize the molecules required by the cell by providing the essential building blocks.

These catabolic pathways and anabolic reaction routes enable homeostasis and growth of the cells [GBR$^+$13].

A cell needs to control and adapt its enzyme production and metabolic activity depending on its requirements for growth. The regulation of microbial metabolic operation is tightly controlled by multiple layers [KZH10] as shown in figure 1.1.

Transcriptional, translational, allosteric, and post-translational regulations ensure that a microbe can respond to **diverse extracellular cues** through global and local circuits [CGKS14]. This is done via signal transduction mechanisms that are closely related to regulatory mechanisms through signalling cascades. The process of transcription and translation are clearly interconnected as illustrated in figure 1.2 but have usually been studied separately in distinct sub-disciplines of cellular biology.

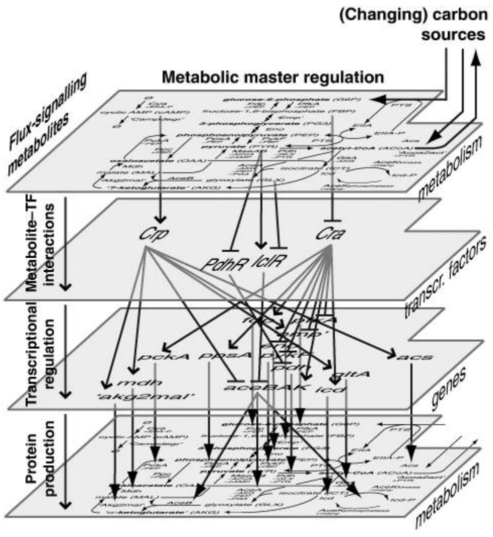

(Changing) carbon sources
Metabolic master regulation
Flux-signalling metabolites
metabolism
Metabolite–TF interactions
Crp
PdhR
IclR
Cra
transcr. factors
Transcriptional regulation
pckA
ppsA
mdh
'akg2mal'
aceBAK
icd
gltA
acs
genes
Protein production
metabolism

Figure 1.1: Different layers that control the regulation of microbial metabolic operations. From [KZH10]

Microbial cells consist of both external and internal sensors. Flux **sensors** measure the rate at which certain metabolic changes like enzymatic reactions take place. Knowing how a microbe senses input and dials a response is important for metabolic engineers trying to direct microbial resources and reactions to specific pathways. They typically aim to dial flux of certain pathways or specific reactions.

To understand the effect of flux sensors on the network, through literature search, we made a comprehensive list of discovered sensors and their potential effects on metabolic activities of microbes. Additionally, through literature search, current desires of the engineering community, as well as using the database RegulonDB [GCSSZ$^+$16] which consists of TFs and their targets, we made a comprehensive list of flux splits i.e **dial** values. Data from RegulonDB will give insight into what pathways the sensors affect.

The objective of this project is the prove the following hypothesis: **Sensors contain enough information to predict the values of the dials**.

From figure 1.2, the only components involved in this process are signals (both internal and external) and metabolites. Unlike other modeling approaches described in chapter 3, this model

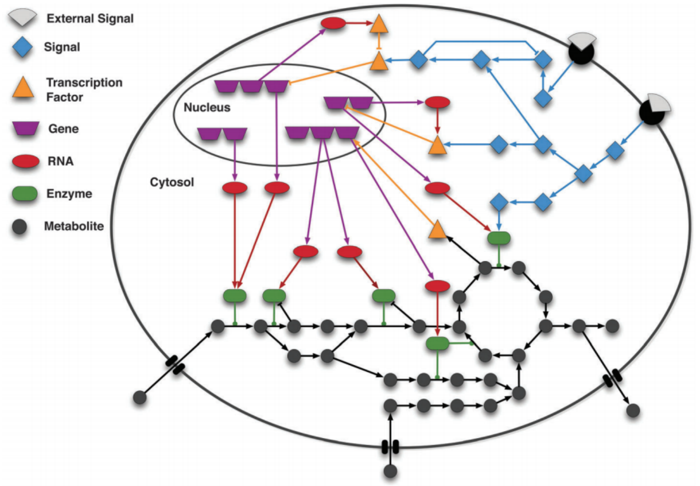


Figure 1.2: Schematic representation of the interconnection among signalling, gene regulation and metabolism from [GBR+13]

does not require additional data collected from literature or analysis of transcriptional or genomic data.

Table 1.1: List of BiGG models and their reference

BiGG Model	Reference
iJO1366	[OCN+11]
iML1515	[MLB+17]

Genome-scale metabolic models (GEMs) are mathematically-structured knowledge bases. GEMs contain descriptions of the biophysical constraints on metabolic systems like nutrient uptake, oxygen availability, reaction stoichiometry and reversibility [FHT+09]. They also contain descriptions of all the biochemical reactions, metabolites and genes in metabolism for a specific organism a Biochemical, Genetic and Genomic (BiGG) knowledge base [FHT+09], [KLD+16].

Using Markov Chain Monte Carlo (MCMC) sampling methods, we sample the flux states of the *E. coli* MG1655 strain under varying nutrient sources. We sample from 34 conditions across the BiGG Models shown in 1.1 to create a dataset with 340000 datapoints. We then apply statistical modeling techniques like linear regression, decision trees and ensemble learning methods to predict the dial values using sensor values as input. The results from the statistical modeling techniques show that sensors can effectively predict the dial values without the need for additional data like transcriptional or genomic data.

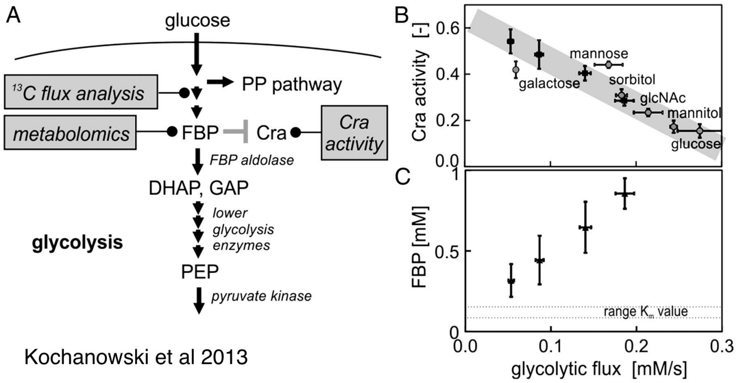
Representing microbial regulation as a state of sensor and dial values is a new paradigm that can aid in metabolic engineering. Based on the predictions of the learning model, follow-up experiments can be designed to engineer desired dial parameters (phenotypes) through optimal perturbations to the sensors or the regulatory network.

Chapter 2

Background

2.1 Sensors

Cells have built-in sensors that can react to certain cues, directly and indirectly. In addition to nutrient sensors, which measure the concentration of nutrients, metabolic flux sensors have recently been discovered [KVG$^+$13]. Flux sensors measure the rate of certain enzymatic reactions, and then adjust, or dial, certain reactions and pathway fluxes. One example as shown in figure 2.1 from [KVG$^+$13] paper is a glycolytic flux sensor represented by fructose -1,6-bisphosphate (FBP), which then represses the transcription factor (TF) Cra, affecting downstream processes.



Kochanowski et al 2013

Figure 2.1: Glycolytic flux sensor from [KVG$^+$13]

Table 2.1 provides a list of all the sensors and their respective BiGG [KLD+16] IDs. For the purposes of this project, only the first four sensors were chosen i.e only PGI, nadh_c, nadhp_c and EX_o2_e sensors are used to predict the dial values.

Table 2.1: List of sensors and their BiGG IDs

Sensor	BiGG ID	Literature reference
fructose-1,6-bisphosphatase (FBP)	PGI	[KVG+13]
Aerobic respiration control protein ArcA (ArcA), Fumarate and Nitrate reductase Regulatory (FNR)	nadh_c metabolite	[FKE+14]
Fumarate and Nitrate reductase Regulatory (FNR)	EX_o2_e	[FKE+14]
	nadph_c metabolite	
Cyclic adenosine monophosphate (cAMP)	ADNCYC	[CGKS14]
L-glutamine (gln__L_c)	EX_gln__L_e; GLNS; GLUDy	[CGKS14]

2.2 Dials

A dial is defined as a change in flux splits between two different pathways. Figure 2.2 from [CGKS14] shows a pictorial representation of a dial. One example is dialing flux from glycolysis to pentose-phosphate pathways as shown in figure 2.3

Table 2.2 lists all the dials and the respective BiGG IDs. Only the first three dials were chosen i.e only Pentose phosphate vs glycolysis, Fermentation - Pyruvate and Fermentation - Acetyl-CoA dials.

Table 2.2: List of dials and their BiGG IDs

Dial	BiGG ID
Pentose phosphate vs glycolysis	PGI / G6PDH2r
Fermentation – Pyruvate	LDH_D / PDH, PFL / PDH
Fermentation – Acetyl-CoA	PTAr / ACALD, PTAr / CS
Glyoxylate shunt	ICL - ICDHyr
Embden-Meyerhoff-Parnass (EMP) vs Entner-Doudoroff (ED)	GND / EDD
Pep Split	(GLCptspp + PYK) / PPC
NH4 uptake options	GLUDy / GLUSy, GLUDy / GLNS

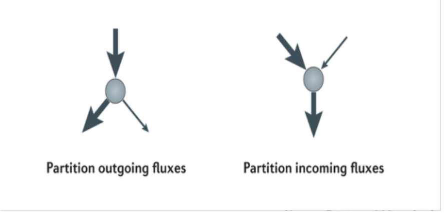


Figure 2.2: Pictographic representation of Dial from [CGKS14]

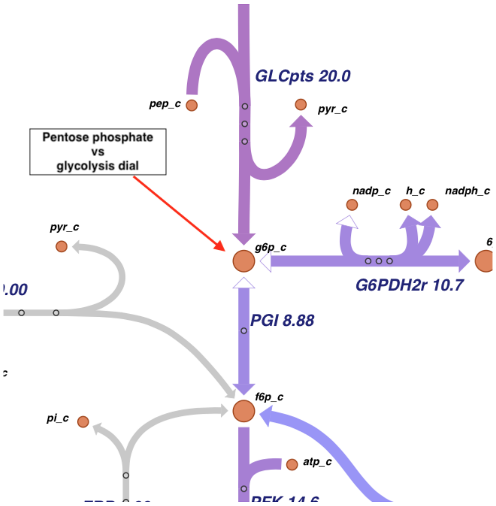


Figure 2.3: Pentose-phosphate vs glycolysis dial

Chapter 3

Related Work

While a lot of work in genome-scale modeling has been devoted to the study of metabolic networks and TRNs separately, a seamlessly integrated metabolic-regulatory network would enable better predictions of how genetic mutations and transcriptional perturbations are translated into flux responses at the metabolic level. There have been significant successes in this endeavor [CKR^{+}04], [HLPP06], yet substantial challenges remain. Among the integrated models, the most commonly used genome-scale analysis method today is regulatory flux balance analysis (RFBA) [CKR^{+}04], [CSP01]. This method links the transcriptome of an organism with metabolism and incorporates regulatory constraints into flux balance analysis (FBA). Figure 3.1 provides the system design and overview of RFBA.

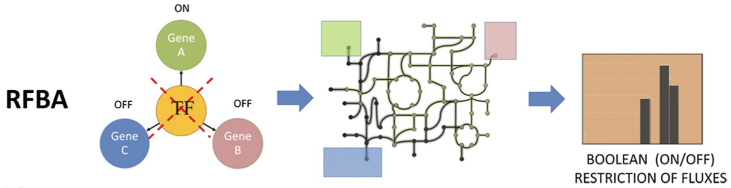


Figure 3.1: Regulatory Flux Balance Analysis method of genome-scale analysis [CP10]

In the case of RFBA, the metabolic network is not only restricted by mass, thermodynamic, and energy constraints, but also by the gene regulatory network that controls it.

Steady-state RFBA (SRFBA) [SESR07] and integrated FBA (iFBA) [CXCK08] are similar methods based on Boolean logic. SR-FBA uses the same genome-scale integrated metabolic regulatory network as RFBA but characterizes its steady-state behavior, whereas iFBA uses differential equations to model a subset of the regulatory network.

Since RFBA model simplifies the relationship between the transcriptome and the metabolome to a binary process, it has several shortcomings associated with it and other boolean logic-based methods. The absence of an automated algorithm to determine the boolean rules for relating the regulator with its target is the biggest shortcoming of RFBA when a large number of species are considered. Although the manual process can be accurate in modeling metabolic regulation, manual reconstruction greatly limits the number of interactions. Finally, this process also requires extensive literature search.

In order to overcome the drawbacks associated with RFBA, probabilistic regulation of metabolism (PROM) model was introduced in [CP10]. PROM, by automatically quantifying the interactions from high-throughput data, enables direct integration of the transcriptional and metabolic networks for modeling. It overcomes the need for manually writing the Boolean rules which was the major drawback of RFBA. By virtue of automatically quantifying the interactions, PROM greatly increases the capacity to generate genome-scale integrated models. Figures 3.2 and 3.3 provide the system design and overview of PROM.

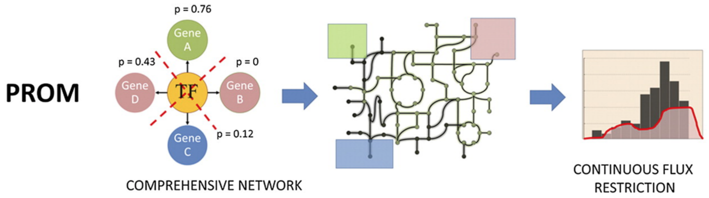
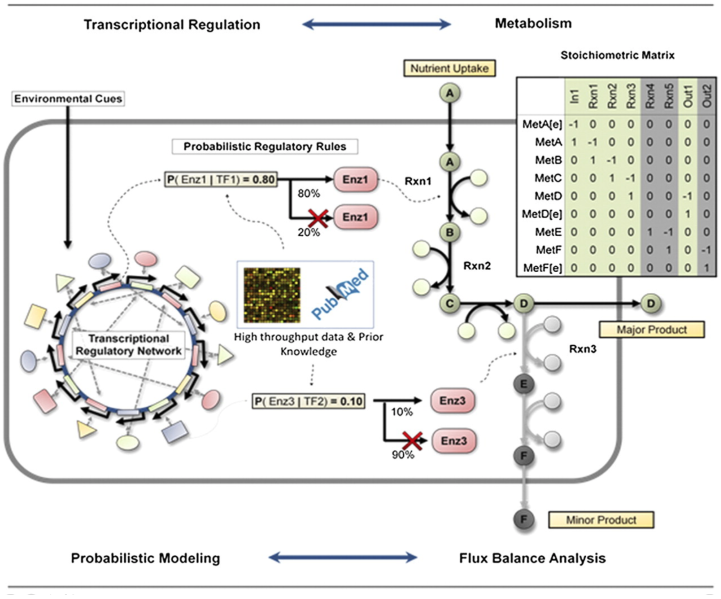


Figure 3.2: Probabilistic Regulation of Metabolism method of genome-scale analysis from [CP10]

PROM is robust to noise in high-throughput data and can be easily integrated with automated algorithms for network inference. The PROM algorithm uses conditional probabilities for modeling transcriptional regulation, similar to the probabilistic Boolean networks of [SDKZ02] and uses FBA [KPE03] for modeling metabolic networks. PROM introduces probabilities to represent gene states and genetranscription factor interactions.



	In1	Rxn1	Rxn2	Rxn3	Rxn4	Rxn5	Out1	Out2
MetA[e]	-1	0	0	0	0	0	0	0
MetA	1	-1	0	0	0	0	0	0
MetB	0	1	-1	0	0	0	0	0
MetC	0	0	1	-1	0	0	0	0
MetD	0	0	0	1	0	0	-1	0
MetD[e]	0	0	0	0	0	0	1	0
MetE	0	0	0	0	1	-1	0	0
MetF	0	0	0	0	0	1	0	-1
MetF[e]	0	0	0	0	0	0	0	1

Figure 3.3: Overview of the process used to integrate the metabolic and regulatory network using PROM from [CP10]

A different approach, one that is not associated with either probabilistic or Boolean networks, but is dependent on signal networks is the Genetic sensory response units (GENSOR units) [LTICV17]. GENSOR formalizes the process of detection and processing of environmental information mediated by individual transcription factors (TFs). These units are composed of four components namely a signal, signal transduction, genetic switch, and a response. In order to assemble a GENSOR unit, experimentally validated data sets from two databases were used for each of the 189 local TFs of *E. coli* K-12 contained in the RegulonDB database. Ideally, GENSOR units describe the information that flows through different layers of cellular organization to produce an appropriate response. Further analysis of the GENSOR unit set showed that less than a quarter of the TFs regulate genes that belong to the same metabolic flux, but feedback is a common occurrence. A gradient of response complexity can be observed and is partially explained by the regulatory effect of the corresponding TF.

Beyond the biological insights of GENSORs, [LTICV17] provides the set of GENSOR units as a standardized framework for small- and large-scale analyses of the interplay between transcriptional regulation and metabolism.

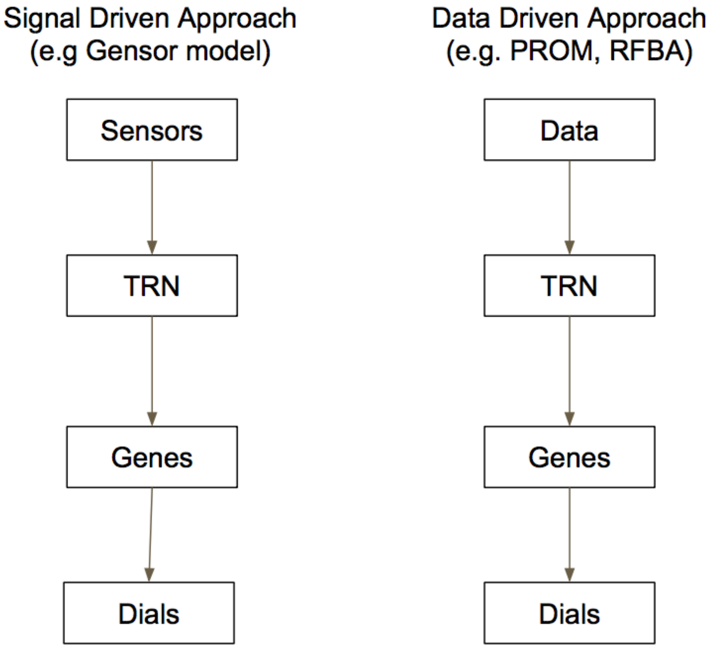


Figure 3.4: Block diagram of different modeling approaches

From figure 3.4, we can see the both RFBA and PROM methods are dependent on data from literature which is hard to curate. Although GENSOR method doesn't require extensive data collection, all three methods require the integeration of metabolic and transcriptional networks. This additional step is not necessary in the **sensors and dials approach**. The approach presented in this paper requires very little data gathering from literature. The sensors and dials approach uses statistical modeling to mimic the workings of the TRN and genetic layer in figure 3.4.

Other methods based on stochastic models or differential equations [MLGEP08] and [WDH$^+$17] are usually restricted to modeling small systems and have not been extended thus far to the genome scale.

Chapter 4

System Design

The block diagram of the system is as show in figure 4.1

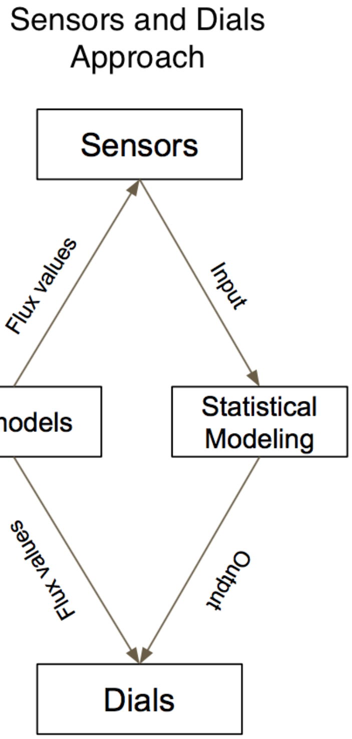


Figure 4.1: Block diagram of the sensors and dials modeling approach

If the values of the sensors are known, using statistical modeling, the values of the dials can be predicted.

This approach is different from both signal driven and data driven approaches shown in figure 3.4. It does not require information about the underlying transcription factors or the gene data. One can directly predict the dial values using the sensor values through statistical modeling approaches.

The following chapters will detail the dataset used in the modeling task. A number of models with varying accuracy are used to predict the dial values. The workings and the results of each of the models are detailed in chapters 6 and 7

Chapter 5

Dataset

The dataset consists of metabolic flux values of *E. coli* str. K-12 substr. MG1655 [BPB^{+}97] obtained by MCMC sampling of BiGG Models as seen in table 1.1.

Table 5.1: List of Sampling conditions used in MCMC sampling

Sources	Lower bound values
EX_glc__D_e	0, 5 and 10
EX_xyl__D_e	0, 5 and 10
EX_rib__D_e	0, 5 and 10
EX_glyc_e	0, 5 and 10
EX_ac_e	0, 5 and 10
EX_pyr_e	0, 5 and 10
EX_nh4_e	0 and 1000
EX_o2_e	0, 2 and 20
EX_arg__L_e	0, 5 and 10
EX_asp__L_e	0, 5 and 10
EX_ser__L_e	0, 5 and 10
EX_cys__L_e	0, 5 and 10
EX_pro__L_e	0, 5 and 10
EX_ala__L_e	0, 5 and 10
EX_trp__L_e	0, 5 and 10

5.1 Sampling Conditions

The dataset was created by constraining the MCMC sampling of BiGG Models by a set of conditions. The conditions were created by varying the magnitude and direction (positive or negative) of the lower bound values of different Carbon, Nitrogen, Oxygen and Amino acid sources. The complete list of sources is as shown in table 5.1.

5.2 MCMC Sampling Procedure

Since the dataset consists of metabolic flux values, the most widely used technique to analyze these fluxes in large-scale metabolic reconstructions is FBA. In FBA, a liner objective function , typically the biomass or some biological proxy of it is introduced, and the problem reduces to finding the subspace of the polytope, which optimizes the objective function. If this subspace consists in only one point, the problem can be efficiently solved using linear programming. However, if one is interested in describing more general growth conditions, or is interested in other fluxes than the biomass, different computational strategies must be envisaged.

As long as no prior knowledge is considered, each point of the polytope is an equally viable metabolic phenotype of the biological system under investigation. Therefore, being able to sample high-dimensional polytopes becomes a theoretical problem with concrete practical applications. From a theoretical standpoint, the problem is known to be NP-hard and thus an approximate solution to the problem must be sought. The approximate solution can be obtained using MCMC sampling technique. MCMC sampling, basically, consists of iteratively collecting samples by choosing random directions from a starting point belonging to the polytope.

For this project, the COBRApy software was used to perform MCMC sampling. A set of 34 conditions were created using all combinations of sources and lower bound values mentioned in table 5.1. An example of the sampling condition set is as shown in listing 5.1. The feasibility of each of the samples was checked by running FBA. If the solution objective of FBA (in this case the growth rate) is greater than 0.05, the optimized model was used for sampling. Before MCMC sampling was run, the lower bound value of the biomass function was set to 99% of the solution objective obtained after performing FBA. For each feasible condition, a sample of 1000 was collected. The dataset consisted of 340000 datapoints.

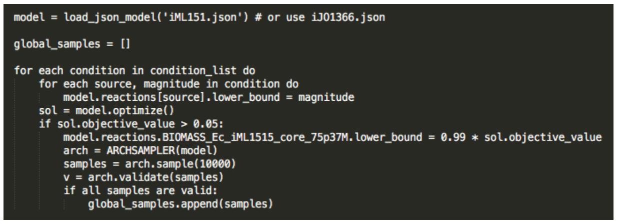
Listing 5.1: Sampling conditions

```
[
{
    "EX_nh4_e": -1000,
    "EX_o2_e": 0,
    "EX_glc__D_e": -10
},
{
    "EX_xyl__D_e": -10,
    "EX_nh4_e": -1000,
    "EX_o2_e": 0
},
{
    "EX_arg__L_e": -10,
    "EX_nh4_e": -1000,
    "EX_o2_e": 0,
    "EX_glc__D_e": -10
},
{
    "EX_ser__L_e": -10,
    "EX_nh4_e": -1000,
    "EX_o2_e": -20,
    "EX_glc__D_e": -10
}

]
```

The COBRApy software contains implementation of two different types of sampling techniques namely optGpSampler [MHM14] and ACHR Sampler [TPVP05]. From literature and from extensive experimentation as shown in the COBRAPY SAMPLING BENCHMARKING website, it was discovered that ACHR sampler performed better than optGpSampler. For the purposes of this project, ACHR sampler was used to perform sampling.

The pseudocode of the sampling technique used is as shown in figure 5.1



```
model = load_json_model('iML151.json') # or use iJO1366.json

global_samples = []

for each condition in condition_list do
    for each source, magnitude in condition do
        model.reactions[source].lower_bound = magnitude
    sol = model.optimize()
    if sol.objective_value > 0.05:
        model.reactions.BIOMASS_Ec_iML1515_core_75p37M.lower_bound = 0.99 * sol.objective_value
        arch = ARCHSAMPLER(model)
        samples = arch.sample(10000)
        v = arch.validate(samples)
        if all samples are valid:
            global_samples.append(samples)
```

Figure 5.1: Pseudocode of the sampling technique

5.3 Dataset Validation

In order to validate the samples obtained using MCMC sampling, a number of experiments were conducted and they are enumerated below.

5.3.1 Validate function

The most readily available validation technique is the ”validate” function that’s part of the ARCH sampler library. The validate function checks to see if the set of points is feasible and give detailed information about feasibility violations.

5.3.2 Auto-correlation Plots

The second technique used to validate the samples is by drawing an autocorrelation plot. From the fig 5.2 we can see that the autocorrelation tends to be close to 0 towards the end of the series. The sampling experiment was run repeatedly to select the best dataset with the lowest autocorrelation value.

5.3.3 Drawing Pathway Maps in Escher

The third technique used to validate the samples is by drawing pathway maps in Escher [KDE$^+$15]. This will rule out the obvious errors in the sampling data like loops that are out of the ordinary.

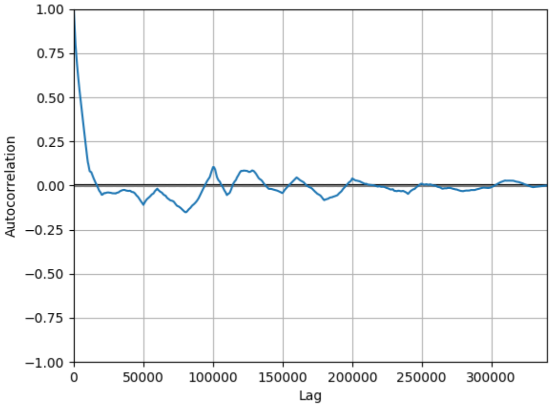


Figure 5.2: Autocorrelation plot for the 340k flux data

5.3.4 Gelman and Rubin Test

The fourth technique used to validate the samples is using the Gelman and Rubin test as explained in [MHM14] under the Empirical Convergence Diagnostics section. The R value obtained was 1.032. This indicated that the dataset has achieved empirical convergence.

Chapter 6

Methodology

In order to test the hypothesis about correlation between sensors and dials, an initial set of line plots of each sensor and dial were drawn. Once a correlation was indeed noticed, statistical modeling was done on the data. A number of statistical modeling techniques like linear regression, tree based models and ensemble learning models were used.

6.1 Initial Analysis

A line plot consisting of both sensor and dial values was plotted. Since the dataset consisted of 340000 datapoints, binned scatter plots of sensor vs dial were drawn.
A high correlation was noticed in case of PGI sensor and PGI-G6PDH2r dial which can be seen in fig 6.1. Similar correlation can be seen in figures 6.2, 6.3 and 6.4

6.2 Inputs and Outputs

The inputs and outputs are flux values of the corresponding sensors and dials.
Since the flux values vary from -1000 to $+1000$, a number of normalization techniques namely atan, log and **min-max** were used. After careful redrawing of line plots mentioned in section 6.1,

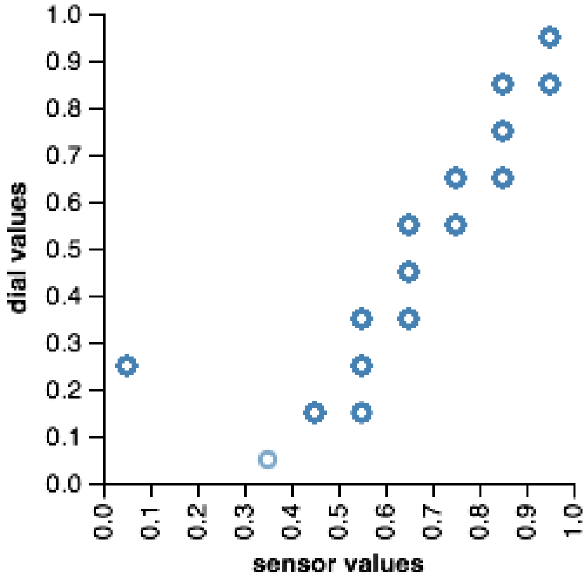


Figure 6.1: A binned scatter plot of PGI sensor vs PGI-G6PDH2r dial.

it was established that min-max normalization best suited the data.

The dials are a pair of metabolic reactions. Different methods namely ratio of the dial and **difference between the absolute values of the dials**, were used to represent the dials. The absolute value of the dials were taken into consideration to prevent penalization while calculating the difference. After careful redrawing of line plots mentioned in section 6.1, it was established that difference between the absolute values of the dials best suited the data.

The sensor values for nadh_c and nadph_c where calculated by taking the sum of the flux values for all the reactions associated with these metabolites.

The dataset was split into training-validation-testing sets. The split was 70%-10%-20% respectively. The dataset was shuffled to make sure that all the conditions were seen in all the sets.

A separate model was built for each dial.

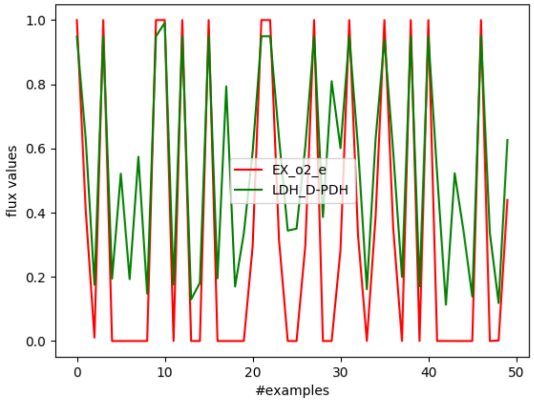


Figure 6.2: A line plot of Oxygen exchange reaction and (LDH_D, PDH) dial.

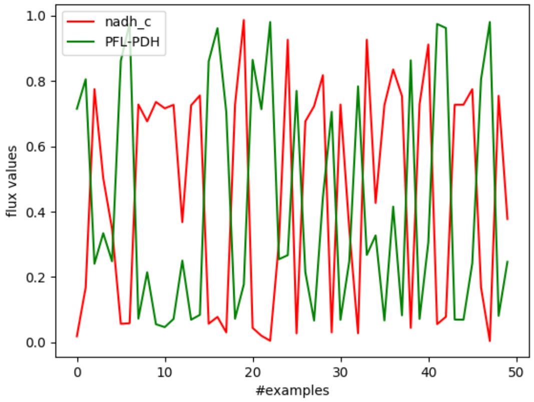


Figure 6.3: A line plot of nadh_c and $(PFL-PDH)$ dial.

6.3 Statistical Modeling

Statistical modeling was done using Scikit-learn [PVG+11], [BLB+13] library.

6.3.1 Linear Regression

From the line plots in figures 6.1, 6.2, 6.3 and 6.4, we can see that there exists a liner correlation between each of the sensors and dials. Since a linear correlation exists, the statistical model best suited was linear regression. The results for linear regression are as shown in section 7.1

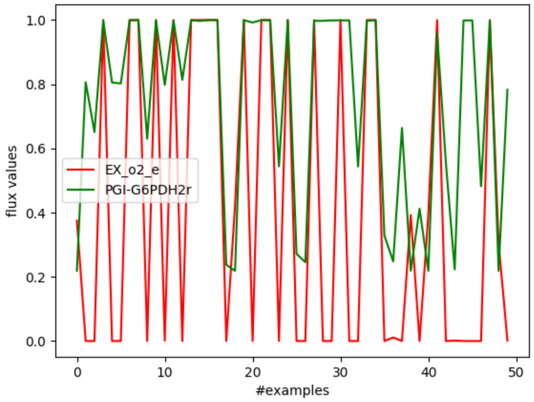


Figure 6.4: A line plot of EX_o2 _c PGI-G6PDH2r dial.

6.3.2 Extreme Gradient Boosted Trees

From section 7.1, we can see that linear regression doesn't perform as well as one would expect. Although there exists a linear relationship between each of the sensors and the dials, a combination of all the sensors leads to non-linearity. Since linear regression tries to fit a straight line for the dataset at hand, it fails to capture this non-linearity. In order to learn this non-linearity, ensemble learning techniques were used.

Extreme Gradient Boosted Trees (XGBoost) [CG16] is a class of gradient boosted tree algorithms that employ the tree ensemble models. The XGBoost library has in-built APIs to retrieve feature importance. Feature importance provides a score that indicates how useful or valuable each feature is in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions with decision trees, the higher its relative importance.

Feature importance is calculated explicitly for each attribute in the dataset, allowing attributes to be ranked and compared to each other. Importance is calculated for a single decision tree by the amount that each attribute split point improves the performance measure, weighted by the number of observations the node is responsible for. The performance measure may be the purity (Gini index) used to select the split points or another more specific error function. The feature importances are then averaged across all of the the decision trees within the model.

Feature importance is calculated as follows -

```
feature_imp = [0] * n_features
#traverse tree
for each internal_node that splits on feature i
    err = compute(error reduction of that node)
    feature_imp[i] = feature_imp[i] + err * len(samples through internal_node)
```

The results from XGBoost and the feature importance plots are as shown in section 7.2

6.3.3 Decision Tree Regressor

From section 7.2, we can see that although XGBoost performs better than linear regression, it is still not up to the mark. This is clearly visible in figure 7.10. We see that a large number of points deviate from the 45° line i.e from the actual value. This is because XGBoost, which employs boosting technique, is based on weak learners (high bias, low variance).

In terms of decision trees, weak learners are shallow trees, sometimes even as small as decision stumps (trees with two leaves). Boosting reduces error mainly by reducing bias and also to some extent variance, by aggregating the output from many models. The idea is to add a classifier/regressor at a time, so that the next classifier/regressor is trained to improve the already trained ensemble.

With respect to this dataset, XGBoost overfits to the training data and hence results in lower accuracy. On the other hand, decision tree by means of not employing weak learners, does not overfit the data and hence leads to more accurate result. XGBoost also requires a lot of parameter tuning and either the lack of tuning or extensive tuning can lead to overfitting.

The results from decision tree regressor and feature importance plots are as shown in section 7.3. The feature importance calculation is done similarly as in section 6.3.2.

6.3.4 Random Forest Regressor

From section 7.3, we can see that random forest regressor performs marginally better than decision trees.

Random Forest creates a large number of decision trees based on bagging. The basic idea is to resample the data over and over and for each sample train a new classifier/regressor. Different classifiers/regressors overfit the data in a different way, and through voting those differences are averaged out.

The results from random forest regressor and feature importance plots are as shown in section 7.4. The feature importance calculation is done similarly as in section 6.3.2.

Chapter 7

Results

The results of statistical modeling will be reported in terms of mean-squared-error (MSE), Pearson-r and Coefficient $R\hat{}2$ values. The results will be accompanied by scatter plots of actual vs predicted values.

7.1 Linear Regression

Table 7.1 reflects the results from linear regression.

From figures 7.1, 7.2, 7.3, 7.4 and 7.5, we can see that even though there exists a linear correlation between each sensor and dials, when all the sensors are considered together, the model performs poorly.

Table 7.1: Results from Linear Regression

Dial	MSE	Pearson-r	Coefficient Rˆ2
(PGI, G6PDH2r)	0.0100	0.9554	0.9128
(LDH_D, PDH)	0.0143	0.9297	0.8639
(PFL, PDH)	0.0113	0.9491	0.8639
(PTAr, ACALD)	0.0246	0.8339	0.6932
(PTAr, CS)	0.0290	0.8343	0.6921

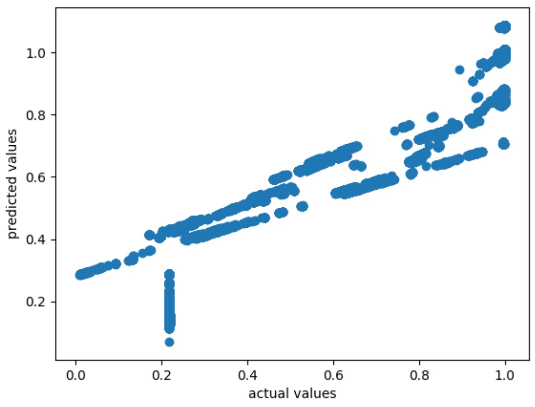


Figure 7.1: Linear Regression scatter plot of actual vs predict values for $(PGI, G6PDH2r)$ dial

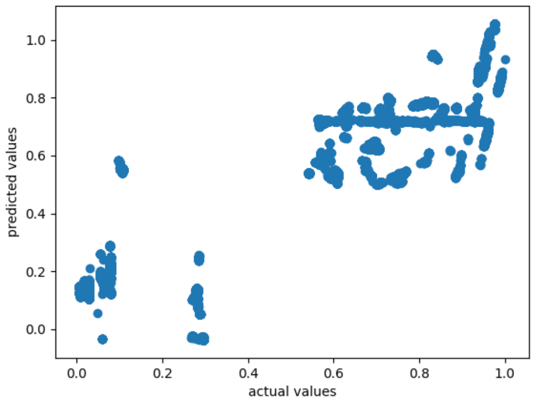


Figure 7.2: Linear Regression scatter plot of actual vs predict values for $(PTAr, CS)$ dial

7.2 Extreme Gradient Boosted Trees

Table 7.2 reflects the results from XGBoost regressor. From figures 7.6, 7.7, 7.8, 7.9 and 7.10, we can see that the xgboost regressor performs better than linear regression. Figures 7.11, 7.12, 7.13, 7.14 and 7.15 shows the feature importance plots for each of the dials.

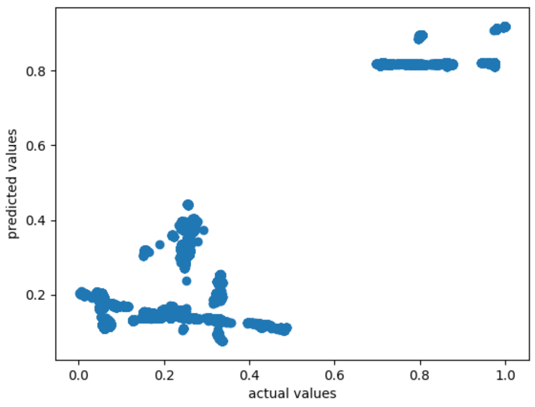


Figure 7.3: Linear Regression scatter plot of actual vs predict values for (PFL, PDH) dial

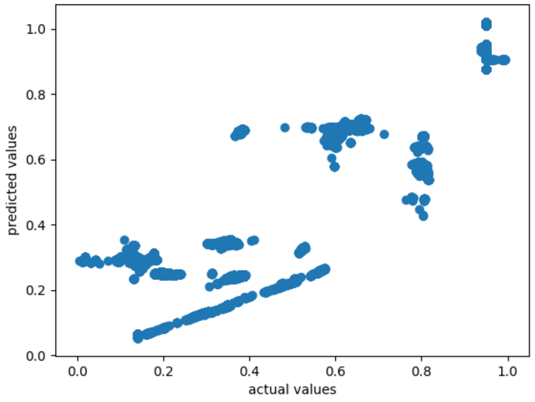


Figure 7.4: Linear Regression scatter plot of actual vs predict values for (LDH_D, PDH) dial

Table 7.2: Results from XGBoost Regressor

Dial	MSE	Pearson-r	Coefficient R^2
(PGI, G6PDH2r)	0.000247	0.99929	0.997847
(LDH_D, PDH)	0.000609	0.997753	0.994214
(PFL, PDH)	0.000744	0.997318	0.993482
(PTAr, ACALD)	0.001241	0.994060	0.9845706
(PTAr, CS)	0.001215	0.995115	0.9873441

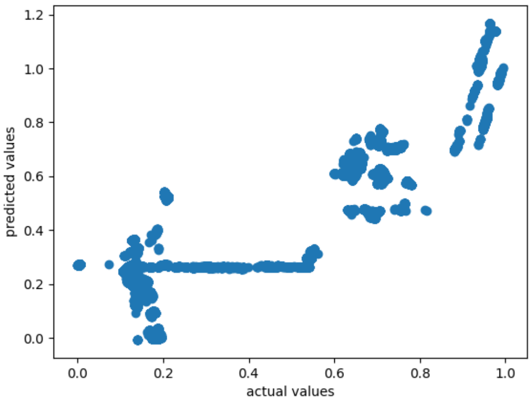


Figure 7.5: Linear Regression scatter plot of actual vs predict values for $(PTAr, ACALD)$ dial

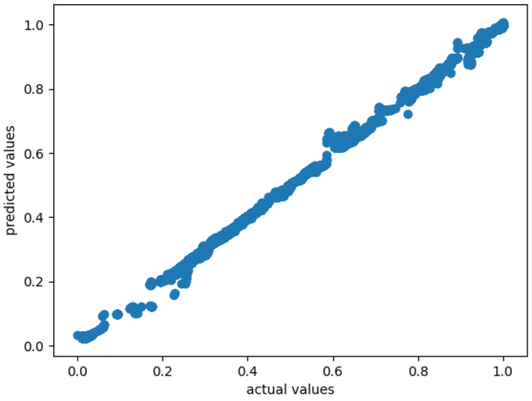


Figure 7.6: XGBoost Regressor scatter plot of actual vs predict values for $(PGI, G6PDH2r)$ dial

7.3 Decision Tree Regressor

Table 7.3 reflects the results from decision tree regressor.

From figures 7.16, 7.17, 7.18, 7.19 and 7.20, we can see that the decision tree regressor performs better than the xgboost regressor. It's the closest to the actual value with precision up to 5 decimal places.

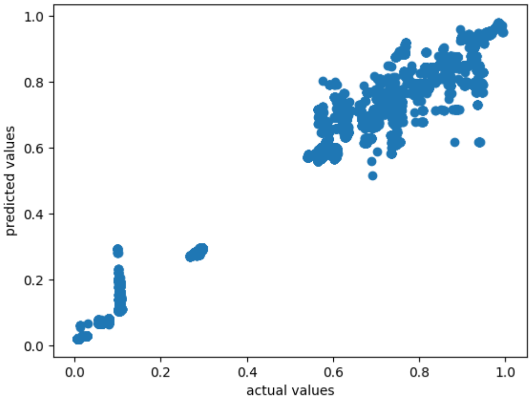


Figure 7.7: XGBoost Regressor scatter plot of actual vs predict values for $(PTAr, CS)$ dial

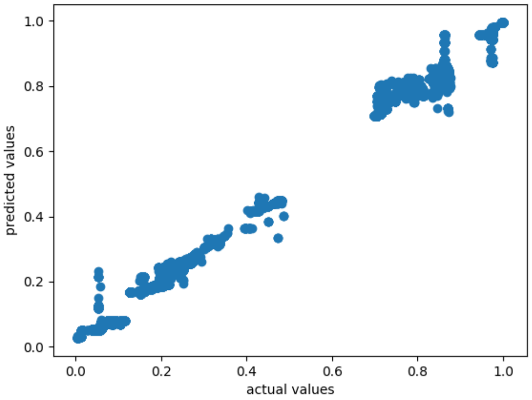


Figure 7.8: XGBoost Regressor scatter plot of actual vs predict values for (PFL, PDH) dial

Table 7.3: Results from Decision Tree Regressor

Dial	MSE	Pearson-r	Coefficient Rˆ2
(PGI, G6PDH2r)	2.0987e-06	0.99999	0.99998
(LDH_D, PDH)	4.9845e-06	0.99997	0.99995
(PFL, PDH)	1.8038e-06	0.99999	0.99998
(PTAr, ACALD)	1.75e-05	0.99989	0.99978
(PTAr, CS)	1.3527e-05	0.99992	0.99985

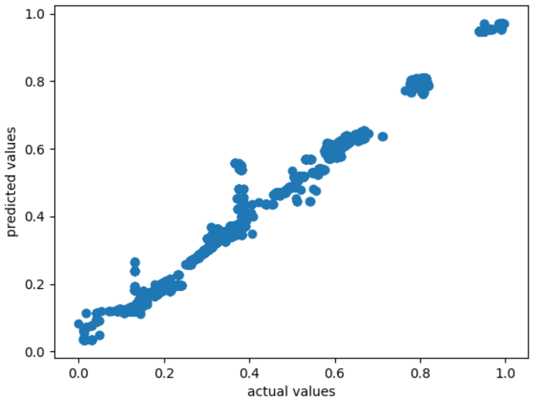


Figure 7.9: XGBoost Regressor scatter plot of actual vs predict values for (LDH_D, PDH) dial

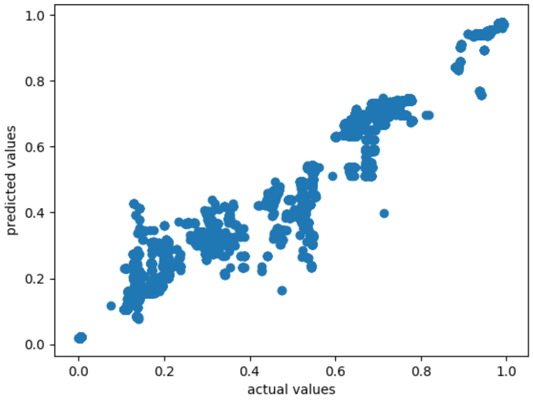


Figure 7.10: XGBoost Regressor scatter plot of actual vs predict values for $(PTAr, ACALD)$ dial

7.4 Random Forest Regressor

Table 7.4 reflects the results from random forest regressor. From figures 7.26, 7.27, 7.28, 7.29 and 7.30, we can see that the random forest regressor performs marginally better than decision tree regressor. It's the closest to the actual value with precision up to 6 decimal places. Figures 7.31, 7.32, 7.33, 7.34 and 7.35 shows the feature importance plots for each of the dials. For (PGI, G6PDH2r) dial, from the figure 7.31, we can see that PGI sensor is the dominating feature. Since statistical models depend largely on the dataset used to train them, it is obvious for

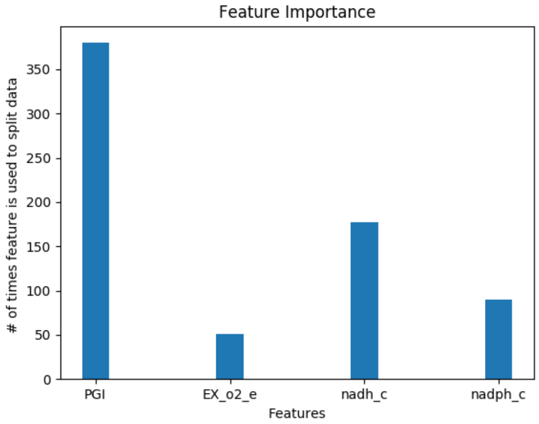


Figure 7.11: XGBoost feature importance plot for $(PGI, G6PDH2r)$ dial

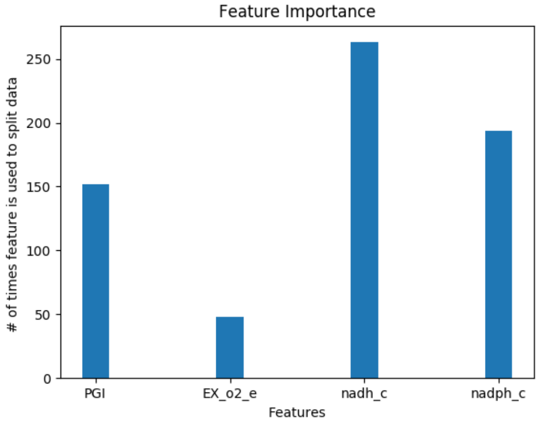


Figure 7.12: XGBoost feature importance plot for $(PTAr, CS)$ dial

PGI sensor to be the dominating feature since PGI is part of both input and output.

Table 7.4: Results from Random Forest Regressor

Dial	MSE	Pearson-r	Coefficient R^2
(PGI, G6PDH2r)	1.92471e-06	0.999991	0.9999994
(LDH_D, PDH)	3.11794e-07	0.999998	0.999995
(PFL, PDH)	2.84958e-06	0.999987	0.999993
(PTAr, ACALD)	1.33654e-05	0.999917	0.999974
(PTAr, CS)	9.5527e-06	0.999950	0.999981

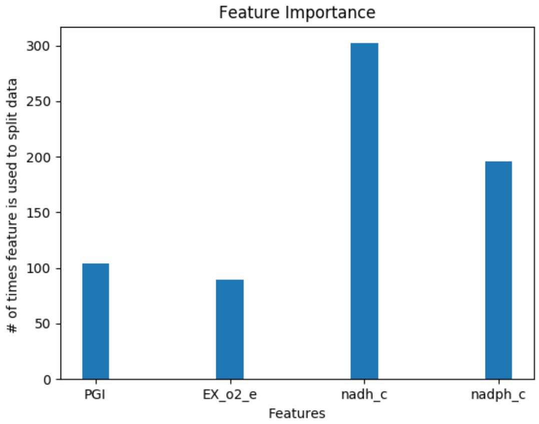


Figure 7.13: XGBoost feature importance plot for (PFL, PDH) dial

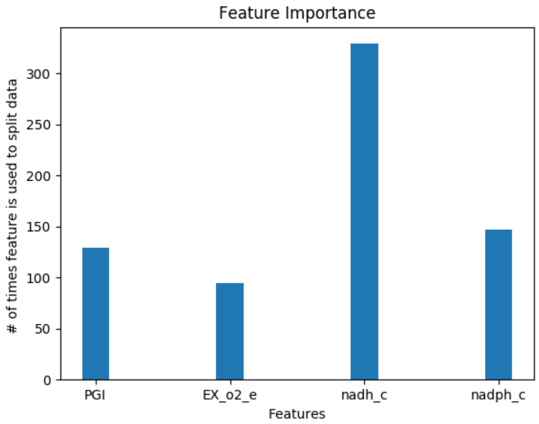


Figure 7.14: XGBoost feature importance plot for (LDH_D, PDH) dial

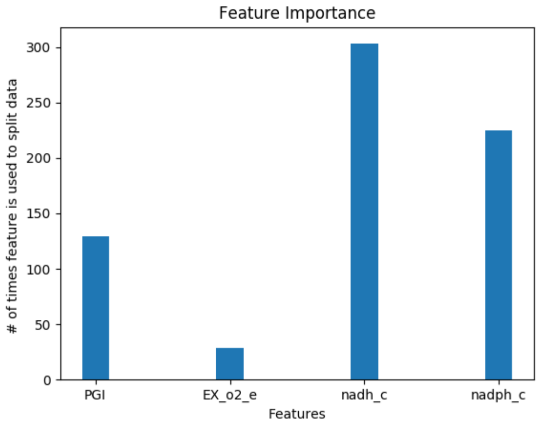


Figure 7.15: XGBoost feature importance plot for $(PTAr, ACALD)$ dial

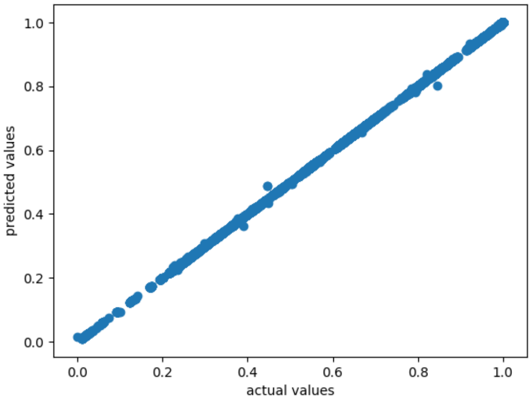


Figure 7.16: Decision Tree Regressor scatter plot of actual vs predict values for $(PGI, G6PDH2r)$ dial

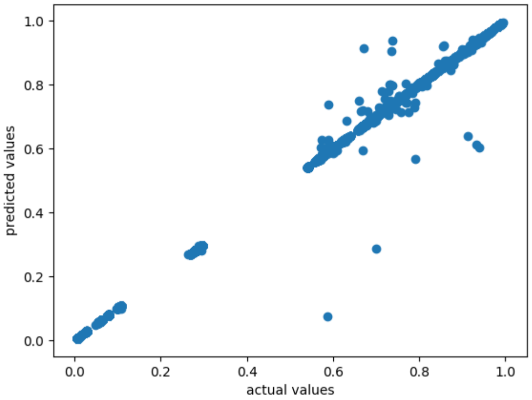


Figure 7.17: Decision Tree Regressor scatter plot of actual vs predict values for $(PTAr, CS)$ dial

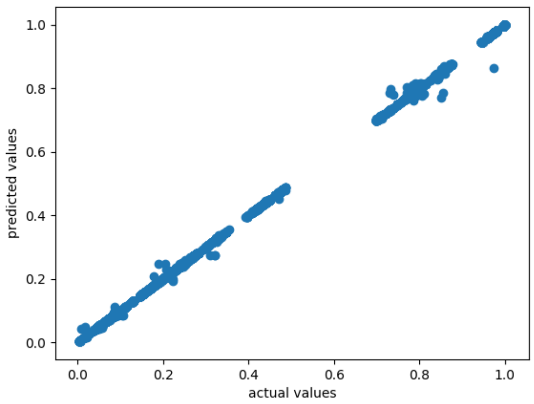


Figure 7.18: Decision Tree Regressor scatter plot of actual vs predict values for (PFL, PDH) dial

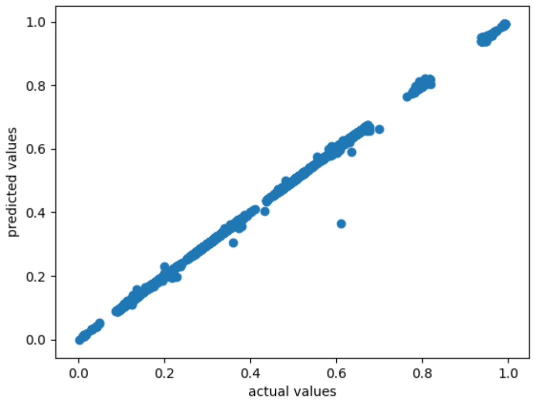


Figure 7.19: Decision Tree Regressor scatter plot of actual vs predict values for (LDH_D, PDH) dial

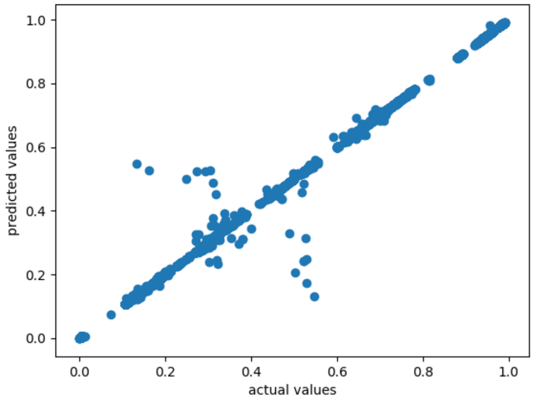


Figure 7.20: Decision Tree Regressor scatter plot of actual vs predict values for $(PTAr, ACALD)$ dial

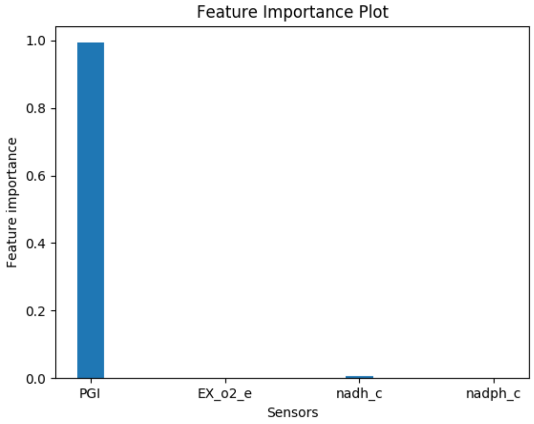


Figure 7.21: Decision Tree feature importance plot for $(PGI, G6PDH2r)$ dial

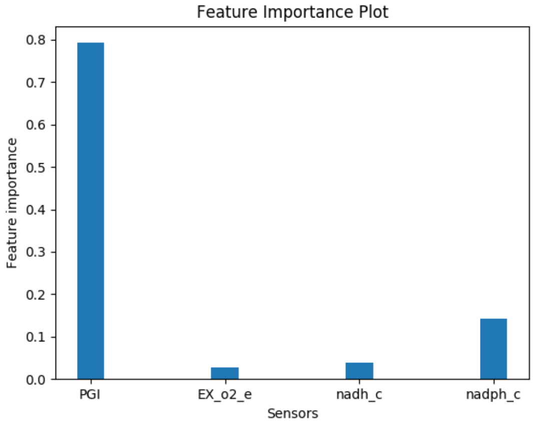


Figure 7.22: Decision Tree feature importance plot for $(PTAr, CS)$ dial

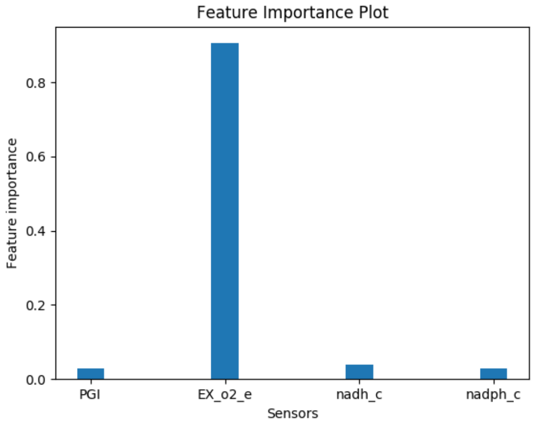


Figure 7.23: Decision Tree feature importance plot for (PFL, PDH) dial

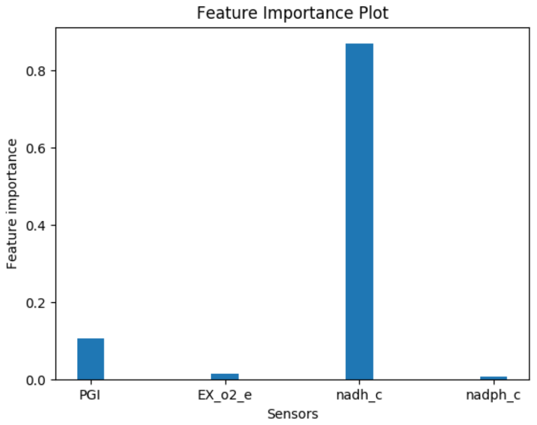


Figure 7.24: Decision Tree feature importance plot for (LDH_D, PDH) dial

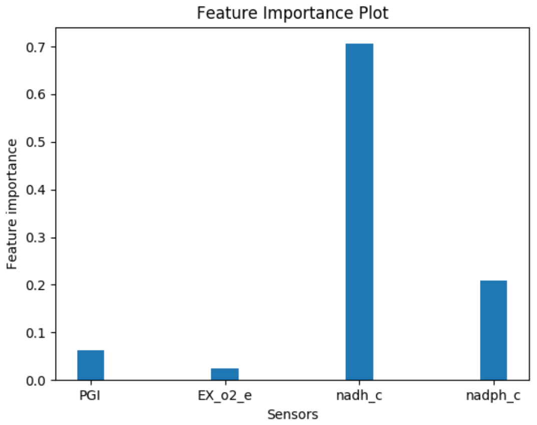


Figure 7.25: Decision Tree feature importance plot for $(PTAr, ACALD)$ dial

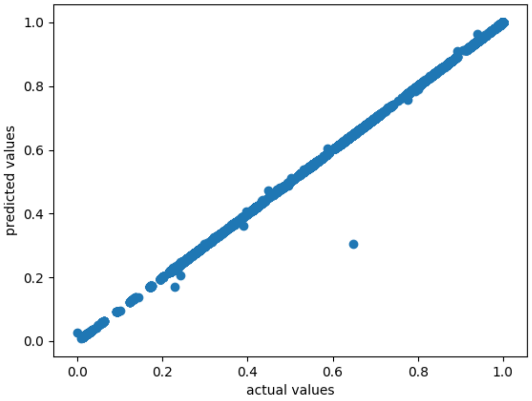


Figure 7.26: Random Forest Regressor scatter plot of actual vs predict values for $(PGI, G6PDH2r)$ dial

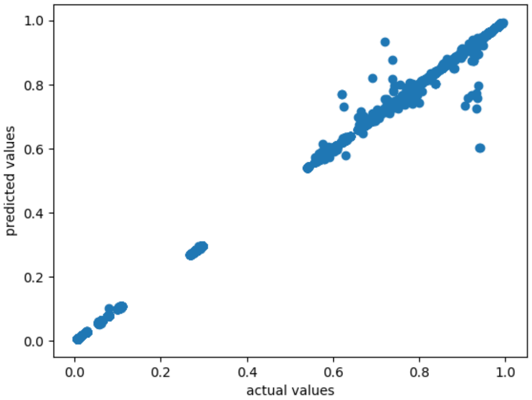


Figure 7.27: Random Forest Regressor scatter plot of actual vs predict values for $(PTAr, CS)$ dial

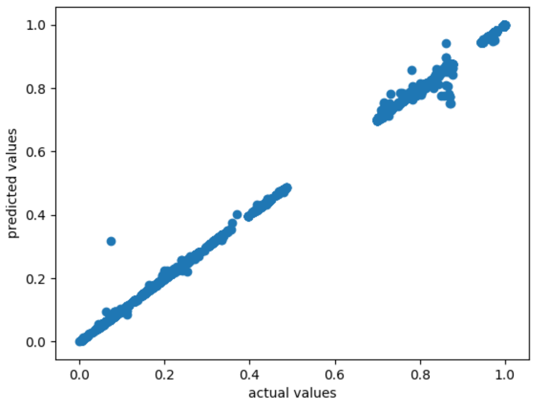


Figure 7.28: Random Forest Regressor scatter plot of actual vs predict values for (PFL, PDH) dial

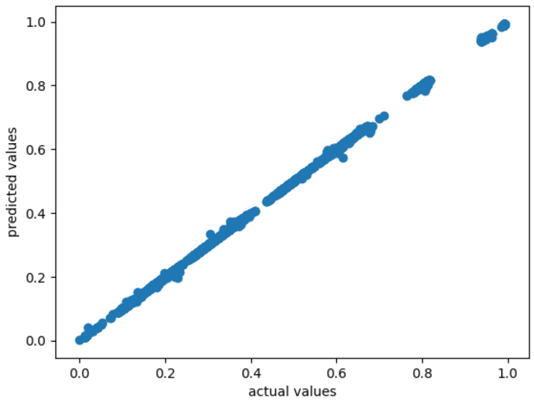


Figure 7.29: Random Forest Regressor scatter plot of actual vs predict values for (LDH_D, PDH) dial

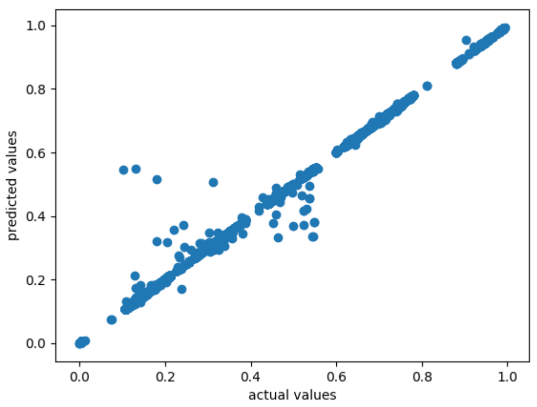


Figure 7.30: Random Forest Regressor scatter plot of actual vs predict values for $(PTAr,ACALD)$ dial

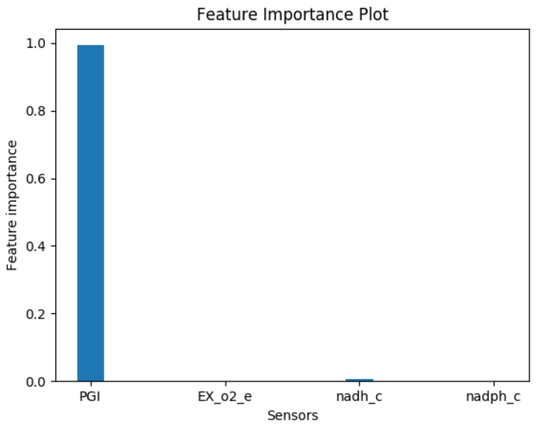


Figure 7.31: Random Forest feature importance plot for $(PGI, G6PDH2r)$ dial

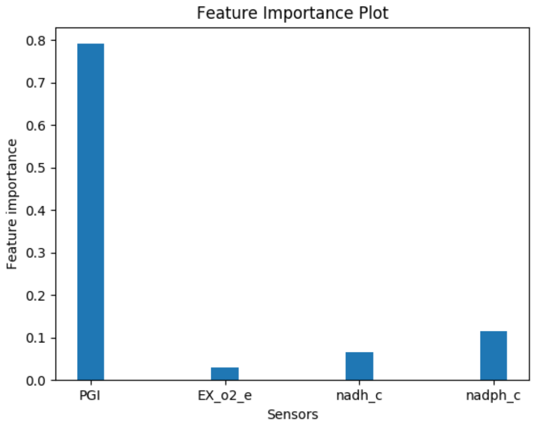


Figure 7.32: Random Forest feature importance plot for $(PTAr,CS)$ dial

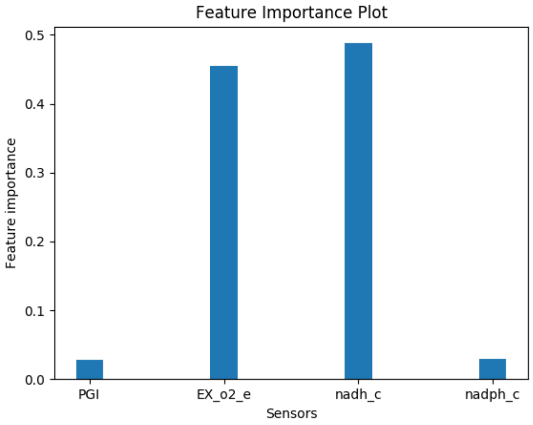


Figure 7.33: Random Forest feature importance plot for (PFL, PDH) dial

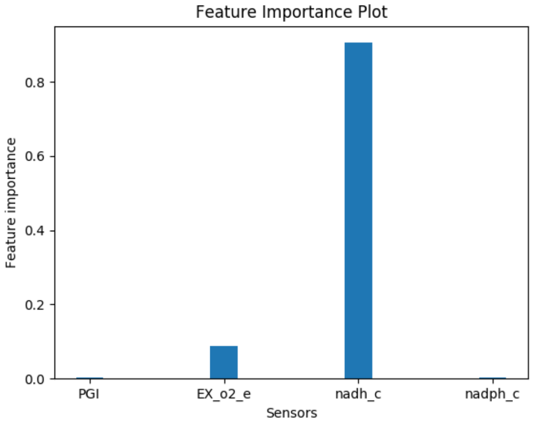


Figure 7.34: Random Forest feature importance plot for (LDH_D, PDH) dial

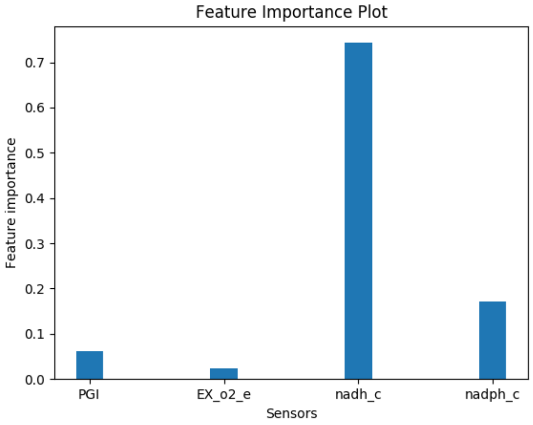


Figure 7.35: Random Forest feature importance plot for $(PTAr, ACALD)$ dial

Chapter 8

Biological Significance of Feature Importance Plots

This chapter talks about the biological significance of feature importance plots by considering three dials namely Pentose Phosphate vs Glycolysis, Fermentation - Acetyl-CoA and Fermentation - Pyruvate dials.

8.1 Pentose Phosphate vs Glycolysis

Figure 7.31 indicates that PGI sensor is the most dominant feature in predicting the (PGI, G6PDH2r) dial. Although, [CF77] and [FL67] indicate that increased flux through the pentose phosphate pathway leads to overproduction of NADPH which would indicate that the dial (PGI, G6PDH2r) should have a high correlation with nadph_c rather than PGI. But from figure 8.1, it is evident that there is a strong correlation between PGI and (PGI, G6PDH2r) dial. This could be due to the fact that the dataset used in this project does not contain sampling conditions where pentose phosphate pathway leads to overproduction of NADPH.

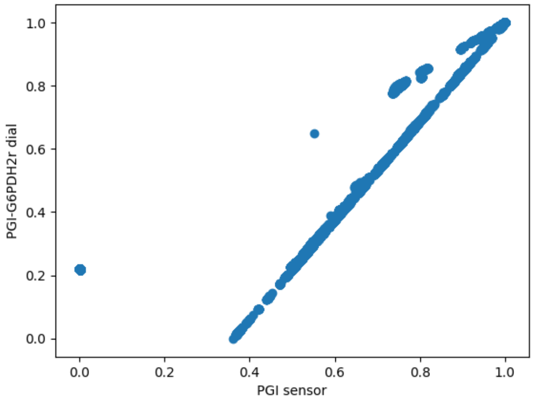


Figure 8.1: Scatter plot of PGI vs (PGI, G6PDH2r) dial

8.2 Fermentation - Acetyl-CoA

Figure 7.32 indicates that PGI sensor is most dominant feature in predicting the (PTAr, CS) dial. Although [HFS98] indicates that the availability of oxygen is the driving factor in the fermentation - Acetyl-CoA (PTAr, CS) dial, the model predicts that glycolysis is as much a good indicator as oxygen.

8.3 Fermentation - Pyruvate

Figure 7.33 indicates that both oxygen and NADH play a key role in predicting (PFL, PDH) dial. Papers [GPP89] and [SDR+12] indicate that presence or absence of pyruvate dehydrogenase affects the levels of NADH in the organism. Fermentation occurs in anaerobic conditions and hence is an indicator of presence or absence of oxygen. The fact that figure 7.33 indicates the same proves that the model predicts the dial values accurately.

Chapter 9

Conclusion

Table 9.1 presents a consolidation of results from all the statistical methods used in this project. The average and standard deviation (stdev) values of MSE, Pearson-r and Coefficient Rˆ2 are calculated for each method and are reported in table 9.1. From the table 9.1, it is evident that through statistical modeling, dial values can be predicted effectively using sensors. From chapter 8, we can see that the model accurately predicts the biological responses of *E. coli* in different media just by using sensors.

Figure 7.33 indicates that both oxygen and NADH are important features for fermentation - pyruvate dial whereas figure 7.23 indicates that only oxygen is an important feature. From section 8.3, it is evident that random forest regressors perform way better than decision tree regressors in terms of modeling the underlying biology.

The research presented in this thesis provides a compelling argument that sensors contain enough information to predict the values of dials. Additional data is not required to model the metabolic activities of *E. coli* MG1655 strain.

Table 9.1: Combined Results from All Methods

Method	Average MSE	Stdev of MSE	Average Pearson-r	Stdev of Pearson-r	Average Coefficient Rˆ2	Stdev of Coefficient Rˆ2
Linear Regression	0.01784	0.007576	0.900479	0.054857	0.80518	0.093599
Extreme Gradient Boosted Trees	0.000811	0.000377	0.99670	0.00188	0.99149	0.004834
Decision Tree Regressor	7.98e-06	6.37e-06	0.99995	4.01e-05	0.99990	7.98e-05
Random Forest Regressor	5.60e-06	4.99e-06	0.999968	3.07e-05	0.999988	9.46e-06

Chapter 10

Future Work

One of the very first steps in the future is to expand the number of sampling conditions. In case of pentose phosphate vs glycolysis dial, there is a high PGI coupling in the dial. By expanding the number of conditions, this high coupling can be avoided and more interesting results can be uncovered. All conditions from evolutionary history of the organisms can be used.

The next step would be to add more sensors and dials to our comprehensive list by looking into literature and by talking to researchers at the Systems Biology Research Group.

A more application oriented next step would be to answer the following question

Can 5 sensors predict flux through all extreme pathways?

If we can predict the flux through all extreme pathways by just using 5 sensor, then we can show that only a small number of degrees of freedom is required for metabolic regulation.

A more ambitious next step would be to constrain the metabolic model using the predictive model described in this research. The predictive model would add further constraints to the metabolic model which at times predicts values which are not achievable by a live organism. The combination of pre-existing metabolic modeling and the predictive model as described in this research can be implemented as a single model using genetic algorithms.

Bibliography

[BLB$^+$13] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[BPB$^+$97] Frederick R. Blattner, Guy Plunkett, Craig A. Bloch, Nicole T. Perna, Valerie Burland, Monica Riley, Julio Collado-Vides, Jeremy D. Glasner, Christopher K. Rode, George F. Mayhew, Jason Gregor, Nelson Wayne Davis, Heather A. Kirkpatrick, Michael A. Goeden, Debra J. Rose, Bob Mau, and Ying Shao. The complete genome sequence of escherichia coli k-12. *Science*, 277(5331):1453–1462, 1997.

[CF77] L. N. Csonka and D. G. Fraenkel. Pathways of NADPH formation in Escherichia coli. *J. Biol. Chem.*, 252(10):3382–3391, May 1977.

[CG16] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[CGKS14] Victor Chubukov, Luca Gerosa, Karl Kochanowski, and Uwe Sauer. Coordination of microbial metabolism. *Nature Reviews Microbiology*, 12:327 EP –, Mar 2014. Review Article.

[CKR$^+$04] Markus W. Covert, Eric M. Knight, Jennifer L. Reed, Markus J. Herrgard, and Bernhard O. Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429:92 EP –, May 2004.

[CP10] Sriram Chandrasekaran and Nathan D. Price. Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in escherichia coli and mycobacterium tuberculosis. *Proceedings of the National Academy of Sciences*, 107(41):17845–17850, 2010.

[CSP01] MARKUS W. COVERT, CHRISTOPHE H. SCHILLING, and BERNHARD PALSSON. Regulation of gene expression in flux balance models of metabolism. *Journal of Theoretical Biology*, 213(1):73 – 88, 2001.

[CXCK08] Markus W. Covert, Nan Xiao, Tiffany J. Chen, and Jonathan R. Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in escherichia coli. *Bioinformatics*, 24(18):2044–2050, 2008.

[FHT$^+$09] A. M. Feist, M. J. Herrgard, I. Thiele, J. L. Reed, and B. ?. Palsson. Reconstruction of biochemical networks in microorganisms. *Nat. Rev. Microbiol.*, 7(2):129–143, Feb 2009.

[FKE$^+$14] Stephen Federowicz, Donghyuk Kim, Ali Ebrahim, Joshua Lerman, Harish Nagarajan, Byung-kwan Cho, Karsten Zengler, and Bernhard Palsson. Determining the control circuitry of redox metabolism at the genome-scale. *PLOS Genetics*, 10(4):1–14, 04 2014.

[FL67] D. G. Fraenkel and S. R. Levisohn. Glucose and gluconate metabolism in an escherichia coli mutant lacking phosphoglucose isomerase. *J Bacteriol*, 93(5):1571–1578, May 1967. 5337843[pmid].

[GBR$^+$13] Emanuel Goncalves, Joachim Bucher, Anke Ryll, Jens Niklas, Klaus Mauch, Steffen Klamt, Miguel Rocha, and Julio Saez-Rodriguez. Bridging the layers: towards integration of signal transduction, regulation and metabolism into mathematical models. *Mol. BioSyst.*, 9:1576–1583, 2013.

[GCSSZ$^+$16] Socorro Gama-Castro, Heladia Salgado, Alberto Santos-Zavaleta, Daniela Ledezma-Tejeida, Luis Muniz-Rascado, Jair Santiago Garcia-Sotelo, Kevin Alquicira-Hernandez, Irma Martinez-Flores, Lucia Pannier, Jaime Abraham Castro-Mondragon, Alejandra Medina-Rivera, Hilda Solano-Lira, Cesar Bonavides-Martinez, Ernesto Perez-Rueda, Shirley Alquicira-Hernandez, Liliana Porron-Sotelo, Alejandra Lopez-Fuentes, Anastasia Hernandez-Koutoucheva, Victor Del Moral-Chavez, Fabio Rinaldi, and Julio Collado-Vides. Regulondb version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Research*, 44(D1):D133–D143, 2016.

[GPP89] L. D. Graham, L. C. Packman, and R. N. Perham. Kinetics and specificity of reductive acylation of lipoyl domains from 2-oxo acid dehydrogenase multienzyme complexes. *Biochemistry*, 28(4):1574–1581, Feb 1989.

[HFS98] C. Hesslinger, S. A. Fairhurst, and G. Sawers. Novel keto acid formate-lyase and propionate kinase enzymes are components of an anaerobic pathway in Escherichia coli that degrades L-threonine to propionate. *Mol. Microbiol.*, 27(2):477–492, Jan 1998.

[HLPP06] Markus J. Herrgård, Baek-Seok Lee, Vasiliy Portnoy, and Bernhard Ø Palsson. Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in saccharomyces cerevisiae. *Genome Res*, 16(5):627–635, May 2006. 16606697[pmid].

[KDE$^+$15] Zachary A. King, Andreas Drager, Ali Ebrahim, Nikolaus Sonnenschein, Nathan E. Lewis, and Bernhard O. Palsson. Escher: A web application for building, sharing, and embedding data-rich visualizations of biological pathways. *PLOS Computational Biology*, 11(8):1–13, 08 2015.

[KLD$^+$16] Zachary A. King, Justin Lu, Andreas Drager, Philip Miller, Stephen Federowicz, Joshua A. Lerman, Ali Ebrahim, Bernhard O. Palsson, and Nathan E. Lewis. Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, 44(D1):D515–D522, 2016.

[KPE03] K. J. Kauffman, P. Prakash, and J. S. Edwards. Advances in flux balance analysis. *Curr. Opin. Biotechnol.*, 14(5):491–496, Oct 2003.

[KVG$^+$13] Karl Kochanowski, Benjamin Volkmer, Luca Gerosa, Bart R. Haverkorn van Rijsewijk, Alexander Schmidt, and Matthias Heinemann. Functioning of a metabolic flux sensor in escherichia coli. *Proceedings of the National Academy of Sciences*, 110(3):1130–1135, 2013.

[KZH10] Oliver Kotte, Judith B. Zaugg, and Matthias Heinemann. Bacterial adaptation through distributed sensing of metabolic fluxes. *Mol Syst Biol*, 6:355–355, Mar 2010. 20212527[pmid].

[LTICV17] Daniela Ledezma-Tejeida, Cecilia Ishida, and Julio Collado-Vides. Genome-wide mapping of transcriptional regulation and metabolism describes information-processing units in escherichia coli. *Frontiers in Microbiology*, 8:1466, 2017.

[MHM14] Wout Megchelenbrink, Martijn Huynen, and Elena Marchiori. optgpsampler: An improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. *PLOS ONE*, 9(2):1–8, 02 2014.

[MLB$^+$17] Jonathan M. Monk, Colton J. Lloyd, Elizabeth Brunk, Nathan Mih, Anand Sastry, Zachary King, Rikiya Takeuchi, Wataru Nomura, Zhen Zhang, Hirotada Mori, Adam M. Feist, and Bernhard O. Palsson. iml1515, a knowledgebase that computes escherichia coli traits. *Nature Biotechnology*, 35:904 EP –, Oct 2017.

[MLGEP08] Jong Min Lee, Erwin P. Gianchandani, James A. Eddy, and Jason A. Papin. Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLOS Computational Biology*, 4(5):1–20, 05 2008.

[OCN$^+$11] Jeffrey D Orth, Tom M Conrad, Jessica Na, Joshua A Lerman, Hojung Nam, Adam M Feist, and Bernhard Ø Palsson. A comprehensive genome-scale reconstruction of escherichia coli metabolism—2011. *Molecular Systems Biology*, 7(1), 2011.

[PVG$^+$11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[SDKZ02] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, Feb 2002.

[SDR$^+$12] Z. Sun, P. M. Do, M. S. Rhee, L. Govindasamy, Q. Wang, L. O. Ingram, and K. T. Shanmugam. Amino acid substitutions at glutamate-354 in dihydrolipoamide dehydrogenase of Escherichia coli lower the sensitivity of pyruvate dehydrogenase to NADH. *Microbiology (Reading, Engl.)*, 158(Pt 5):1350–1358, May 2012.

[SESR07] Tomer Shlomi, Yariv Eisenberg, Roded Sharan, and Eytan Ruppin. A genome-scale computational study of the interplay between transcriptional regulation and metabolism. *Mol Syst Biol*, 3:101–101, Apr 2007. 17437026[pmid].

[TPVP05] Ines Thiele, Nathan D. Price, Thuy D. Vo, and Bernhard O. Palsson. Candidate metabolic network states in human mitochondria: Impact of diabetes, ischemia, and diet. *Journal of Biological Chemistry*, 280(12):11683–11695, 2005.

[WDH$^+$17] Zhuo Wang, Samuel A. Danziger, Benjamin D. Heavner, Shuyi Ma, Jennifer J. Smith, Song Li, Thurston Herricks, Evangelos Simeonidis, Nitin S. Baliga, John D. Aitchison, and Nathan D. Price. Combining inferred regulatory and reconstructed metabolic networks enhances phenotype prediction in yeast. *PLOS Computational Biology*, 13(5):1–23, 05 2017.