การทำนายการแปลงเมทาบอไลท์ในรูปแบบต้นทาง กลางทาง ปลายทาง ที่เป็นไปได้โดยการใช้หลายโครงข่ายประสาทแบบมีการชี้แนะและคุณสมบัติทางเคมี

นางสาวศศิพร ทองแม้น

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556

PREDICTION OF POSSIBLE BEGINNING-INTERMEDIATE-TERMINATING METABOLITE TRANSFORMATION PATTERN USING SUPERVISED NEURAL NETWORKS AND CHEMICAL PROPERTIES

Miss Sasiporn Tongman

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy Program in Computer Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2013

Thesis Title	PREDICTION OF POSSIBLE BEGINNING-INTERMEDIATE-TERMINATING METABOLITE TRANSFORMATION PATTERN USING SUPERVISED NEURAL NETWORKS AND CHEMICAL PROPERTIES
By	Miss Sasiporn Tongman
Field of Study	Computer Science
Thesis Advisor	Professor Chidchanok Lursinsap, Ph.D.
Thesis Co-advisor	Assistant Professor Suchart Chanama, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

..... Dean of the Faculty of Science
(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

..... Chairman
(Assistant Professor Rajalida Lipikorn, Ph.D.)

..... Thesis Advisor
(Professor Chidchanok Lursinsap, Ph.D.)

..... Thesis Co-advisor
(Assistant Professor Suchart Chanama, Ph.D.)

..... Examiner
(Assistant Professor Suphakant Phimoltares, Ph.D.)

..... Examiner
(Professor Boonserm Kijsirikul, Ph.D.)

..... External Examiner
(Associate Professor Manee Chanama, Ph.D.)

..... External Examiner
(Chularat Tanprasert, Ph.D.)

ศศิพร ทองแม้น: การทำนายการแปลงเมทาบอไลท์ในรูปแบบต้นทาง กลางทาง ปลายทางที่เป็นไปได้โดยการใช้หลายโครงข่ายประสาทแบบมีการชี้แนะและคุณสมบัติทางเคมี. (PREDICTION OF POSSIBLE BEGINNING-INTERMEDIATE-TERMINATING METABOLITE TRANSFORMATION PATTERN USING SUPERVISED NEURAL NETWORKS AND CHEMICAL PROPERTIES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ศ.ดร. ชิดชนก เหลือสินทรัพย์, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร. สุชาติ ชะนะมา 103 หน้า.

งานวิจัยนี้ได้ศึกษาปัญหาเส้นทางของการแปลงเมทาบอไลท์โดยใช้วิธีการสร้างตัวแบบซึ่งเรียนรู้แบบมีเป้าหมายจากข้อมูล ปัญหานี้อยู่ในรูปการระบุว่าเมทาบอไลท์ต้นทาง กลางทาง ปลายทางที่กำลังพิจารณานั้น มีความสัมพันธ์อยู่ในเส้นทางการเปลี่ยนแปลงชีวโมเลกุลหรือไม่ และปัญหานี้ยังทำให้อยู่ในรูปปัญหาการแบ่งประเภทแบบมีการชี้แนะ เซตของเมทาบอไลท์ต้นทาง กลางทาง ปลายทาง รวมถึงความสัมพันธ์ที่เป็นคำตอบเป้าหมายได้มาจากกระบวนการค้นหาเส้นทางในกราฟของการเปลี่ยนแปลงชีวโมเลกุลที่เชื่อมต่อกันเป็นโครงข่ายปฏิกิริยาและเมทาบอไลท์ที่เกี่ยวข้อง ส่วนข้อมูลที่เข้าไปเรียนรู้ในตัวแบบได้มาจากการแปลงคุณสมบัติทางโมเลกุลสามมิติของโมเลกุลเมทาบอไลท์ต้นทาง กลางทาง ปลายทาง นอกจากนี้เรายังได้นำเสนอวิธีการปรับปรุงสัดส่วนของสองเป้าหมายให้ใกล้เคียงกันในแต่ละนิยามความสัมพันธ์ที่เป็นคำถามเพื่อสร้างตัวแบบที่ประสิทธิภาพเหมาะสมสำหรับทำนาย ซึ่งเราได้ทำการทดสอบวิธีการที่นำเสนอดังกล่าวข้างต้นกับโครงข่ายการเปลี่ยนแปลงทางชีวโมเลกุลของแบคทีเรีย *E.coli* อีกทั้งทดลองเชิงเปรียบเทียบในข้อมูลเข้าที่ไม่เคยพบมาก่อน ตัวแบบโครงข่ายประสาทที่ได้รับการชี้แนะความสัมพันธ์ทั้งสี่ประเภทได้ทำนายข้อมูลทดสอบเข้าที่ไม่เคยพบมาก่อนในการวัดประสิทธิภาพโดยให้ผลที่ยอมรับได้ เช่น ค่า G-mean $\sim 0.77-0.93$ เมื่อค่า cut-off $= 0.4$. ส่วนการทดลองเชิงเปรียบเทียบผลการทดลองสรุปได้ว่า ทั้งค่าการทำนายจากแต่ละตัวแบบที่รวมเอาการเพิ่มข้อมูลการมีหรือไม่มีความเหมือนกันบางส่วนของโครงสร้างโมเลกุลแบบสองมิติ และค่าเฉลี่ยของค่าจากการทำนายจากสี่ตัวแบบสำหรับทำนายแต่ละความสัมพันธ์ได้แสดงประสิทธิภาพการทำนายโดยทั่วไปที่เหนือกว่าการไม่เพิ่มข้อมูลดังกล่าวในทั้งสี่ประเภทตัวแบบ

ภาควิชา . คณิตศาสตร์และวิทยาการคอมพิวเตอร์ .	ลายมือชื่อนิสิต .....
สาขาวิชา .....วิทยาการคอมพิวเตอร์.....	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ...
ปีการศึกษา .....2556.....	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม ....

4973847223: MAJOR COMPUTER SCIENCE

KEYWORDS: METABOLITE GRAPH / FEED-FORWARD NEURAL NETWORK / IMBALANCED DATA / BIOCHEMICAL TRANSFORMATION NETWORK / SUPERVISED LEARNING / FEATURE PATTERN / PREDICTIVE MODEL

SASIPORN TONGMAN : PREDICTION OF POSSIBLE BEGINNING-INTERMEDIATE-TERMINATING METABOLITE TRANSFORMATION PATTERN USING SUPERVISED NEURAL NETWORKS AND CHEMICAL PROPERTIES. ADVISOR : PROF. CHIDCHANOK LURSINSAP, Ph.D., CO-ADVISOR : ASST. PROF. SUCHART CHANAMA, Ph.D., 103 pp.

The problem of atomically transformable metabolite route by using the supervised learning paradigm was studied. This problem in forms of specifying whether the considered beginning-(intermediate)-terminal metabolite queries are related to any transferable path and was treated as supervised classification problem. A defined metabolite query set and a class target set were obtained from graph path searching strategy in a predefined biochemical transforming graph of reactions and their associated metabolites. For each metabolite query, input data were the transformed features earned by using a computed molecular property set of all metabolites in that query. A method to treat the unequal proportions of binary class training data in all four defined problems was developed. Consequently, the proposed method was tested with the *E.coli* compound transformation networks including comparative tasks with the unseen data. The four predictive models yielded acceptable performance results i.e. G-mean $\sim 0.77 - 0.93$ at output cut-off value = 0.4. In comparative works, the results concluded that both the improved original values and the mean of the improved mean output values provided by adding information about existing/nonexisting compound structure alignment showed the generally better performance in all four defined questions.

Department : .. Mathematics and Computer Science ..	Student's Signature .....
Field of Study : ..... Computer Science	Advisor's Signature .....
Academic Year : 2013	Co-advisor's signature .....

Acknowledgements

Many persons both directly and indirectly help me to shape up this dissertation until it becomes this final one. First person, that I would like to thank is my advisor, Dr. Chidchanok Lursinsap, for his guidance and countless ideas which is valuable not only for this works but also my future researches. Second person, I would like to thank is my co-advisor, Dr. Sushart Chanama, for his continuous suggestion since I was an under-graduated student until I am a Ph.D. candidate. Third person is Dr. Kitiporn Plaimas that I would like to thank for several great and useful discussions. Fourth persons, I would like to thank to people in department of mathematics and computer science, especially in AVIC laboratory, which I regularly receive and give many things including a bunch of knowledge. Fifth persons, I thank to the office of higher education commission for my Ph.D. scholarship that I received since 2006 through the project called the strategic scholarships fellowships frontier research networks. Sixth persons, I would like to thank them for making not too slow progression of this works. One, they are people who had faced technical problems as same as I when writing computer programs. Their questions and answers /hints on the internet still have valuable advantages. Another, they are people who have done various previous works that I have studied their works and also inspired me many ideas for my works. Last persons, I thank to people in my family who always support me and cheer me up all the time from the beginning to the end of the works.

Contents

Page

Abstract (Thai) . . . . . **iv**

Abstract (English) . . . . . **v**

Acknowledgements . . . . . **vi**

Contents . . . . . **vii**

List of Tables . . . . . **x**

List of Figures . . . . . **xii**

Chapter

I Introduction **1**

1.1 Backgrounds 1

1.2 Problem Statement 2

1.3 Objectives 3

II Literature Review **4**

2.1 Path finding problems in metabolic network models 4

2.2 Supervised learning techniques in bioinformatics applications 6

2.3 Imbalanced class distribution solving for supervised learning methods 7

2.4 Dissertation Outline 7

III Methods **10**

3.1 Preliminary Terms And Conditions 10

3.2 Encoding Relevant Metabolites in Paths Problem as Supervised Classification Problems 12

3.2.1 A Metabolite Input Query Set and Binary Class Target Formation 12

3.2.2 Input Feature Calculation and Transformation 14

3.2.3 Artificial Neural Network 16

3.3 Class Imbalance Data Treatment 18

3.3.1 Clustering each class data set into small sub-data 18

3.3.2 Combining each sub-data with additional data from resampling method and calculating the standard deviation of previous and current sub-data 19

3.3.3 Finding border data point sets for each pair of the minority−majority sub-data 21

Chapter Page

3.3.4 Generating new data for both classes with nearly equal distribution 21
3.4 Data Preparation 25
3.5 Performance Evaluation 26

IV Experiments and Results 28
4.1 Training Neural Network Models and Evaluating Model Performance 28
4.2 Comparative Study 43
4.2.1 Sub-pre−training data size impact 44
4.2.2 Cut−off value variation and significance test 46
4.2.3 Unseen data prediction and comparison 57
4.2.3.1 *AUC* performance: sub-model vs. pathway perspective 57
4.2.3.2 Correctness performance: compound perspective 60
4.2.3.3 Correctness performance: pathway perspective 64
4.2.3.4 Traditional map visualization: comparison with original reference maps and metabolite transformation network 67

V Discussion 75
5.1 Atomically convertibility of each considered metabolite input query in biochemical transformation routes from a predefined graph 75
5.2 The four defined supervised classification problems and their corresponding binary answers 76
5.3 The algorithm to handle the seriously imbalanced binary class data before constructing four feed-forward neural network predictive models 77
5.4 The experimental outcomes and comparison 79
5.5 Using the four predictive models in unseen data prediction 81
5.6 Combined four predictive models versus each predictive binary class model in unseen data prediction 82
5.7 Other discussion 84

VI Conclusion 85

References 86

Chapter Page

Appendix **95**

Appendix A Confusion Matrices of Unseen Data Prediction **96**

Biography **103**

List of Tables

Table Page

4.1 *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the 3− fold cross validation for separating class 1 and non-class 1. 36

4.2 *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the 3− fold cross validation for separating class 2 and non-class 2. 37

4.3 *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the 3− fold cross validation for separating class 3 and non-class 3. 37

4.4 *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the 3− fold cross validation for separating class 4 and non-class 4. 38

4.5 *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off = 0.5 including *AUC* performance of two predictive sub-model types, *o* and *n* 40

4.6 The selected parameter values and distribution ratios of class 1 and non-class 1. . . 41

4.7 The selected parameter values and distribution ratios of class 2 and non-class 2. . . 41

4.8 The selected parameter values and distribution ratios of class 3 and non-class 3. . . 41

4.9 The selected parameter values and distribution ratios of class 4 and non-class 4. . . 42

4.10 *AUC* performance of the combined 3-fold testing data of the total $g = 6, 8, 9$ sub-models 44

4.11 The *C* performance evaluation of four defined models in pathway perspectives . . . 65

4.12 Pie charts represent amount of the seen/unseen compounds participated in each metabolite query set in the *C* performance evaluation 66

A.1 The 4×4 confusion matrix of Purine metabolism 96

A.2 The 4×4 confusion matrix of Valine leucine and isoleucine biosynthesis 96

A.3 The 4×4 confusion matrix of Streptomycin biosynthesis 97

A.4 The 4×4 confusion matrix of Nicotinate and nicotinamide metabolism 97

A.5 The 4×4 confusion matrix of Nicotinate and nicotinamide metabolism 98

A.6 The 4×4 confusion matrix of Phospholipid biosynthesis 98

A.7 The 4×4 confusion matrix of Pyruvate oxidation pathway 99

Table Page

A.8 The 4×4 confusion matrix of Fluorobenzoate degradation 99
A.9 The 4×4 confusion matrix of Novobiocin biosynthesis 100
A.10 The 4×4 confusion matrix of Phosphonate and phosphinate metabolism 100
A.11 The 4×4 confusion matrix of Naphthalene degradation 101
A.12 The 4×4 confusion matrix of Nitrotoluene degradation 101
A.13 The 4×4 confusion matrix of Caprolactam degradation 102
A.14 The 4×4 confusion matrix of Biosynthesis of siderophore group nonribosomal peptides 102
A.15 The additional 7 other pathways from KEGG Pathway database 102

List of Figures

Figure		Page
2.1	The work flow diagram illustration.	9
3.1	The problem formulation presentation.	13
3.2	The illustration of 2D binary class imbalance data in generating additional data procedures before clustering each class data set into sub-data sets.	19
3.3	The illustration of 2D binary class imbalanced data in generating additional data procedures before finding the two side border data sets.	20
3.4	Evaluation Metrics.	26
4.1	Parallel coordinate plots of sub-data 1	29
4.2	Parallel coordinate plots of sub-data 2	30
4.3	Parallel coordinate plots of sub-data 3	31
4.4	Parallel coordinate plots of sub-data 4	32
4.5	Parallel coordinate plots of sub-data 5	33
4.6	Parallel coordinate plots of sub-data 6	34
4.7	Precision-Recall graphs of the combined 3-fold testing data of the total $g = 6, 8, 9$ sub-models	45
4.8	Performance evaluation of class 1 vs. non-class 1 prediction model at cut-off value = $0.05, 0.1, 0.15, ..., 0.95$.	48
4.9	Performance evaluation of class 2 vs. non-class 2 prediction model at cut-off value = $0.05, 0.1, 0.15, ..., 0.95$.	49
4.10	Performance evaluation of class 3 vs. non-class 3 prediction model at cut-off value = $0.05, 0.1, 0.15, ..., 0.95$.	50
4.11	Performance evaluation of class 4 vs. non-class 4 prediction model at cut-off value = $0.05, 0.1, 0.15, ..., 0.95$.	51
4.12	Performance evaluation of five different scores at cut-off value = 0.4 of every class q vs. non-class q model where $q = 1, 2, 3, 4$.	52
4.13	Performance evaluation of five different scores at cut-off value = 0.3 of every class q vs. non-class q model where $q = 1, 2, 3, 4$.	53
4.14	Performance evaluation of five different scores at cut-off value = 0.5 of every class q vs. non-class q model where $q = 1, 2, 3, 4$.	54
4.15	Performance evaluation of five different scores at cut-off value = 0.8 of every class q vs. non-class q model where $q = 1, 2, 3, 4$.	55

Figure Page

4.16 AUC performance comparison between the combine model result II by our# method and those results by SCL method for all 4 defined question models. 59
4.17 Distribution amount of pathways according to metabolite types of 50 *Ecoli* pathways from aMAZE 59
4.18 The C performance evaluation of four defined models in compound perspectives . . 63
4.19 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* purine metabolism. 68
4.20 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* valine leucine and isoleucine biosynthesis. 69
4.21 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* streptomycin biosynthesis. 70
4.22 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* methane metabolism. 71
4.23 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* nicotinate and nicotinamide metabolism. 72
4.24 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* phospholipid biosynthesis. 73
4.25 Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* pyruvate oxidation pathway. 74
5.1 An example of the shortest path criteria issue 83

CHAPTER I

INTRODUCTION

1.1 Backgrounds

The intricate cellular processes resulting in energy and growth called metabolisms are the complicated networks composing biochemical substances called metabolites and their transformational mechanisms. The better understanding about the metabolite production and degradation processes leads to the helpful knowledge in many applications e.g. finding drug target (Baths et al., 2011), metabolic flux analysis (Rantanen et al., 2008), metabolic engineering (Finley et al., 2009) and structural network analysis (van Helden et al., 2002). Now, the vast amount of metabolic-related data, namely, genomics, proteomics and metabolomics stored on databases like KEGG and MetaCyc (Caspi et al., 2010; Kanehisa et al., 2008) is enable us to develop a lot of computational approaches (Pitkänen et al., 2010) bringing about the analysis of several metabolic aspects which can be done by building many models using single or multi-omic data based upon what questions are. A variety of biochemical network models in the form of graphs such as metabolic networks, regulatory networks and signal transduction networks have been suggested in an attempt to explain the systems of metabolisms in cells (Deville et al., 2003). The closest realistic detailed metabolic models were presented by the dynamic models because the various data including stoichiometric data and kinetic data have to be integrated, but, nowadays, for some multiple species or whole species dynamic metabolic network model analysis, there are no such entirely required data. The metabolic graph-based models added stoichiometric properties are studied by solving an optimization problem at steady-state when given stoichiometric data and constraints of the defined biochemical reactions sets (Oberhardt et al., 2009). Although, this method is powerful and widely used in many applications (Raman and Chandra, 2009), there are some limitations. For example, to receive the good results, it requires the accurate metabolic network as the predefined system which it is time-consuming procedure if the size of the network is large. Another example, due to alternative objective pathway definitions, there exist some considered routes of metabolites on the metabolic network that are not under the steady-state assumption with stoichiometrically balanced compounds (Félix and Valiente, 2007). For above reasons, the non-stoichiometric metabolic graph-based models still be an es-

sential analysis as not only the basic information for building the stoichiometric models as well as dynamic models (Faust et al., 2011) but also as preliminary methods for investigating such huge metabolic network models (Aittokallio and Schwikowski, 2006). One problem studied by using these models in metabolic network reconstruction and pathway analysis is to predict compound transformation routes when given compounds and their related reactions. In other words, there exists the mechanisms/reactions to transform one compound to others or not (Zhao et al., 2006). To depict the whole steps of the start compounds to the end compounds, a criterion like the shortest path or the extension of shortest path concept can be combined (Rahman et al., 2005).

1.2 Problem Statement

The main problem was defined that whether each metabolite input query received from a pre-defined biochemical transformation graph is in the same routes. A metabolite input query set is composed of the list of possible two end vertices as beginning and terminal metabolites with/without every possibly intermediate metabolite.

This main problem is indirectly studied by stating the following four supervised binary classification problems when each metabolite input query is assigned:

1) whether it is one or two consecutive steps of reaction transformation (class 1 vs. non-class 1),

2) whether it is one step of reaction transformation and multi-step reaction transformations through a certain intermediate metabolite(class 2 vs. non-class 2),

3) whether it is multi-step reaction transformations through a certain intermediate metabolite, and

4) whether it is not all above cases (class 4 vs. non-class 4).

Every binary class target is labeled by routine path searching method on the defined graph. Each input feature pattern associated with its metabolite input query obtains from the proposed numerical transformed properties from each calculated $3D$ molecular property set of its metabolite in such query.

A procedure to treat the imbalanced binary class distribution was also offered, because the imbalanced class distributions of the pre-training data affect the model prediction performances in the supervised learning approaches.

1.3 Objectives

1. To build the supervised feed-forward neural network model for the problems of predicting atomically transferable metabolite in the *E.coli* metabolic pathway set.

2. To offer the four defined types of metabolite transformation as the four binary target classes when given each metabolite input query after preparing the predefined *E.coli* graph consisting of metabolite and reaction links.

3. To present the transformed and reduced features of the input data corresponding to each metabolite query from the computed 3D molecular features using the 2D atomic data of each metabolite.

4. To propose the algorithms to handle the seriously imbalanced binary class data.

5. To show the four trained predictive model performance of the predicted output values, called our result I, and compare them with SCL approach, using various metrics in a few aspects such as the various sub-data size splitting for improving effectiveness, the cut-off value variation and significance test for classifying output value as binary class, as well as the unseen data prediction and comparison.

6. To explore the three adjusted output values of the predicted output values for performance improvement, namely, our result II which is the mean of all four our result I values from all four models for each corresponding input data, our$^{\#}$ result I which the output values are set to negative class if SCL is zero, and our$^{\#}$ result II which is the mean of all four our$^{\#}$ result I values from four models for each corresponding input data set.

CHAPTER II

LITERATURE REVIEW

2.1 Path finding problems in metabolic network models

Metabolic network models built from organism-specific data or multi-organism data can be used for representing elements, e.g. genes, proteins (enzymes) and metabolites, as well as their interactions depending on the studying problems. One key question so-called the path finding problem (Planes and Beasley, 2008) is about searching the paths or eventually getting the systems of transformation processes in the manner that initial substrates are sequentially transformed by each reaction step to final or desired products which is beneficial exploration for several tasks, especially, pathway designs (McClymont and Soyer, 2013) and network studies (Lacroix et al., 2008; Cottret and Jourdan, 2010).

Path finding methods is based on a metabolic reaction network model which mostly represents biochemically systemic processes of metabolite and/or reaction and their relationship as metabolite-metabolite connections and/or reaction-metabolite connections. The consecutive related metabolites at each step of reactions are searched by the path finding methods. Generally, the term 'related metabolites' refers to biochemically relevant, plausible, and/or feasible metabolites (Planes and Beasley, 2008; Faust et al., 2011) in former works.

There are two main metabolic reaction network graph-based models known that each other is complement, stoichiometric models and non-stoichiometric models (Hatzimanikatis et al., 2004), though in some works (Pey et al., 2011) their advantages were combined. The stoichiometric models generally are suitable for specific and detailed purpose of the organism-specific system due to the computational complexity (Schuster et al., 2000; Klamt and Stelling, 2003). Though, there are some modified methods that try to avoid complexity in the large scale stoichiometric model analysis (de Figueiredo et al., 2009; Kaleta et al., 2009), it is still a challenge to improve the better effectiveness. In the stoichiometric approaches (Schilling and Palsson, 1998), they solved the objective function to find a set of steady-state zero flux metabolites when given a set of considered reactions including directions, internal-external metabolites as well as stoichiometric data and a set of constraints, while in the metabolic reaction graph-

theoretic approaches without stoichiometric properties, they used the graph theoretic methods to query the related metabolites of successive reactions according to the graph-theoretic questions such as the shortest step between a source metabolite or a group of metabolites and a target metabolite or a group of metabolites in the considered network (Pitkänen et al., 2005) which can be an organism system or whole organism-merged system.

Without any criteria to identifying related metabolites, the graph-theoretic methods yielded abundant results of related metabolites routes (Küffner et al., 2000) in consequence of combinatorial possibility of related metabolites and meaningless short steps of processes because of absurd related routes passing through some metabolites such as kind of proton/electron carrier and cofactor (Arita, 2004). These metabolites are called pool, currency or side metabolites (Huss and Holme, 2007) which they are found in almost every metabolic reaction. Defining them exactly and removing them from the network causes some missing reactions from the system by result of the context dependent properties of these metabolites (Ma and Zeng, 2003). Later, two core ideas for identifying related metabolites have been offered which they depend on additional data used for avoiding misleading links. The first one have used atomic data represented as atom graph of metabolite and defined mapping methods, for instance, (sub)graph isomorphic approaches and common (sub)graph matching (Raymond and Willett, 2002; Akutsu, 2004; Crabtree and Mehta, 2009; Hattori et al., 2010; Heinonen et al., 2011) so as to identify how each metabolite can be related by the others via some defined measurements i.e. similarity (Raymond et al., 2002; Le et al., 2004). Another idea has used degree connectivity of metabolite nodes called weight in a graph model as searching criteria for obtaining the routes with minimum weight (Croes et al., 2005) thanks to the fact that pool metabolites often contain high weight and they must be avoided by searching procedure. After that, the works based on combining two above ideas have been proposed for finding linear related routes (Blum and Kohlbacher, 2008b) and also branched routes (Pitkänen et al., 2009). Apart from that, RPAIR database (Kotera et al., 2004) build on atomic data is one of KEGG databases storing a list of metabolite pairs as atomically transferable information associated with a set of reactions have been applied by some path finding methods coupled with degree connectivity scheme for linear related pathway searching approach (Faust et al., 2009) and also branched pathways searching approach (Heath et al., 2010). In various path searching conditions, many concepts have been presented which provide different pathway discovery results, example, the (k)shortest path as minimum (k)steps (Arita, 2000; Blum and Kohlbacher, 2008a), the lightest path as the smallest sum of degree connectivity (Faust et al., 2009, 2010), at least one atom conserved or at least a

number of atoms conserved (Mithani et al., 2009; Heath et al., 2011).

At this point, the combined ideas together with different searching path conditions are able to reduce unreasonable connection and yield the quite related linear and branched pathways. In spite of the usefulness of extracted atomic graph properties in path finding conditions, atomic mapping definitions will be useless if there are incomplete or no atomic data in some metabolites as well as their RPAIR data is unavailable. Counting only on the usages of degree connectivity data in the way of the smallest weight metabolite chosen at each searching step, in some reaction whose metabolites are all high or all low degree connectivity, it is not always successful (Croes et al., 2006). Because of the context-specific nature, when given a metabolite pair in one reaction it is difficult to clearly identify each metabolite is the importantly transformable metabolite in that context of a reaction. However, atomically transferable information still be the valuable properties since it reflects the real mechanism that change one metabolite to another metabolite in the one step biochemical process such as a reaction.

2.2 Supervised learning techniques in bioinformatics applications

In the tasks of metabolic reaction prediction, when a novel metabolite with its atomic data has elucidated, the possible biochemical transforming mechanism is predicted by the expert system (Li et al., 2004; Hou et al., 2004) with the well-organized rules from the known mechanisms. Lately, support vector machine (SVM), one of supervised learning approaches, was applied for learning and predicting possible substrates and possible products of well-classified enzyme mechanism (Mu et al., 2011) by feeding the calculated atomic and molecular properties from its optimized 3D atomic structure. In recent times, the prediction of potential enzymatic reactions in metabolic pathways was studied. The chemical fingerprints of compound pairs was converted into feature vectors as input data patterns with binary target class for the SVM binary classifiers to construct model in order to identify that whether the first compound is changed to the second compound in some enzymatic reactions (Kotera et al., 2013).

The supervised learning paradigm is able to extract the knowledge from the representative data as the trained model and use that trained model to predict the new data. Another popular supervised learning approach is the bio-inspired method in the type of artificial intelligent algorithm called artificial neural networks (ANNs). The simple organization of ANNs is composed of an input layer, a hidden layer (neurons) and an output layer. The input layer is fully linked by each weight as the synapse to each neuron. It is widely used in many ar-

eas of applications including bioinformatics tasks e.g. protein structure and function prediction (Wood and Hirst, 2005), gene finding (Browne et al., 2004), gene expression data analysis (Xu et al., 2002) as well as parameter estimation for the small metabolic dynamic model (Almeida and Voit, 2003) due to the powerful ability to design input-output schemes as well as the architectures. Nevertheless, many factors such as suitable number of neurons and epoch must be considered so as to receive an effective model (Zhang, 2000).

2.3 Imbalanced class distribution solving for supervised learning methods

The imbalanced data training usually faces in in real world data with traditional supervised learning that always makes the predictive model trained by unequal class proportion data bias toward the big class data (Sun et al., 2009). Fortunately, many methods were offered to manage and fix this bias (Guo et al., 2008; He and Garcia, 2009). These methods can be categorized into two main approaches. First one, they are aim to manage data such as fixing unequal class proportion data into the approximately equal class proportion data (Chawla et al., 2002; Bunkhumpornpat et al., 2012; Liu et al., 2006) or selecting/generating the most informative data sets as training data sets (Ertekin et al., 2007; Barua et al., 2011). Second one, they are aim to fix the algorithms to make them handle imbalanced data situation effectively (Fu et al., 2002; Hong et al., 2007; Liu and Yu, 2007; Adam et al., 2010; Batuwita and Palade, 2010). Furthermore, there are training techniques which creates a groups of trained models for prediction rather than a single predictive model such as ensemble schemes with sampling techniques (Liu et al., 2006; Estabrooks et al., 2004; Kraipeerapun and Fung, 2009) and multi-binary classification methods (Jeatrakul and Wong, 2012), sometimes one or both above aims are also applied (Yan et al., 2003; Chen et al., 2010; Thanathamathee and Lursinsap, 2013). Apart from managing data, fixing traditional classification algorithms and building series of classifiers, other methods were proposed to solve imbalance data classification, for instance, offering a new metric (Batuwita and Palade, 2009) or feature selection technique(Zheng et al., 2004).

2.4 Dissertation Outline

In this paper, the problem of identifying the relevant metabolites in biochemical transformation routes was considered in a new perspective as a supervised learning problem. First, biochemical transformation was defined from the combined reactions and their involved metabolites (Section 3.1). Second, the problem of identifying the relevant metabolites in biochemical transformation routes was changed into four defined questions suitable for model training by

the feed-forward neural network method, then a defined metabolite query set obtained from the defined graph in the first step used as input query set to seek 'yes'/'no' answer set according to each defined question using conventional graph path searching algorithm (Sections 3.2.1 and 3.4). Third, input feature data set, the transformed molecular properties calculated from 3D atomic structures, was prepared by using a defined metabolite query set and the associated answer set for each defined question was target classes for each defined question (Sections 3.2.2 and 3.4). Fourth, the data division method was applied to split the prepared data into adequate size. Since imbalanced data problem occurred, the proposed method was offered to handle this problem. So, they were trained by the feed-forward neural network and selected the sub-models with effective performance. After that, sub-models were combined in the predictive model for each defined question (Sections 3.2.3, 3.3 and 4.1). Fifth, various performance evaluations were done in many output score types of each predictive model corresponding to each defined question for the following comparison with the strength of chemical linkage approaches: the pre-training sub-data size varying, the cut-off score value variation and significance test, the unseen data prediction and comparison in a sub-model aspect, a conventional pathway aspect, and a compound aspect (Section 4.2). Finally, the whole works were discussed and concluded (Sections 5 to 6). The work flow diagram was briefly presented by Figure 2.1.

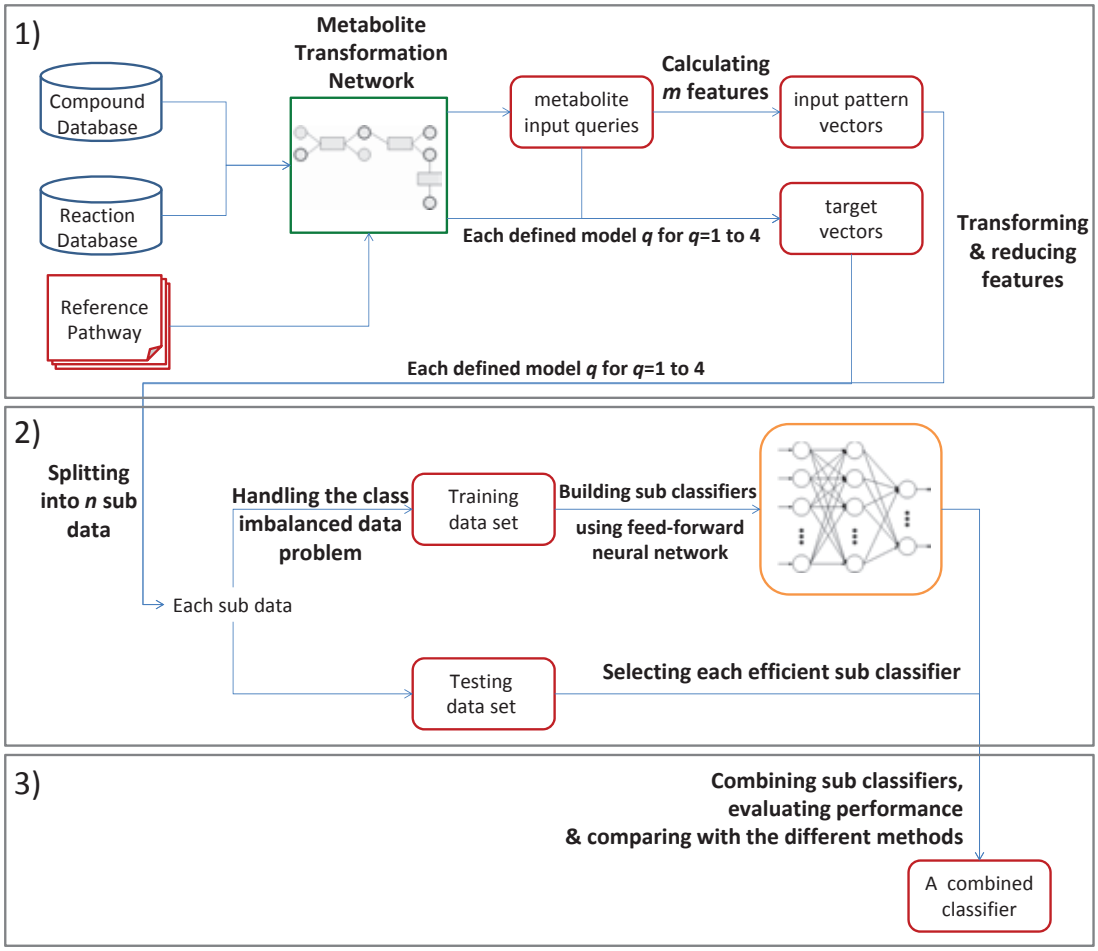


Figure 2.1: The work flow diagram illustration. 1) The defined graph model is converted into the input-target data set which can be trained and built predictive models; 2) According to the large data set from step 1, a data set is divided. Next, the class imbalanced sub-data is handled by our proposed methods. Later, each approximately balanced sub-data is trained and performance of models with chosen parameters are evaluated; 3) The effective sub models are integrated. After that, more performance metrics are evaluated and analysed with comparing methods.

CHAPTER III

METHODS

The defined terms for metabolite transformation, the defined problem as supervised classification problems, the proposed method to handle the seriously unequal proportions of class distribution a in data set, and the data preparation for experiments are described.

3.1 Preliminary Terms And Conditions

Let u and v be two metabolites.

A metabolite pair,(u,v): (u,v) is called *transformable metabolite pair* if u and v are in at least one plausible metabolite path. Otherwise, they are called *non-transformable metabolite pair*. Denote that *forward* and *backward direction* are omitted from consideration, so, (u,v) and (v,u) indicate the same transformation.

A transformation process, R_i: it is a process concerning a metabolite pair, (u,v) such that u can be changed into v in one step, or vice versa. In this work, it is defined that there exists at least one reaction $r \in R_i$ that can make such biochemical transformation happens.

One step transformable metabolite pair: (u,v) is called *one step transformable metabolite pair* if 1) u and v are in the same transformation process, in this case, a reaction set R_i; 2) u is a substrate and v is a product, or vice versa; and 3) u and v appear on the reference map.

Multi-step transformable metabolite pair: (u,v) is called *multi-step transformable metabolite pair* if 1) u and v are two-end metabolites of at least one plausible metabolite path with length more than 2;and 2) u and v appear on the reference map.

In the supervised data set preparation(see Section 3.4), every *metabolite pair* was extracted from *a metabolite set* of *all reactions* in the whole considered *reference maps* from Lemer et al. (2004).

Let *b*, *t*, and *n* be the beginning, terminal, and intermediate metabolites, respectively.

Metabolite Transformation Network, **K**: a simple fully-connected undirected graph, $\mathbf{K} = (\mathbf{V}, \mathbf{E})$. Denote that 1) $\mathbf{V} = \mathbf{V}' \cup \mathbf{V}''$ is a non-empty finite metabolite set where $\mathbf{V}'$ and $\mathbf{V}''$ are two metabolite lists that one appears and another does not appears on *reference maps*, respectively; and 2) two edge sets, $\mathbf{E}'$ and $\mathbf{E}''$ where $\mathbf{E} = \mathbf{E}' \cup \mathbf{E}''$, represent *one step transformable metabolite pair* and *other pairs*, respectively.

A plausible metabolite path, **P**: given **K**, *a plausible metabolite path*, $\mathbf{P} = (b, ..., n, ..., t)$, is a simple path such that 1) *n* is one intermediate metabolite locating on **P** between two-end metabolites *b* and *t*; and 2) every edge $e \in \mathbf{E}'$, otherwise, it is *non-plausible path*. Denote that 1) *a plausible metabolite path* with length one has no *n* and it is so-called *one step transformable metabolite pair*; and 2) **P** can be considered either *forward* and *backward path*.

A metabolite input query set, **H**: given *b*, *t*, and *n*, determine whether *b* is transformable to *n* and *n* is transformable to *t*. Every *metabolite query*, $\mathbf{h_j} \in \mathbf{H}$ such that $\mathbf{h_j} = (b, n, t)$ or $\mathbf{h_j} = (b, t)$ is obtained from a given **K**.

3.2 Encoding Relevant Metabolites in Paths Problem as Supervised Classification Problems

The representation as a graph allows us to observe in many sides when the graph problem is well-defined. In this paper, the problem of plausible metabolite path is transformed into supervised binary classification problems of predicting whether the given beginning and terminal metabolites, sometimes including an intermediate metabolite, are in some plausible metabolite path. The associated features of each metabolite query, beginning, intermediate and terminal metabolites, are also proposed as in the following sections.

3.2.1 A Metabolite Input Query Set and Binary Class Target Formation

From the defined graph **K** and some statements in previous section, a metabolite input query set $\mathbf{H} = \{\mathbf{h}_j | j = 1, 2, ..., l\}$ is given. Each input query **h** reflects to several questions from the graph **K**. Here, with respect to the path from b to t including the determination of n locating this path, the binary targets for the following four basic transformation questions can be assigned:

Question 1: 1) if (b, n, t) is considered, whether (b, n) and (n, t) are both *one step transformable metabolite pair*; or 2) if (b, t) is considered, whether (b, t) is *one step transformable metabolite pair*.

Question 2: whether (b, n) is *one step transformable metabolite pair* but (n, t) is *multi-step transformable metabolite pair*, or vice versa.

Question 3: whether (b, n) and (n, t) are both *multi-step transformable metabolite pair*.

Question 4: whether (b, n, t) does not meet any above conditions in questions 1, 2, or 3.

For each binary target set, the class with a large data set is called *majority data* class and the other class is called *minority data* class. Hence, each question and its answer can be interpreted as each input feature data set and its class target in a two-class pattern recognition problem. Many algorithms with supervised learning methods can be applied for constructing classifiers that can efficiently guess the class of each new data pattern.

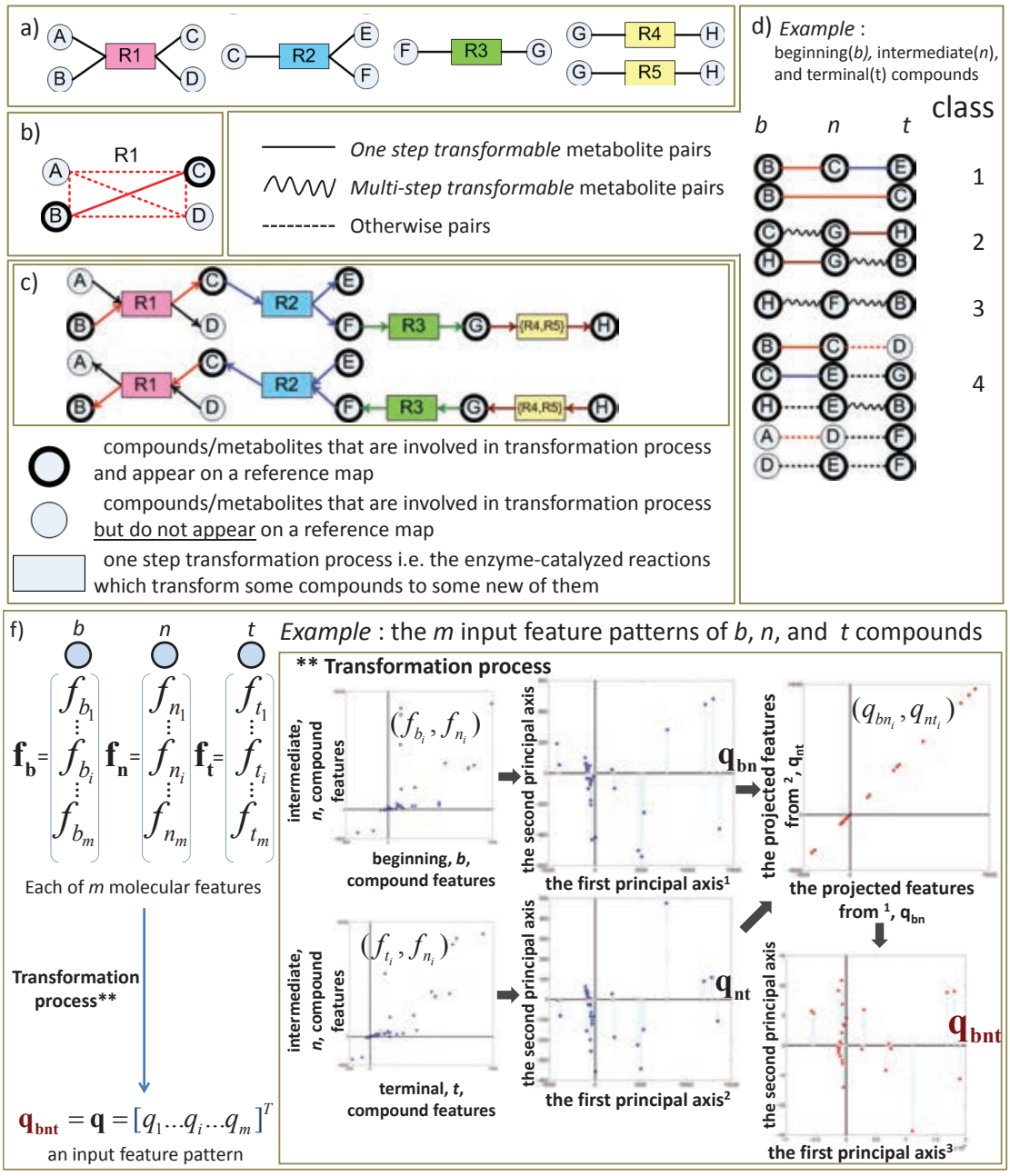


Figure 3.1: The input-target formation presentation. a) A set of example reactions and their involved compounds; b) An example of all possible links in a reaction R1; c) The complete graph showing all possible links from the example set in a); d) Forward and backward transformation routes in the form of two direct graphs from the example set in a); e) Each example of each class compound query set acquired from the defined complete graph in c), when each each class compound query set reflects to each defined question q for $q = 1, 2, 3, 4$; f) The proposed process to transform each of m 3D features in a metabolite query to an input feature pattern ready for being trained by neural network method(see *Transformed feature procedure* in details).

3.2.2 Input Feature Calculation and Transformation

Considering each metabolite input query, each molecular property of each metabolite was calculated. These molecular properties (Mu et al., 2011) consists of the surface, shape, energy and charge distribution of a $3D$ compound molecule. Let $\mathbf{f_b} = [f_{b_1}, ..., f_{b_s}]^T$ be a vector storing s properties of a beginning metabolite b, similarly, let $\mathbf{f_t}$ and $\mathbf{f_n}$ be a terminal metabolite t and an intermediate metabolite n if n exists, respectively. The aim is to create an input pattern $\mathbf{a}$ as the s representative properties from $\mathbf{f_b}$, $\mathbf{f_t}$ and $\mathbf{f_n}$ if n exists (see $Transformed\ feature\ procedure$).

For each metabolite query in the training and the testing data, the number of features may be equal to 81×3 for 3-tuple $(m^{(b)}, m^{(n)}, m^{(t)})$ and 81×2 for 2-tuple $(m^{(b)}, m^{(t)})$. This number of features is too large for computation and it must be reduced. The following process was proposed to reduce the number of features.

Let $\mathbf{f}_b$, $\mathbf{f}_n$, and $\mathbf{f}_t$ be sets of features for $m^{(b)}$, $m^{(n)}$, and $m^{(t)}$, respectively.

Transformed feature procedure

Case 1: 2-tuple $(m^{(b)}, m^{(t)})$

1. Let $\mathbf{F}^{(b,t)} = \left[\begin{pmatrix} f_{b,1} \\ f_{t,1} \end{pmatrix} \cdots \begin{pmatrix} f_{b,s} \\ f_{t,s} \end{pmatrix} \right]^T$.
2. Compute the covariance matrix $\mathbf{C}$ from $\mathbf{F}^{(b,t)}$.
3. Let $\mathbf{u}$ be the eigenvector of $\mathbf{C}$ whose eigenvalue is maximum.
4. Compute new feature vector $\mathbf{q}^{(b,t)} = \mathbf{u}^T\mathbf{F}^{(b,t)}$ for 2-tuple $(m^{(b)}, m^{(t)})$.

Case 2: 3-tuple $(m^{(b)}, m^{(n)}, m^{(t)})$

1. Compute new feature vector $\mathbf{q}^{(b,n)} = \mathbf{u}^T\mathbf{F}^{(b,n)}$ for 2-tuple $(m^{(b)}, m^{(n)})$ as in Case 1 procedure.
2. Compute new feature vector $\mathbf{q}^{(n,t)} = \mathbf{u}^T\mathbf{F}^{(n,t)}$ for 2-tuple $(m^{(n)}, m^{(t)})$ as in Case 1 procedure.
3. Let $\mathbf{Q}^{(b,n,t)} = \left[\begin{pmatrix} q_1^{(b,n)} \\ q_1^{(n,t)} \end{pmatrix} \cdots \begin{pmatrix} q_s^{(b,n)} \\ q_s^{(n,t)} \end{pmatrix} \right]^T$.
4. Compute the covariance matrix $\mathbf{C}$ from $\mathbf{Q}^{(b,n,t)}$.
5. Let $\mathbf{u}$ be the eigenvector of $\mathbf{C}$ whose eigenvalue is maximum.
6. Compute new feature vector $\mathbf{a}^{(b,n,t)} = \mathbf{u}^T\mathbf{Q}^{(b,n,t)}$ for 3-tuple $(m^{(b)}, m^{(n)}, m^{(t)})$.

Feature vector $\mathbf{a}^{(b,n,t)}$ is used as an input patttern of the training and the testing data.

3.2.3 Artificial Neural Network

In the general concept of the supervised learning technique (Bishop, C.M., 2006), a training data set along with class target $\mathbf{\Gamma} = \{(\mathbf{a}_j, \mathbf{c}_j) | j = 1, 2, ..., l\}$, where $\mathbf{a}_j \in \Re^m$ and $\mathbf{c}_j \in \Re^n$, is used for building a particular model in order to predict the value of each class **c** associated with each new input pattern $\tilde{\mathbf{a}}$.

The feed-forward neural network with a hidden layer is one among suitable model for supervised classifying data patterns (Haykin, 1998). A chosen architecture consists of three layers in fully connected structures, namely, input layer, hidden layer and output layer. In the forward direction from one layer to another, excepting input layer which delivers a training data set for a network system, each layer does a linear combination of its served inputs in each neuron where coefficients and bias are gathered as adjustable parameters, after that its outputs are yielded by taking a differentiable function in each neuron. In each round of training, parameters are adjusted to reduce the differences between class target set and class output set in the form of an error function.

Given an input pattern vector $\mathbf{a} = [a_1, ..., a_i, ..., a_m]^T$ and its class target vector $\mathbf{c} = [c_1, ..., c_k, ..., c_n]^T$, the k^{th} element in its output vector $\mathbf{o} = [o_1, ..., o_k, ..., o_n]^T$ can be shown as

$$o_k = f_k^{(2)}\left(\sum_{j=1}^{s} w_{kj}^{(2)} f_j^{(1)}(z_j) + w_{k0}^{(2)}\right)$$

such that $z_j = \sum_{i=1}^{m} w_{ji}^{(1)} a_i + w_{j0}^{(1)}$ where $w_{j0}^{(1)}$ as well as $w_{k0}^{(2)}$ are biases and $w_{ji}^{(1)}$ as well as $w_{kj}^{(2)}$ are coefficients. Denote that $f_k^{(2)}(\cdot)$ is an activation function of the k^{th} neuron in the output layer producing output o_k , similarly, $f_j^{(1)}(\cdot)$ which is a function of the j^{th} hidden neuron in the hidden layer. Apart from that s and m are the number of hidden neurons in the hidden layer and the size of input features in the input layer, respectively.

In each round r of network training, an error function to measure the learning performance is

$$\xi = \frac{1}{2} \sum_{i=1}^{l} \sum_{k=1}^{n} \left(c_k^{(n)} - o_k^{(n)}\right)^2$$

where n and l are the number of neurons in the output layer and the number of input patterns, respectively. The aim is to find a value set of parameters that produces the acceptable minimum error ξ which can be obtained by using one of various numerical optimization techniques. All

parameters gathered as weight vector $\mathbf{w}$ initialized as $\mathbf{w}_0$ is iteratively updated by scaled conjugate gradient (SCG) algorithm (Moller, 1993) developed from the conjugate gradient (CG) method and Levenberg-Marquardt algorithm in order to decrease error ξ until reach the desired small value using the information of the partial derivatives with respect to weights. The SCG weight-update rule is given by $\mathbf{w}_{r+1} = \mathbf{w}_r + \Delta\mathbf{w}$ such that $\Delta\mathbf{w} = \alpha_r\mathbf{p}_r$ where a learning rate parameter α and a conjugate gradient direction $\mathbf{p}$ are systemically adapted by some rules. Denote that $\mathbf{p}_0$ is a steepest descent direction at an initial round. The benefit of SCG technique is the effective convergence resulting from no computation of line search procedure in calculating α unlike the original CG method.

In practical situation, when a training data set along with class target $\mathbf{\Gamma}$ is very big, it will take a long time to yield an appropriate trained network model. Therefore, to reduce the time of training such one big network, $\mathbf{\Gamma}$ is partitioned by key feature concept into g disjoint sub-data set, $\tau_1 ... \cup \tau_i \cup ... \tau_g = \mathbf{\Gamma}$. Each sub-data set τ_i is trained by sub-network i in parallel fashion (Plaimas et al., 2005). The final output results from all sub-networks are determined by defined criteria. By this way, the complexity of data and training time would be reduced. Shortly, the key feature is one feature of data set used as an identifier for dividing a data set into several small groups which helps to build supervised classification sub-models, simultaneously. First, the range of all values in a data set is found. Second, the data set is equally divided by the range into n intervals. After that, an important interval which contains the maximum number of values considered from the entire data set is discovered. Next, for each feature of the whole data set, the numbers of values that are members of an important interval are counted, then, the minimum one is defined as a key feature.

3.3 Class Imbalance Data Treatment

From previous sections, the path inference problems can be expressed as the four defined questions of the supervised binary classification problems. Once an input pattern set associated with its metabolite input query $\mathbf{A}$ was prepared, the class target $\mathbf{c}_j^{(q)} \in \{0,1\}$, where $(\mathbf{a}_j, \mathbf{c}_j^{(q)}) \in \mathbf{\Gamma}^{(q)}$ and $\mathbf{a}_j \in \mathbf{A}$, corresponding to binary answer of a defined question q was assigned. Hence, the four training data sets along with class targets, $\mathbf{\Gamma}^{(q)}$ for $q = 1, 2, 3$ and 4, were available for training each neural network model resulting in the four classifiers, one for each defined question-answer. However, each class occurrence in each training data set is not equal frequency. In fact, they are very imbalanced. Without a procedure to deal with these class imbalance problems, the performances of the trained neural network models are inefficient, especially, in the minority class prediction (Visa, 2005). Thereby, to improve the performance of such trained models, the proposed algorithm was designed for handling the highly class-imbalanced training data based on combination of local under samplings and local over sampling manners including a defined nearest neighbor rule for generating added minority data applied on a particular situation. Some protocols were adapted from the recent methods (Thanathamathee and Lursinsap, 2013).

In brief, there are the following four main processes. First, each class data set is clustered into small sub-data. Second, for each sub-data, the standard deviations before and after combining sub-data with additional data from resampling method are computed. Third, the border data point sets for each pair of the minority-majority sub-data are found out. Finally, the new data are generated for both classes with approximately equal distribution using each border data point set and the corresponding difference of previous and current standard deviations. The following is the explanation of each process.

3.3.1 Clustering each class data set into small sub-data

Based on a chance that each class data set complexly locates in the data space, this entire class data set would be hard to be separated. An unsupervised clustering method helps to partition each class data set into many sub-groups, so, the imbalanced situations between the paired sub-groups of the binary class are easily managed by the next procedures. Similar to the recent work (Thanathamathee and Lursinsap, 2013), the self-organizing map (SOM) method was applied.

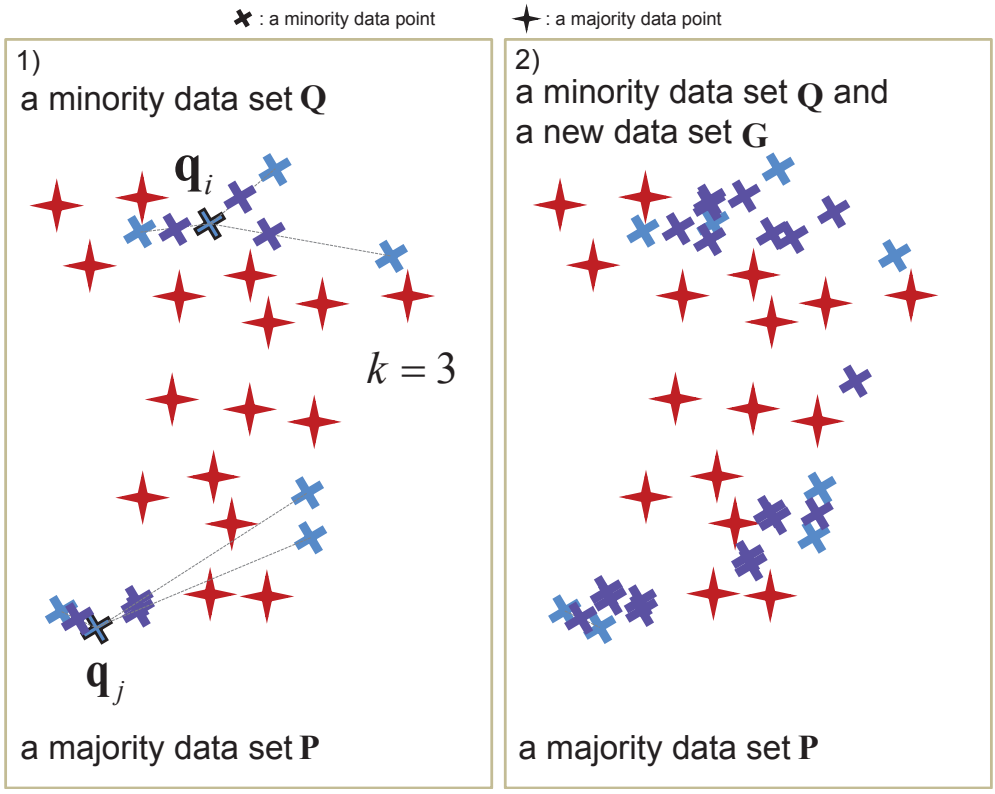


Figure 3.2: The illustration of 2D binary class imbalance data in generating additional data procedures before clustering each class data set into sub-data sets. Synthesized data are depicted as purple points. 1) The concept of algorithm 1 *Generating Synthetic Inner-Class Data* is depicted when nearest minority neighbor $k = 3$. 2) The result after using algorithm 1 is shown.

3.3.2 Combining each sub-data with additional data from resampling method and calculating the standard deviation of previous and current sub-data

Bootstrap method was implemented for estimating the natural standard deviation of each sub-cluster by repetitively sampling data with replacement. Subsequently, the difference of two standard deviations from bootstrap method and the initial standard deviation was used as information to position the new synthetic data for lessening the imbalanced data situations (Thanathamathee and Lursinsap, 2013). But, in this paper, not only Bootstrap strategy but also two proposed procedures to handle extremely imbalanced data were proposed.

First process, the new generated data are added to minority class data (Algorithm *Generating Synthetic Inner-Class Data*) before the process 3.3.1. This procedure considers the whole binary class data. For each data point **q** in the minority data set, no more than new k minority data points are generated by finding k nearest minority data points in the form of Euclidean distance and locating them along each line between q and each of k nearest data points with a

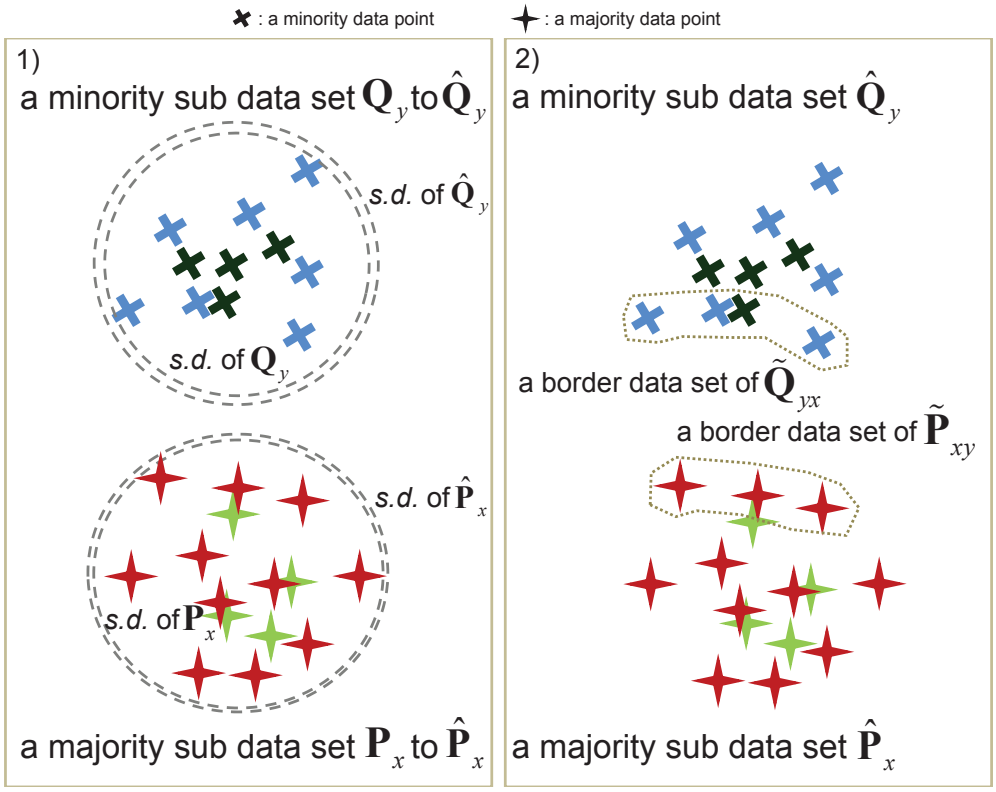


Figure 3.3: The illustration of 2D binary class imbalanced data in generating additional data procedures before finding the two side border data sets. Synthesized data are depicted as dark/light green points. 1) The concept of resampling technique called Boostrapping within the last step of creating a nearly balanced training data set process is depicted. 2) Two closet data sets after applying algorithm 2 *Identifying Boundary Data* are represented.

distance no longer than the minimum length (see Algorithm 1 and Figure 3.2). This concept is supposed to help increasing amount of the data points and also to expand the possibly occupied data space of the minority class data before the clustering process.

Second process, unlike the recent method (Thanathamathee and Lursinsap, 2013), bootstrap method was also used in slightly different ways before finding the rim data of each minority-majority sub-cluster pair on the next process. Bootstrap method was performed in order to both create the new additional data and calculate the difference of the previous and the current standard deviations which was used in the last procedure. This process considers each of minority−majority sub-clusters. For each sub-data with size l, initially, the standard deviation is calculated. Next, data are sampled with replacement for l times. Then, sampled data are averaged as a new additional data to the sub-data. These are repeatedly done until reaching the desired amount of additional data. After that, the current standard deviation is calculated (see Figure 3.3). Because the severely imbalanced data as well as the sub-cluster division yield

the tiny size in the terms of space and quantity of particular sub-clusters, in some cases, a border data set is possibly represented by that whole sub-cluster. The above procedure tries to carefully enlarge such sub-clusters by embedding each new additional data set. Thus, these combined sub-data are used for finding the border data sets. The new additional data could be one member of the border data sets which are expected to increase the chance of the built model to predict the new incoming data, effectively.

3.3.3 Finding border data point sets for each pair of the minority−majority sub-data

Every minority−majority sub-data pair contains two border data sets on each side. These border data sets are identified by using Hausdorff distance concept which relied on Euclidean distance (see Algorithm 2 and Figure 3.3) similar to the previous work (Thanathamathee and Lursinsap, 2013). Each data point in one border side is the nearest point for at least one point in the whole opposite side. This method helps to discard unnecessary data and retain only crucial data in building separating model.

3.3.4 Generating new data for both classes with nearly equal distribution

At this point, the nearly balanced training data in binary class distribution are prepared by synthesizing new data from the border data sets. The size of entirely balanced both class data should be adequately available for providing information to efficiently build each predictive model. Hence, first added synthetic data set is based on a rough criterion that the number of small class data in each class should not be less than the dimension of data. Later, new more synthetic data is added in order to obtain the nearly balanced training data set ready for training supervised neural network models (see algorithm 4). The synthetic data are generated. Each border data point is added by the standard deviation difference of its sub-cluster scaled by random values from $[-1,0) \cup (0,1]$ (see algorithm 3-4).

Therefore, the feed-forward neural network models with a sufficiently certain number of epochs were trained by feeding the approximately balanced training data from the new synthetic data including the border data sets. Afterwards, the performances of models were measured.

Creating a nearly balanced training data set process

1. Generate synthetic inner-class data of minority class by using Algorithm *Generating Synthetic Inner-Class Data*.
2. Identify clusters in each class by using self-organizing mapping method.
3. For any two clusters from different classes, find their boundary data by using Algorithm *Identifying Boundary Data* which the concept of Hausdorff distance between two sets is applied as in Thanathamathee and Lursinsap (2013).
4. Adjust imbalanced class ratio to the nearly balanced class ratio by generating new synthetic boundary data of each cluster using Algorithm *Generating Synthetic Boundary Data* and Algorithm *Adjusting Number of Boundary Data*.

Algorithm 1: *Generating Synthetic Inner-Class Data*

1. Let **d** denote a considered data point in a minority data class. **d** can be referred as $\mathbf{q}^{(b,t)}$ in case 1 or $\mathbf{a}^{(b,n,t)}$ in case 2 in Section 3.2.2
2. **For** each data point **d do**
3. Let **K** be a set of k original nearest data points of **d**.
4. Let ℓ be minimum values of $\|\mathbf{d}-\rho\|$ where $\rho \in \mathbf{K}$.
5. **For** each data point $\rho \in \mathbf{K}$ **do**
6. **If** $\frac{\|\mathbf{d}-\rho\|}{2} > \ell$ **then**
7. Generate a new data point
$$\mathbf{d}' = \mathbf{d} + \left(\frac{\ell}{\|\mathbf{d}-\rho\|}(\rho - \mathbf{d})\right).$$
8. **else**
9. Generate a new data point
$$\mathbf{d}' = \frac{\mathbf{d}+\rho}{2}.$$
10. **EndIf**
11. **EndFor**
12. **EndFor**

$\|\mathbf{d}-\rho\|$ is the Euclidean distance between **d** and ρ.

Let $\mathbf{c}^{(A)}$ and $\mathbf{c}^{(B)}$ be two close clusters from classes *A* and *B*, respectively. The following algorithm is for identifying the boundary data of two close clusters $\mathbf{c}^{(A)}$ and $\mathbf{c}^{(B)}$.

Algorithm 2: *Identifying Boundary Data*

---

1. Let $\mathcal{B}^{(A)} = \phi$ be a set of boundary data for $\mathbf{c}^{(A)}$.
2. Let $\mathcal{B}^{(B)} = \phi$ be a set of boundary data for $\mathbf{c}^{(B)}$.
3. **For** each data point $\mathbf{d} \in \mathbf{c}^{(A)}$ **do**
4. Find all data points $\rho \in \mathbf{c}^{(B)}$ whose $\|\mathbf{d} - \rho\|$ is minimum and put them in $\mathcal{B}^{(B)}$.
5. **EndFor**
6. **For** each data point $\rho \in \mathbf{c}^{(B)}$ **do**
7. Find all data points $\mathbf{d} \in \mathbf{c}^{(A)}$ whose $\|\mathbf{d} - \rho\|$ is minimum and put them in $\mathcal{B}^{(A)}$.
8. **EndFor**

Before the last step, the concept of well-known resampling technique called Bootstrapping was adapted to estimate the natural standard deviation of data distribution of each cluster. Let $\sigma^{(org)}$ and $\sigma^{(nat)}$ be the original standard deviation and the natural standard deviation of training data of a cluster, respectively. A set of synthetic data is generated within the suitable space calculated by using the difference of $\sigma^{(org)}$ and $\sigma^{(nat)}$.

Suppose class *A* is a considered class.

Algorithm 3: *Generating Synthetic Boundary Data*

---

1. Let $\beta_i \in \{-1, 1\}$ be a random sign value for each feature *i*.
2. Let $\alpha_i \in (0, 1]$ be a random constant for each feature *i*.
2. Let $\mathcal{D}^{(A)} = \phi$ be the set new boundary data points.
3. **For** each $\mathbf{d} \in \mathcal{B}^{(A)}$ **do**
4. Generate a new boundary data point $\mathbf{d}'$ such that

$$\mathbf{d}'_i = \mathbf{d}_i + \alpha_i \beta_i |\sigma_i^{(org)} - \sigma_i^{(nat)}|.$$

5. $\mathcal{D}^{(A)} = \mathcal{D}^{(A)} \cup \{\mathbf{d}'\}$.
6. **EndFor**

Let $\mathcal{N}^{(A)}$ and $\mathcal{N}^{(B)}$ be two sets of all identified boundary data points in classes *A* and *B*, respectively.

Suppose η is the dimensions of feature space.

Algorithm 4: *Adjusting Number of Boundary Data*

If $|\mathcal{N}^{(A)}| < |\mathcal{N}^{(B)}|$ **then**
  If $\mathcal{N}^{(B)} < \eta$ **then**
    Use Algorithm *Generating Synthetic Boundary Data* to generate additional data points to $\mathcal{N}^{(B)}$ until $\mathcal{N}^{(B)} \geq \eta$.
  EndIf
  If $|\mathcal{N}^{(A)}| < |\mathcal{N}^{(B)}|$ **then**
    Use Algorithm *Generating Synthetic Boundary Data* to generate additional data points to $\mathcal{N}^{(A)}$ until $|\mathcal{N}^{(A)}|$ equals $|\mathcal{N}^{(B)}|$.
  EndIf
else
  If $\mathcal{N}^{(A)} < \eta$ **then**
    Use Algorithm *Generating Synthetic Boundary Data* to generate additional data points to $\mathcal{N}^{(A)}$ until $\mathcal{N}^{(A)} \geq \eta$.
  EndIf
  If $|\mathcal{N}^{(B)}| < |\mathcal{N}^{(A)}|$ **then**
    Use Algorithm *Generating Synthetic Boundary Data* to generate additional data points to $\mathcal{N}^{(B)}$ until $|\mathcal{N}^{(A)}|$ equals $|\mathcal{N}^{(B)}|$.
  EndIf
EndIf

3.4 Data Preparation

A set of 55 *E.coli* reference pathways were obtained from aMAZE database (Lemer et al., 2004). All metabolites associated with all reactions in this reference pathway set were listed. There are 166 reactions and their associated 215 metabolites. Based on the available 2D chemical structures in LIGAND database from KEGG database (Goto et al., 2002) downloaded in July, 14, 2010, the available 208 metabolites and their involved reactions were used to construct a defined graph $\mathbf{K}$ as described in Sections 3.1 and 3.2.1 and obtained $C_2^m \cdot (m-1)$ metabolite input queries where $m = 208$. So, there are 4,456,296 metabolite input queries which they are too huge. Instead of a big metabolite input query set acquired from a graph $\mathbf{K}$, $\mathbf{K}_1, ..., \mathbf{K}_s$ from s disjoint sets of metabolites $V_1, ..., V_s$ were constructed where s is a number of connected components discovered by constructing a combined graph of all *E.coli* reference pathways. Then, $\sum_{i=1}^{s} C_2^{|V_i|} \cdot (|V_i| - 1)$ metabolite input queries were obtained in order to prepare input feature patterns. The whole 13 connected components were found out from combined 50 *E.coli* reference pathways and obtained each metabolite input query $\mathbf{h}_j \in \mathbf{H}$ where $j = 1, 2, ..., l$ and $l = 44,048$. Note that 5 reference pathways, namely, Phospholipid biosynthesis, Proline degradation, Proto heme and heme O biosynthesis, Pyruvate oxidation pathway including Siroheme biosynthesis, were excluded since some metabolites in such pathways have no 2D structure information.

Transformable or convertible properties were checked for every possible metabolite pair as defined in Section 3.1. In order to detect the transformable property for each metabolite pair (b, t), p possible simple paths with length no more than a certain value k were searched by applying bread-first search graph traversal routine (Cormen et al., 2001) to discover each new path. Later, information about intermediate vertices gained from p possible simple paths was used for assigning 4 binary class targets to their relevant metabolite input queries according to 4 questions in Section 3.2.1.

To prepare each input feature pattern $\mathbf{a}_j \in \mathbf{A}$ as explained in Section 3.2.2, molecular features introduced by (Mu et al., 2011) including their calculation protocols were used as the following. For each metabolite, 81 molecular properties were calculated by using JOELib 2004 (JOELib, 2004), CDK 1.4.6 (Steinbeck et al., 2003) and MOPAC 2009 (Stewart, 2009). Before computing all properties, its optimized 3D chemical structure was computed by using MOPAC 2009 with a PM3 parameter set which the 3D structure originated from a 2D structure with added explicit hydrogen atoms that was prepared by MolConvertor 5.5.1 (Marvin, 2011).

After calculating all properties of all metabolites, all of them were checked that whether some properties of some metabolites were able to be properly computed, if they are $\pm\infty$ or NaN, then each of them is set to a constant distinct value comparing with all distinct values in its property.

3.5 Performance Evaluation

After a balanced binary class training data set $\mathbf{B} = \{(\mathbf{x}_k, \mathbf{c}_k)|k = 1, ..., l_{\mathbf{B}}\}$, prepared by the method in Section 3.3, was trained by the feed-forward neural networks, each class output $\mathbf{o}_i \in \mathbf{O}$ related to its testing data $\tilde{\mathbf{a}}_i$ such that $(\tilde{\mathbf{a}}_i, \mathbf{c}_i) \in \mathbf{T}$ was predicted by these trained models where $i = 1, ..., l_{\mathbf{T}}$. Let $\mathbf{c}_i = 1$ be a positive(minority) class and $\mathbf{c}_i = 0$ be a negative(majority) class. According to confusion matrix, two aspects of the correct prediction between a class output $\mathbf{o}_i$ and its corresponding class target $\mathbf{c}_i$ were evaluated as true positive TP and true negative TN, whereas two aspects of the misprediction were measured as false positive FP and false negative FN. Then, some traditional metrics in Figure 3.4, namely, accuracy, sensitivity (recall), specificity, precision(positive predictive value), F-measure for positive and negative classes, G-means and Matthews correlation coefficient were calculated in order to assess more characteristics of model performance. In addition, area under the curve AUC was also computed by plotting ROC curve where x-axis is 1-specificity and y-axis is sensitivity.

Accuracy: $Acc = \dfrac{TP + TN}{TP + TN + FP + FN}$ (a)

Sensitivity/Recall/TP rate: $TPR = \dfrac{TP}{TP + FN}$ (b)

Specificity/TN rate: $TNR = \dfrac{TN}{TN + FP}$ (c)

Precision/Positive predictive value: $PPV = \dfrac{TP}{TP + FP}$ (d)

Positive class F-measure: $F_\beta^P = (1 + \beta^2)\dfrac{PPV \cdot TPR}{\beta^2 PPV + TPR}$, where $\beta = 1$ (t)

Negative class F-measure: $F_\beta^N = (1 + \beta^2)\dfrac{NPV \cdot TNR}{\beta^2 NPV + TNR}$, where $\beta = 1$ and $NPV = \dfrac{TN}{TN + FN}$ (t)

G-mean: $Gm = \sqrt{TPR \cdot TNR}$ (f)

Matthews correlation coefficient: $MCC = \dfrac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$ (g)

Figure 3.4: Evaluation Metrics.

In comparative study (see Section 4.2), the statistical significance was analyzed by the paired t-test (Walpole et al., 2011) of the paired value sets from two comparing methods such that each set was computed by the same performance evaluation. Each evaluation was calculated from the combined results of all sub-models. The paired t-test is based on assumption

that differences of pair are close to normally distributed. The null hypothesis is a difference of mean between the paired results from two methods equivalent to zero which it is given as $H_0 : \mu_1 - \mu_2 = 0$, while the alternative hypothesis is such mean difference not equivalent to zero which it is written as $H_1 : \mu_1 - \mu_2 \neq 0$. Denote that $\mu_1 - \mu_2$ is the population mean difference. Let $d_i = x_{1i} - x_{2i}$ be a difference of two values calculated by the same performance metric from method 1 and method 2 where i is the i^{th} trial(or the i^{th} fold testing data) and $d_i \in D$. There are $|D| = n$ trials for each considered method. The test statistic is given as follows: $t = \bar{d} \Big/ (s_d \Big/ \sqrt{n})$ where $\bar{d}$ and s_d are the mean and the standard deviation of the difference set D, respectively. Critical regions are found by constructing the t-distribution with $\upsilon = n - 1$ degree of freedom. The test process does not reject H_0 when $-t_{\alpha/2,\upsilon} < t < t_{\alpha/2,\upsilon}$ where the significant level is $\alpha = 0.05$, otherwise, it implies that H_1 is accepted.

CHAPTER IV

EXPERIMENTS AND RESULTS

This section exhibits the experimental results from applying the proposed methodology. First, the ANNs supervised learning model results of the predictive class q vs. non-class q models were shown. Second, performance results of these predictive model output values, both original output scores and adjusted output scores were compared with those of another method in many points such as the different number of the parallel trained sub-models, the different output cut-off values for classifying the input data as class q or non-class q, and various aspects of new unseen data performance results i.e. performance measurement according to each particular sub-model, pathway and compound.

4.1 Training Neural Network Models and Evaluating Model Performance

From the previous section, the four input-target data sets $\mathbf{\Gamma}^{(q)}$ such that $(\mathbf{a}_j, \mathbf{c}_j^{(q)}) \in \mathbf{\Gamma}^{(q)}$, $\mathbf{a}_j \in \mathbf{A}$ and $q = 1,2,3,4$ were ready for the next procedure that produced the balanced training data set. Before handling an imbalanced training data set problem, although the number of metabolite input queries were reduced by preparing them from graph $\mathbf{K}_1,...,\mathbf{K}_s$, the size of a data set, $l = 44,048$, still be not easy to yield an effective feed-forward neural network model with suitable parameters in reasonable time. Therefore, using a key feature as mentioned in Section 3.2.3, $\mathbf{A}$ was finally divided into g disjoint input pattern sub-sets $\mathbf{A}_1,...,\mathbf{A}_i,...\mathbf{A}_g$. On account of model performance evaluation, each of $\mathbf{A}_i$ was randomly partitioned into $k = 3$ disjoint sub-groups according to the k-fold cross validation, $\sum_{k=1}^{3} \mathbf{A}_i^{(k)} = \mathbf{A}_i$, with preserving the proportion of class q and non-class q for $q = 1,2,3,4$ as nearly the same as before dividing them. Later, each sub-data group marked as $\mathbf{A}_i^{(k')}$ became an input pattern part in the testing data sets $\mathbf{T}_i^{(q)}$ such that $\mathbf{a'}_j \in \mathbf{A}_i^{(k')}$ and $(\mathbf{a'}_j, \mathbf{c}_j^{(q)}) \in \mathbf{T}_i^{(q)}$. The rest two sub-data groups were combined as an input part of each pre-training sub-data set $\tau_i^{(q)}$ such that $a_j \in \sum_{k=1,k\neq k'}^{3} \mathbf{A}_i^{(k)}$ and $(\mathbf{a}_j, \mathbf{c}_j^{(q)}) \in \tau_i^{(q)}$. Afterwards, all pre-training sub-data sets $\tau_1^{(q)},...,\tau_i^{(q)},...,\tau_g^{(q)}$ were applied for creating its corresponding balanced training data set $\mathbf{B}_1^{(q)},...,\mathbf{B}_i^{(q)},...,\mathbf{B}_g^{(q)}$ as detailed in Section 3.3. An example of the total 6 sub-data visualization before and after fixing imbalanced data using the proposed method are shown in Figures 4.1 to 4.6.

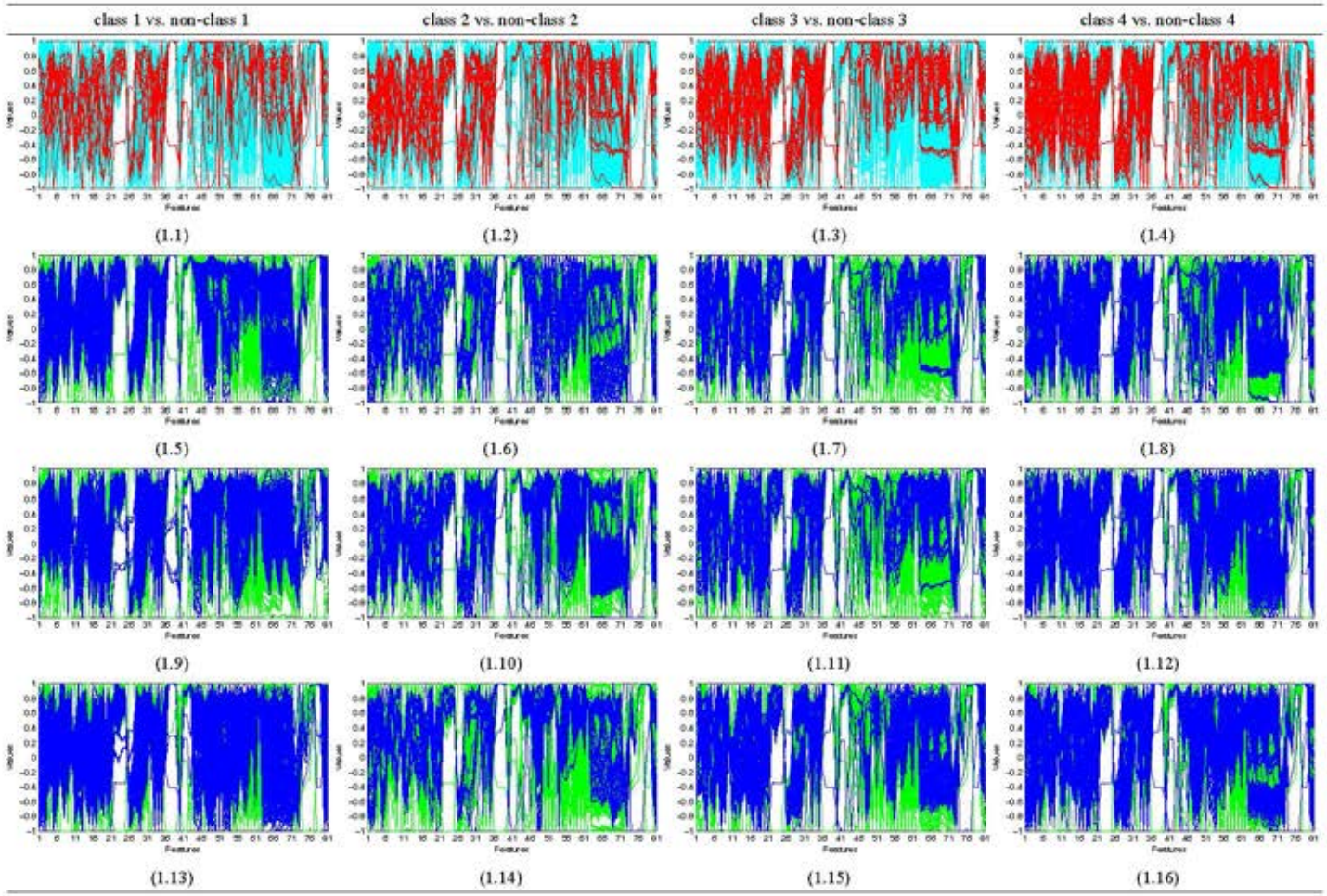


Figure 4.1: Parallel coordinate plots of sub-data 1 in class q vs. non-class q where $q = 1, 2, 3, 4$ ((1.1)-(1.4)). Each line represents each of input feature pattern, while the values in each feature were normalized in $[-1, 1]$. Red and cyan lines belong to the pre-training data with minority class and majority class, respectively. Blue and green lines belong to the nearly balanced training data with minority class and majority class, respectively. The fold 1 ((1.5)-(1.8)), fold 2 ((1.9)-(1.12)), and fold 3 ((1.13)-(1.16)) nearly balanced training data were created by the fold 1, fold 2, and fold 3 imbalanced pre-training data, respectively, using the proposed method.

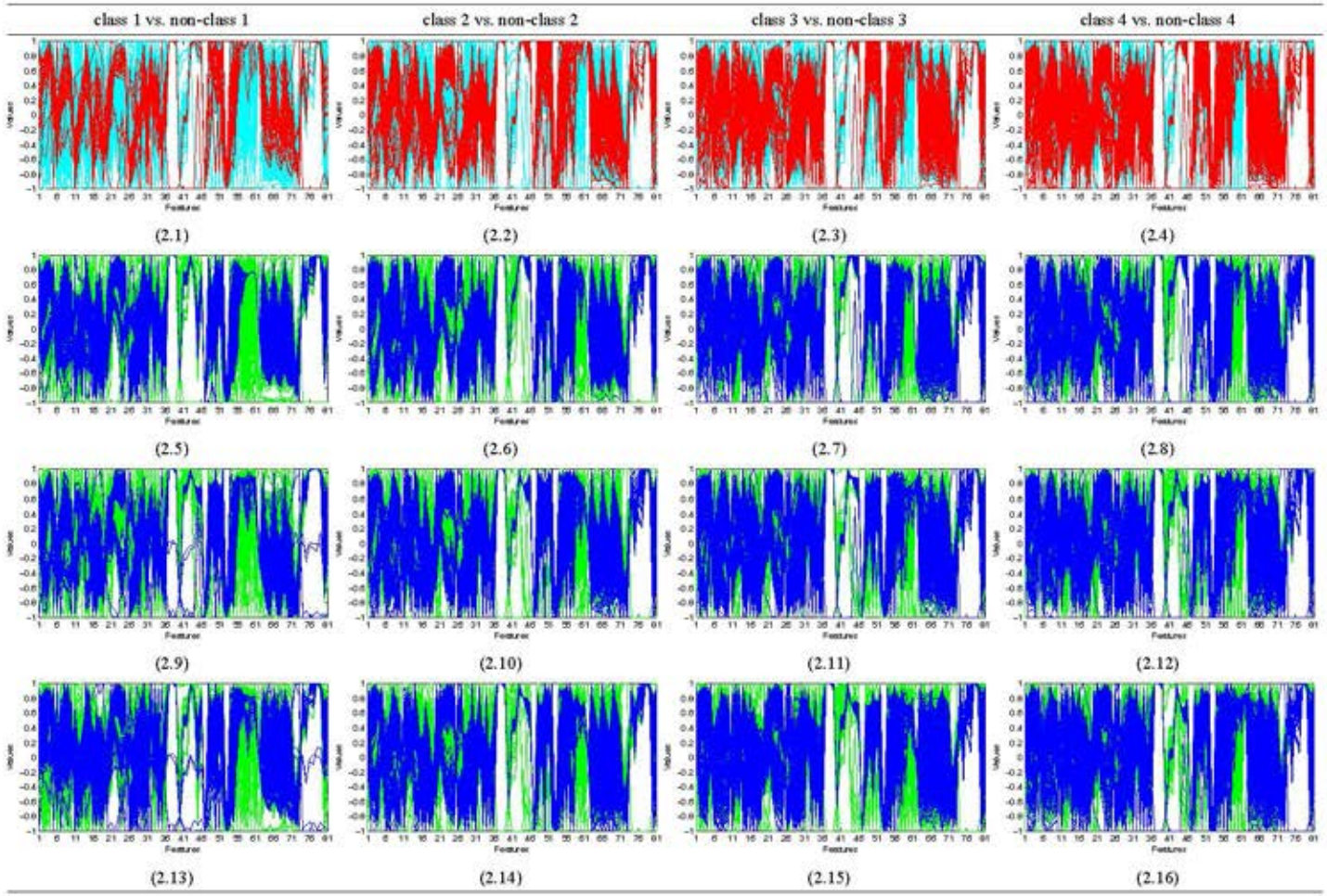


Figure 4.2: Parallel coordinate plots of sub-data 2 in class q vs. non-class q where $q = 1, 2, 3, 4$ ((2.1)-(2.4)). Each line represents each of input feature pattern, while the values in each feature were normalized in $[-1, 1]$. Red and cyan lines belong to the pre-training data with minority class and majority class, respectively. Blue and green lines belong to the nearly balanced training data with minority class and majority class, respectively. The fold 1 ((2.5)-(2.8)), fold 2 ((2.9)-(2.12)), and fold 3 ((2.13)-(2.16)) nearly balanced training data were created by the fold 1, fold 2, and fold 3 imbalanced pre-training data, respectively, using the proposed method.

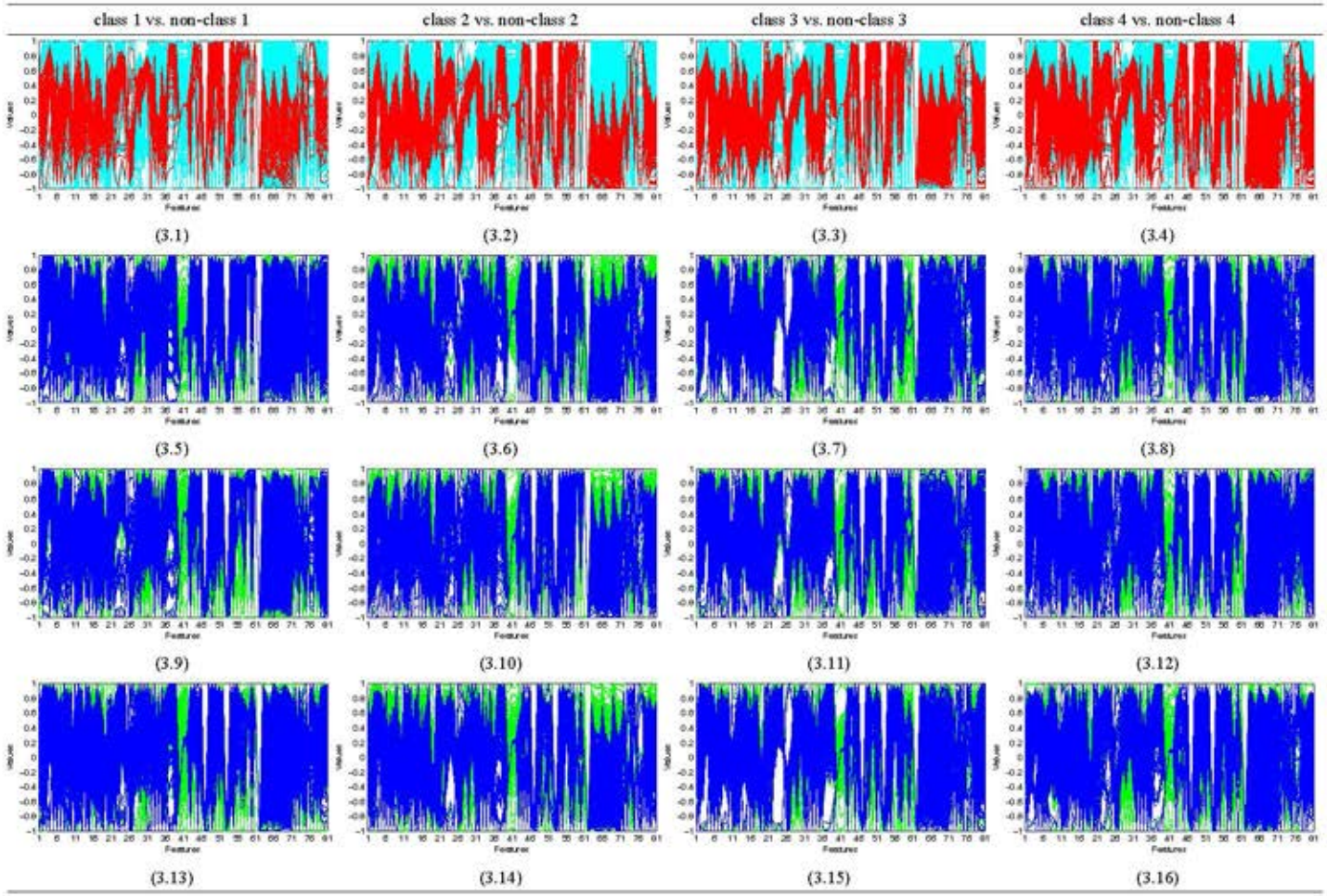


Figure 4.3: Parallel coordinate plots of sub-data 3 in class q vs. non-class q where $q = 1, 2, 3, 4$ ((3.1)-(3.4)). Each line represents each of input feature pattern, while the values in each feature were normalized in $[-1, 1]$. Red and cyan lines belong to the pre-training data with minority class and majority class, respectively. Blue and green lines belong to the nearly balanced training data with minority class and majority class, respectively. The fold 1 ((3.5)-(3.8)), fold 2 ((3.9)-(3.12)), and fold 3 ((3.13)-(3.16)) nearly balanced training data were created by the fold 1, fold 2, and fold 3 imbalanced pre-training data, respectively, using the proposed method.

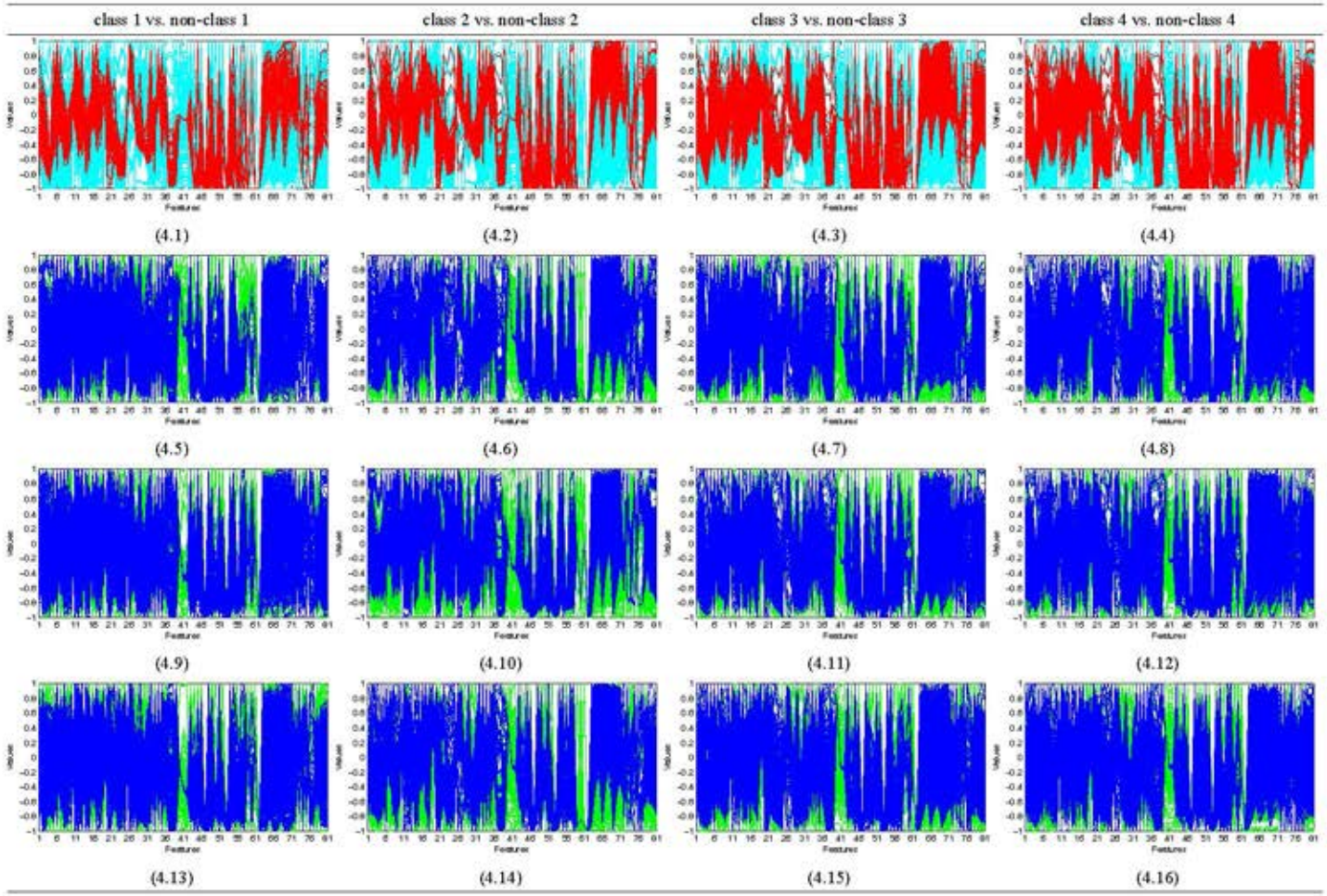


Figure 4.4: Parallel coordinate plots of sub-data 4 in class q vs. non-class q where $q = 1, 2, 3, 4$ ((4.1)-(4.4)). Each line represents each of input feature pattern, while the values in each feature were normalized in $[-1, 1]$. Red and cyan lines belong to the pre-training data with minority class and majority class, respectively. Blue and green lines belong to the nearly balanced training data with minority class and majority class, respectively. The fold 1 ((4.5)-(4.8)), fold 2 ((4.9)-(4.12)), and fold 3 ((4.13)-(4.16)) nearly balanced training data were created by the fold 1, fold 2, and fold 3 imbalanced pre-training data, respectively, using the proposed method.

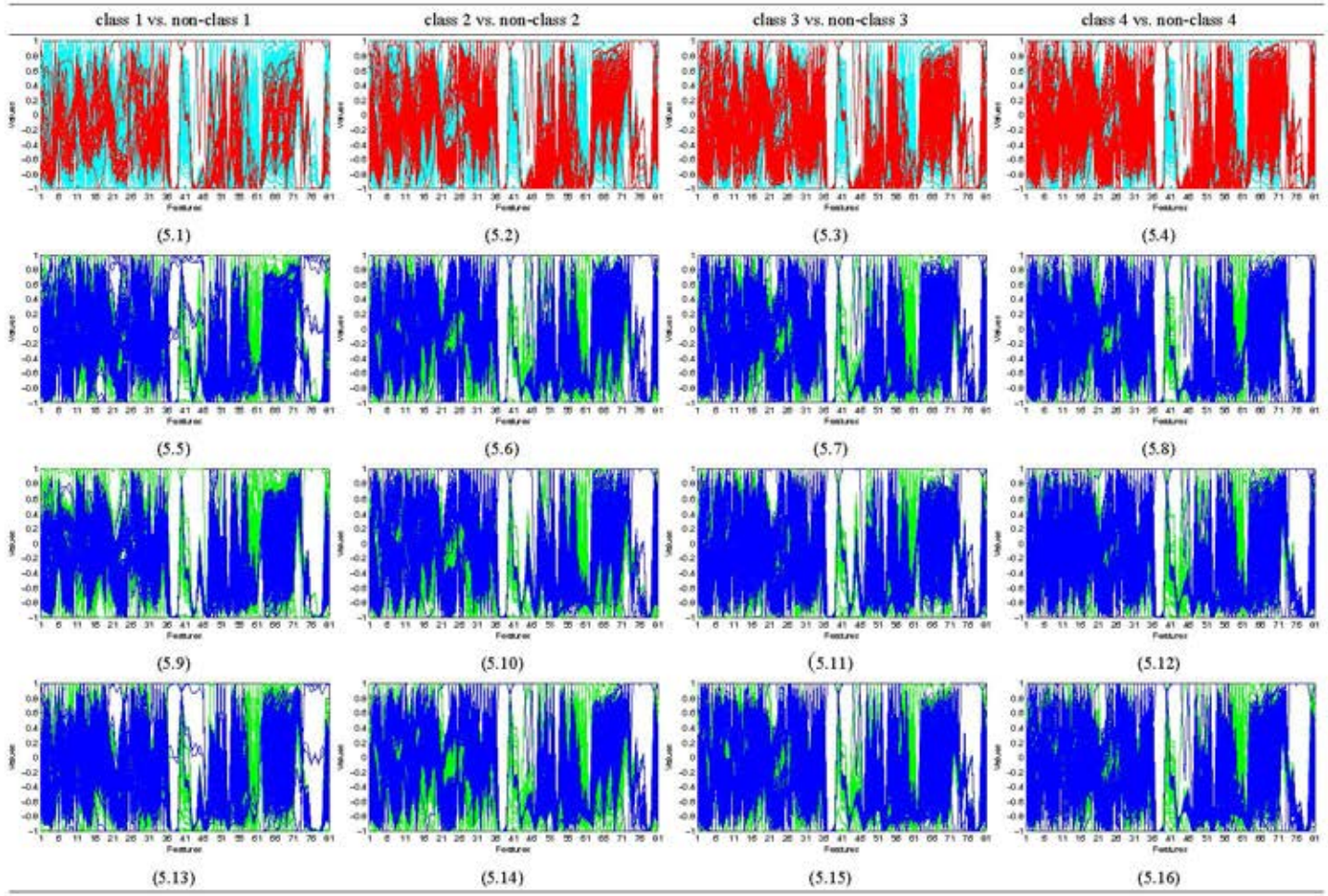


Figure 4.5: Parallel coordinate plots of sub-data 5 in class q vs. non-class q where $q = 1, 2, 3, 4$ ((5.1)-(5.4)). Each line represents each of input feature pattern, while the values in each feature were normalized in $[-1, 1]$. Red and cyan lines belong to the pre-training data with minority class and majority class, respectively. Blue and green lines belong to the nearly balanced training data with minority class and majority class, respectively. The fold 1 ((5.5)-(5.8)), fold 2 ((5.9)-(5.12)), and fold 3 ((5.13)-(5.16)) nearly balanced training data were created by the fold 1, fold 2, and fold 3 imbalanced pre-training data, respectively, using the proposed method.

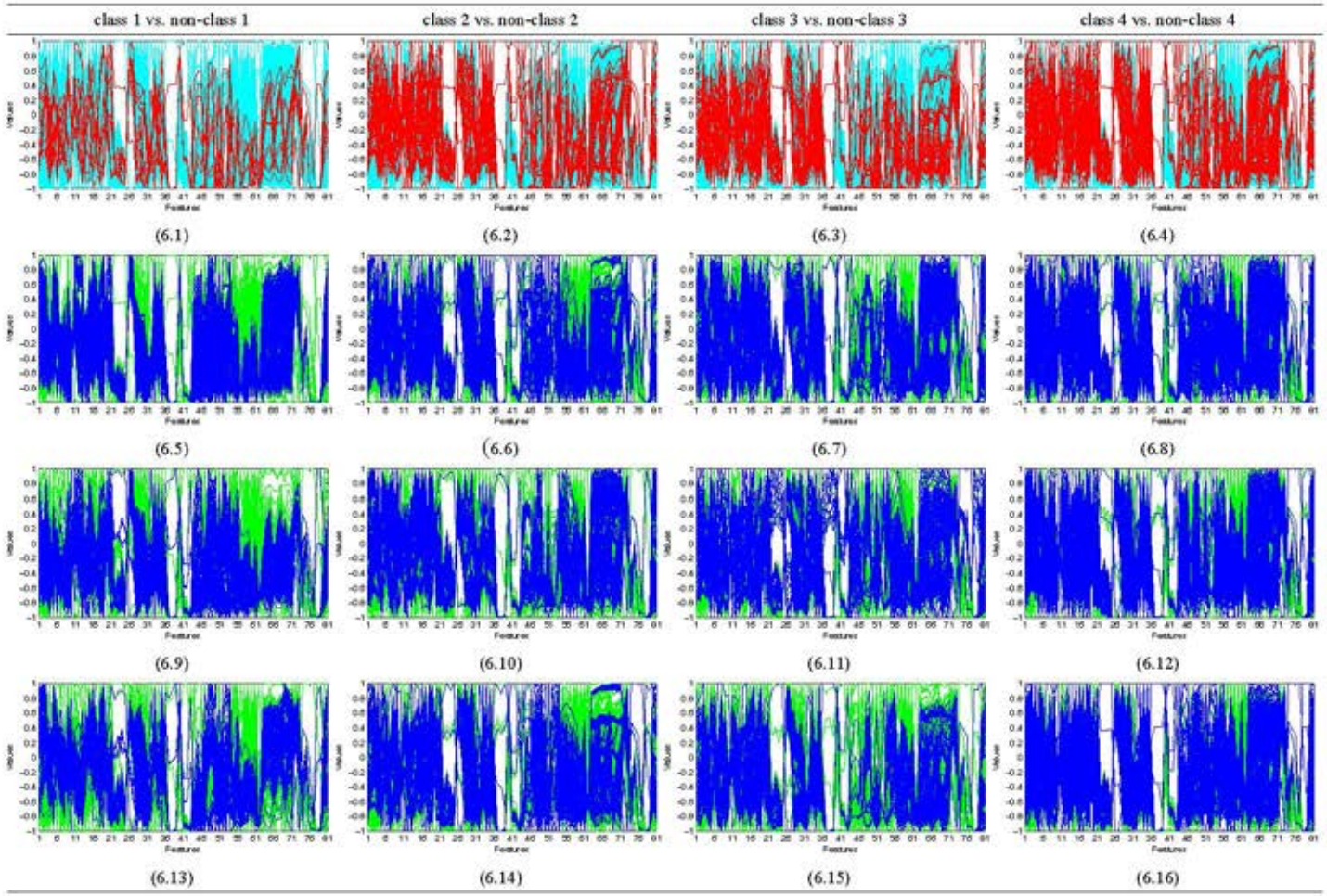


Figure 4.6: Parallel coordinate plots of sub-data 6 in class q vs. non-class q where $q = 1, 2, 3, 4$ ((6.1)-(6.4)). Each line represents each of input feature pattern, while the values in each feature were normalized in $[-1, 1]$. Red and cyan lines belong to the pre-training data with minority class and majority class, respectively. Blue and green lines belong to the nearly balanced training data with minority class and majority class, respectively. The fold 1 ((6.5)-(6.8)), fold 2 ((6.9)-(6.12)), and fold 3 ((6.13)-(6.16)) nearly balanced training data were created by the fold 1, fold 2, and fold 3 imbalanced pre-training data, respectively, using the proposed method.

Any of supervised learning methods can take $\mathbf{B}_i^{(q)}$ to train a sub-model $\mathbf{M}_{(i,q)}$ for $i = 1,2,...,g$ and $q = 1,2,3,4$. In this work, the feed-forward neural network with scaled conjugate gradient technique was used to build each predictive sub-model $\mathbf{M}_{(i,q)}$. The activation function of both a hidden layer and an output layer is sigmoid function. Conventional parameters not only in the feed-forward neural network training procedure but also in Kohonen′s self-organizing map(SOM) procedure, for instance, a number of epochs and a learning rate etc., were identically set for every pre-training sub-data set $\tau_i^{(q)}$ and their corresponding sub-model $\mathbf{M}_{(i,q)}$. However, there are some concerned factors that considerably impact performance of each sub-model, $\mathbf{M}_{(i,q)}$. One is numbers of neurons, (x_-,y_-) and (x_+,y_+), in 2D SOM method (Section 3.3.1) for clustering majority(negative) and minority(positive) class data points, respectively. Another is the number of neurons, *s*, in a hidden layer of each feed-forward neural network. Moreover, a neighbor threshold, *k*, is varied which may help improving performance of sub-models (Section 3.3).

There are no clues how to select these three concerned parameters. Based on above sub-data splitting methods as well as the following scheme of training and selecting predictive model, the $g \cdot |q|$ sub-models were parallel trained. Then, each sub-model with suitable concerned parameter values was picked out as the predictive sub-model $\mathbf{M}_{(i,q)}$ that yielded acceptable (maybe not optimum) performances in practical time. Initially, for each pre-training sub-data set $\tau_i^{(q)}$, a neighbor threshold *k* was set as 0, the numbers of neurons in 2D self-organizing map both (x_-,y_-) and (x_+,y_+) were both set as $(10,10)$ for the big size of $\tau_i^{(q)}$ or $(7,7)$ for the small-medium size of $\tau_i^{(q)}$. Note that the size of each $\tau_i^{(q)}$ was about $2,600-16,000$ input patterns. The feed-forward neural network models were trained with various number of hidden neurons *s* as following: $s = 1,2,4,8,16,32$ and 64. If there exists a trained model with *s* yielding all performance values being greater than 0.75 for all 3 different training and testing sub-data sets, then, the process of seeking the appropriate parameters, (x_-,y_-,x_+,y_+,k,s), was stopped, else the values of *k*, (x_-,y_-) and (x_+,y_+) were simultaneously selected guided by prior performance results hoping that they would produce the better performance than the previous round of selecting parameter values. The new values of *k*, (x_-,y_-) and (x_+,y_+) were varied for no more than 5 rounds due to the practicable time of performing the whole experiments in this work. In the case of no appropriate parameter values, one trained model with *s* on a round that returned the better performance values among all of them was selected. These metrics (Section 3.5) for selecting sub-models are as the following:

1) Accuracy (*Acc*) measures the whole correctness which is the fraction of the truly pre-

dicted minority and majority class samples in all predicted samples,

2) True positive rate (TPR) measures the minority class sample correctness which is the fraction of the truly predicted minority class samples in all minority class samples,

3) True negative rate (TNR) calculates in the same way as true positive rate, but it measures the majority class sample correctness,

4) Gm is Geometric mean of TPR and TNR, and

5) Area under the ROC curve (AUC) represents a single value from ROC graph to show performance of each classifier which if it is more than 0.5, then the model is better than randomly guessing.

Table 4.1: Acc, TPR, TNR and Gm performances at an output cut-off value = 0.5 including AUC performance of the $g = 6$ sub-models with selected parameter values from the $3-$ fold cross validation for separating class 1 and non-class 1.

	Sub-data	Acc	TPR	TNR	Gm	AUC
Pre-training data set	1	0.950 ± 0.012	1.000 ± 0.000	0.950 ± 0.012	0.975 ± 0.006	0.978 ± 0.004
	2	0.857 ± 0.060	0.959 ± 0.001	0.856 ± 0.061	0.906 ± 0.032	0.910 ± 0.043
	3	0.780 ± 0.024	0.839 ± 0.048	0.779 ± 0.025	0.808 ± 0.010	0.854 ± 0.021
	4	0.735 ± 0.041	0.885 ± 0.015	0.732 ± 0.042	0.804 ± 0.022	0.808 ± 0.018
	5	0.931 ± 0.025	0.986 ± 0.025	0.931 ± 0.025	0.958 ± 0.025	0.955 ± 0.025
	6	0.945 ± 0.041	0.967 ± 0.058	0.945 ± 0.041	0.955 ± 0.025	0.975 ± 0.018
	Overall	0.888 ± 0.015	0.902 ± 0.017	0.887 ± 0.015	0.894 ± 0.003	0.912 ± 0.022
Nearly balanced training data set	1	0.978 ± 0.004	1.000 ± 0.000	0.953 ± 0.006	0.976 ± 0.003	0.975 ± 0.004
	2	0.947 ± 0.047	0.987 ± 0.015	0.908 ± 0.078	0.946 ± 0.047	0.944 ± 0.048
	3	0.880 ± 0.019	0.943 ± 0.024	0.809 ± 0.048	0.873 ± 0.022	0.903 ± 0.028
	4	0.855 ± 0.028	0.961 ± 0.023	0.732 ± 0.037	0.839 ± 0.031	0.854 ± 0.021
	5	0.979 ± 0.031	0.989 ± 0.019	0.969 ± 0.044	0.979 ± 0.031	0.977 ± 0.034
	6	0.988 ± 0.006	0.999 ± 0.001	0.974 ± 0.014	0.986 ± 0.007	0.983 ± 0.008
	Overall	0.910 ± 0.010	0.967 ± 0.012	0.848 ± 0.025	0.905 ± 0.011	0.937 ± 0.019
Testing data set	1	0.948 ± 0.016	0.767 ± 0.088	0.949 ± 0.016	0.852 ± 0.043	0.861 ± 0.038
	2	0.846 ± 0.042	0.489 ± 0.100	0.851 ± 0.044	0.642 ± 0.051	0.719 ± 0.048
	3	0.767 ± 0.030	0.604 ± 0.072	0.770 ± 0.030	0.681 ± 0.044	0.708 ± 0.068
	4	0.728 ± 0.031	0.672 ± 0.099	0.729 ± 0.034	0.698 ± 0.037	0.720 ± 0.055
	5	0.923 ± 0.021	0.689 ± 0.119	0.925 ± 0.022	0.796 ± 0.064	0.797 ± 0.053
	6	0.945 ± 0.041	0.633 ± 0.153	0.945 ± 0.042	0.770 ± 0.098	0.836 ± 0.037
	Overall	0.883 ± 0.013	0.633 ± 0.051	0.884 ± 0.013	0.748 ± 0.036	0.797 ± 0.065

Table 4.2: *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the 3− fold cross validation for separating class 2 and non-class 2.

	Sub-data	*Acc*	*TPR*	*TNR*	*Gm*	*AUC*
Pre-training data set	1	0.947 ± 0.015	1.000 ± 0.000	0.947 ± 0.015	0.973 ± 0.008	0.975 ± 0.006
	2	0.917 ± 0.035	0.966 ± 0.006	0.912 ± 0.039	0.938 ± 0.022	0.937 ± 0.028
	3	0.878 ± 0.028	0.939 ± 0.038	0.875 ± 0.028	0.906 ± 0.032	0.935 ± 0.014
	4	0.926 ± 0.036	0.972 ± 0.010	0.923 ± 0.038	0.947 ± 0.023	0.953 ± 0.020
	5	0.919 ± 0.028	0.950 ± 0.028	0.917 ± 0.031	0.933 ± 0.009	0.943 ± 0.009
	6	0.948 ± 0.025	0.995 ± 0.009	0.948 ± 0.025	0.971 ± 0.013	0.979 ± 0.010
	Overall	0.929 ± 0.011	0.959 ± 0.007	0.928 ± 0.011	0.944 ± 0.005	0.954 ± 0.006
Nearly balanced training data set	1	0.980 ± 0.016	1.000 ± 0.000	0.959 ± 0.034	0.979 ± 0.017	0.978 ± 0.015
	2	0.975 ± 0.018	0.993 ± 0.000	0.954 ± 0.040	0.973 ± 0.021	0.967 ± 0.022
	3	0.924 ± 0.048	0.965 ± 0.032	0.876 ± 0.062	0.920 ± 0.048	0.934 ± 0.026
	4	0.962 ± 0.027	0.987 ± 0.011	0.932 ± 0.046	0.959 ± 0.029	0.959 ± 0.024
	5	0.922 ± 0.021	0.982 ± 0.007	0.857 ± 0.050	0.917 ± 0.024	0.919 ± 0.023
	6	0.959 ± 0.020	0.996 ± 0.004	0.922 ± 0.036	0.958 ± 0.021	0.958 ± 0.020
	Overall	0.948 ± 0.011	0.982 ± 0.007	0.908 ± 0.015	0.945 ± 0.011	0.958 ± 0.005
Testing data set	1	0.939 ± 0.014	0.821 ± 0.044	0.939 ± 0.014	0.878 ± 0.020	0.927 ± 0.018
	2	0.853 ± 0.035	0.714 ± 0.043	0.869 ± 0.043	0.787 ± 0.010	0.840 ± 0.006
	3	0.856 ± 0.021	0.774 ± 0.042	0.860 ± 0.024	0.816 ± 0.013	0.888 ± 0.019
	4	0.896 ± 0.022	0.737 ± 0.017	0.905 ± 0.023	0.816 ± 0.007	0.875 ± 0.035
	5	0.912 ± 0.014	0.900 ± 0.028	0.913 ± 0.016	0.906 ± 0.008	0.919 ± 0.005
	6	0.946 ± 0.022	0.892 ± 0.099	0.946 ± 0.022	0.918 ± 0.042	0.957 ± 0.013
	Overall	0.915 ± 0.008	0.785 ± 0.010	0.919 ± 0.008	0.849 ± 0.004	0.898 ± 0.011

Table 4.3: *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the 3− fold cross validation for separating class 3 and non-class 3.

	Sub-data	*Acc*	*TPR*	*TNR*	*Gm*	*AUC*
Pre-training data set	1	0.968 ± 0.008	0.989 ± 0.010	0.968 ± 0.008	0.978 ± 0.007	0.984 ± 0.007
	2	0.947 ± 0.002	0.947 ± 0.037	0.947 ± 0.008	0.947 ± 0.014	0.957 ± 0.009
	3	0.920 ± 0.024	0.935 ± 0.016	0.918 ± 0.025	0.927 ± 0.021	0.933 ± 0.021
	4	0.911 ± 0.037	0.945 ± 0.033	0.906 ± 0.038	0.925 ± 0.034	0.936 ± 0.026
	5	0.942 ± 0.003	0.982 ± 0.008	0.939 ± 0.003	0.960 ± 0.005	0.966 ± 0.002
	6	0.970 ± 0.005	0.983 ± 0.015	0.970 ± 0.005	0.976 ± 0.008	0.984 ± 0.003
	Overall	0.950 ± 0.007	0.951 ± 0.017	0.950 ± 0.007	0.950 ± 0.012	0.962 ± 0.009
Nearly balanced training data set	1	0.973 ± 0.014	0.999 ± 0.002	0.946 ± 0.030	0.972 ± 0.016	0.971 ± 0.018
	2	0.985 ± 0.007	0.987 ± 0.011	0.983 ± 0.005	0.985 ± 0.007	0.982 ± 0.007
	3	0.968 ± 0.013	0.978 ± 0.007	0.956 ± 0.021	0.967 ± 0.014	0.965 ± 0.015
	4	0.967 ± 0.021	0.982 ± 0.012	0.951 ± 0.030	0.966 ± 0.021	0.966 ± 0.020
	5	0.979 ± 0.012	0.995 ± 0.005	0.961 ± 0.020	0.978 ± 0.013	0.977 ± 0.010
	6	0.969 ± 0.011	0.997 ± 0.004	0.937 ± 0.022	0.967 ± 0.012	0.966 ± 0.014
	Overall	0.971 ± 0.012	0.984 ± 0.007	0.957 ± 0.018	0.970 ± 0.012	0.970 ± 0.012
Testing data set	1	0.963 ± 0.007	0.901 ± 0.067	0.964 ± 0.008	0.932 ± 0.031	0.973 ± 0.010
	2	0.869 ± 0.013	0.707 ± 0.066	0.898 ± 0.011	0.796 ± 0.038	0.888 ± 0.031
	3	0.877 ± 0.014	0.804 ± 0.018	0.886 ± 0.016	0.844 ± 0.012	0.898 ± 0.014
	4	0.842 ± 0.036	0.725 ± 0.045	0.859 ± 0.040	0.789 ± 0.032	0.858 ± 0.020
	5	0.907 ± 0.015	0.809 ± 0.017	0.914 ± 0.016	0.860 ± 0.014	0.913 ± 0.018
	6	0.968 ± 0.001	0.899 ± 0.035	0.968 ± 0.001	0.933 ± 0.018	0.961 ± 0.021
	Overall	0.925 ± 0.005	0.777 ± 0.006	0.934 ± 0.005	0.852 ± 0.005	0.922 ± 0.005

Table 4.4: *Acc*, *TPR*, *TNR* and *Gm* performances at an output cut-off value = 0.5 including *AUC* performance of the $g = 6$ sub-models with selected parameter values from the $3-$ fold cross validation for separating class 4 and non-class 4.

	Sub-data	*Acc*	*TPR*	*TNR*	*Gm*	*AUC*
Pre-training data set	1	0.964 ± 0.006	0.990 ± 0.010	0.963 ± 0.005	0.976 ± 0.008	0.983 ± 0.000
	2	0.963 ± 0.007	0.959 ± 0.004	0.965 ± 0.009	0.962 ± 0.006	0.978 ± 0.011
	3	0.973 ± 0.004	0.985 ± 0.001	0.970 ± 0.005	0.978 ± 0.002	0.986 ± 0.002
	4	0.974 ± 0.009	0.990 ± 0.002	0.970 ± 0.011	0.980 ± 0.006	0.988 ± 0.006
	5	0.975 ± 0.009	0.986 ± 0.003	0.974 ± 0.010	0.980 ± 0.006	0.990 ± 0.004
	6	0.966 ± 0.021	0.996 ± 0.003	0.965 ± 0.022	0.980 ± 0.012	0.983 ± 0.009
	Overall	0.968 ± 0.006	0.983 ± 0.001	0.967 ± 0.007	0.975 ± 0.003	0.985 ± 0.002
Nearly balanced training data set	1	0.965 ± 0.006	0.995 ± 0.007	0.933 ± 0.011	0.963 ± 0.006	0.962 ± 0.010
	2	0.985 ± 0.008	0.989 ± 0.002	0.981 ± 0.014	0.985 ± 0.008	0.984 ± 0.009
	3	0.995 ± 0.003	0.995 ± 0.002	0.995 ± 0.004	0.995 ± 0.003	0.994 ± 0.003
	4	0.996 ± 0.001	0.998 ± 0.001	0.994 ± 0.002	0.996 ± 0.001	0.995 ± 0.002
	5	0.996 ± 0.003	0.999 ± 0.001	0.992 ± 0.006	0.995 ± 0.003	0.994 ± 0.004
	6	0.987 ± 0.004	0.998 ± 0.002	0.974 ± 0.007	0.986 ± 0.004	0.984 ± 0.004
	Overall	0.992 ± 0.002	0.996 ± 0.001	0.987 ± 0.004	0.991 ± 0.002	0.991 ± 0.003
Testing data set	1	0.960 ± 0.006	0.925 ± 0.059	0.961 ± 0.007	0.942 ± 0.028	0.974 ± 0.012
	2	0.931 ± 0.004	0.909 ± 0.012	0.940 ± 0.001	0.924 ± 0.007	0.955 ± 0.006
	3	0.928 ± 0.002	0.899 ± 0.011	0.934 ± 0.005	0.916 ± 0.004	0.954 ± 0.007
	4	0.916 ± 0.016	0.870 ± 0.036	0.927 ± 0.012	0.898 ± 0.023	0.947 ± 0.021
	5	0.949 ± 0.009	0.912 ± 0.003	0.954 ± 0.010	0.933 ± 0.006	0.966 ± 0.007
	6	0.962 ± 0.016	0.930 ± 0.020	0.963 ± 0.016	0.946 ± 0.015	0.973 ± 0.011
	Overall	0.948 ± 0.006	0.898 ± 0.002	0.952 ± 0.006	0.925 ± 0.004	0.961 ± 0.001

In each trial of different training and testing sub-data sets, the g selected sub-models $\mathbf{M}_{(1,q)},...,\mathbf{M}_{(i,q)},...,\mathbf{M}_{(g,q)}$ for class q were combined. Since a sigmoid function of an output layer yields each output value in the range from 0 to 1, the selected cut-off value was 0.5 for categorizing output values into class q vs. non-class q. Each performance evaluation result was averaged as presented by Tables 4.1 to 4.4. The selected class q vs. non-class q predictive sub-models for $q = 1,2,3,4$ show the satisfied average performance values including its standard deviation from 3 trails in almost sub-data as well as the overall values.

Considering Tables 4.1 to 4.4, every overall performance value of the nearly balanced data sets is better than such overall values of the testing data set, while almost each of overall performance values of the pre-training data sets still be competitive with each of such values of the nearly balanced data sets. Each nearly balanced data set is composed of the border data from its pre-training data set and also the new synthetic data according to the proposed approaches. The calculated molecular properties including the pre-process for reducing and transforming the input features provide the pre-training data set with their class targets. The methods produce the nearly balanced data set from the pre-training data set which is not difficult to learn by each neural network sub-model with adequate parameter values.

Generally, with the appropriate parameter values, the $g = 6$ nearly balanced sub-data sets are quite suitable for training neural network models as predictive models for the defined questions 1 to 4. When the performance values of unseen data like the testing sub-data sets were determined, the combined model built for the broadest question i.e. class 4 vs. non-class 4, yielded the superior performance values. Moreover, its class distribution is slightly imbalanced data situation (Table 4.9). This question is about whether input compounds are not roughly related in some routes unlike the kinds of class $1, 2$ or 3. In sequence, the models of more specific questions like class 3 vs. non-class 3 and class 2 vs. non-class 2 produced the good performance values (Tables 4.8 and 4.7). Meanwhile, the models built for the most specific question which asking about whether one/two step relation of transformable compounds such as class 1 vs. non-class 1 yielded the less effective performance values among four defined question models. Also, its class distribution is the most critically imbalanced data situation (Table 4.6).

The chosen parameter sets along with each binary class data ratios of the pre-training and the testing sub-data sets including the nearly balanced training sub-data sets are shown by Tables 4.6 to 4.9.

In each sub-data set for each question in sub-model building tasks, the more the seriously imbalanced binary class sub-data sets exist, the harder the imbalanced data handling is. Moreover, the effective models were yielded by the suitable parameter values. Based on the four pairs of Tables, namely, Tables 4.6 and 4.1, Tables 4.7 and 4.2, Tables 4.8 and 4.3 and Tables 4.9 and 4.4, the class q vs. non-class q ratio of each sub-data set can be observed before/after making them to each corresponding nearly balanced sub-data set along with their performance values. Additionally, when setting adequate numbers of SOM neurons for coarsely clustering data in practical time as well as starting with no need to apply generating more minority data process, the first initial assigned parameters x_-, y_-, x_+, y_+ and k gave the satisfied results.

Furthermore, the models using original imbalanced data before becoming the nearly balanced data were also trained by the proposed processes. The g selected models corresponding to g sub-space for each binary class were combined in each fold data set. Each performance evaluation result from all 3-fold data sets, both training and testing data, was averaged and presented by Table 4.5 including its standard deviation. There were two predictive sub-model types for each question model. One was built by the imbalanced training data, τ, whereas another was

Table 4.5: *Acc*, *TPR*, *TNR* and *Gm* performance at output cut-off = 0.5 including *AUC* performance of the $g = 6$ combined sub-models with selected parameter values from the 3−fold cross validation for separating each model output set into binary classes.

Model		*Acc*	*TPR*	*TNR*	*Gm*	*AUC*
1	τ_o	**0.997 ± 0.000**	0.589 ± 0.064	**1.000 ± 0.000**	0.766 ± 0.043	0.797 ± 0.039
	τ_n	0.888 ± 0.015	0.902 ± 0.017	0.887 ± 0.015	0.894 ± 0.003	0.912 ± 0.022
	$\mathbf{B_n}$	0.910 ± 0.010	**0.967 ± 0.012**	0.848 ± 0.025	**0.905 ± 0.011**	**0.937 ± 0.019**
	$\mathbf{T_o}$	**0.991 ± 0.000**	0.170 ± 0.048	**0.997 ± 0.001**	0.409 ± 0.059	0.636 ± 0.022
	$\mathbf{T_n}$	0.883 ± 0.013	**0.633 ± 0.051**	0.884 ± 0.013	**0.748 ± 0.036**	**0.797 ± 0.065**
2	τ_o	**0.993 ± 0.003**	0.813 ± 0.074	**0.999 ± 0.001**	0.900 ± 0.041	0.932 ± 0.012
	τ_n	0.929 ± 0.011	0.959 ± 0.007	0.928 ± 0.011	0.944 ± 0.005	0.954 ± 0.006
	$\mathbf{B_n}$	0.948 ± 0.011	**0.982 ± 0.007**	0.908 ± 0.015	**0.945 ± 0.011**	**0.958 ± 0.005**
	$\mathbf{T_o}$	**0.972 ± 0.001**	0.484 ± 0.031	**0.987 ± 0.001**	0.691 ± 0.022	0.869 ± 0.015
	$\mathbf{T_n}$	0.915 ± 0.008	**0.785 ± 0.010**	0.919 ± 0.008	**0.849 ± 0.004**	**0.898 ± 0.011**
3	τ_o	**0.992 ± 0.001**	0.883 ± 0.022	**0.999 ± 0.000**	0.939 ± 0.012	0.947 ± 0.009
	τ_n	0.950 ± 0.007	0.951 ± 0.017	0.950 ± 0.007	0.950 ± 0.012	0.962 ± 0.009
	$\mathbf{B_n}$	0.971 ± 0.012	**0.984 ± 0.007**	0.957 ± 0.018	**0.970 ± 0.012**	**0.970 ± 0.012**
	$\mathbf{T_o}$	**0.969 ± 0.001**	0.692 ± 0.017	**0.985 ± 0.000**	0.825 ± 0.010	0.915 ± 0.006
	$\mathbf{T_n}$	0.925 ± 0.005	**0.777 ± 0.006**	0.934 ± 0.005	**0.852 ± 0.005**	**0.922 ± 0.005**
4	τ_o	**0.994 ± 0.001**	0.940 ± 0.011	**0.999 ± 0.001**	0.969 ± 0.005	0.971 ± 0.010
	τ_n	0.968 ± 0.006	0.983 ± 0.001	0.967 ± 0.007	0.975 ± 0.003	0.985 ± 0.002
	$\mathbf{B_n}$	0.992 ± 0.002	**0.996 ± 0.001**	0.987 ± 0.004	**0.991 ± 0.002**	**0.991 ± 0.003**
	$\mathbf{T_o}$	**0.982 ± 0.001**	0.883 ± 0.014	**0.992 ± 0.001**	**0.936 ± 0.007**	**0.964 ± 0.012**
	$\mathbf{T_n}$	0.948 ± 0.006	**0.898 ± 0.002**	0.952 ± 0.006	0.925 ± 0.004	0.961 ± 0.001

Two predictive sub-model types denoted by subscript *o* and *n* which were trained by the imbalanced training data,τ, and the nearly balanced training data set, **B**, respectively. τ, **B** and the unseen testing data, **T** , were applied to both types of selected sub-models to measure the performance. Each bold performance value of each binary class is the highest one comparing among the training data or the testing data.

constructed by the nearly balanced training data, **B**. The output values were in the range from 0 to 1 due to a sigmoid function, so the selected cut-off value was 0.5 for categorizing output values into two classes for each question model. Considering Table 4.5, generally, the combined sub-models trained by **B** yielded the clearly improved *TPR*, *Gm* and *AUC* values of the unseen testing data, **T**. The high *Acc* and *TNR* values including the low *TPR* values reflect misrepresentation of separating function trained by the imbalanced data. Except in the question 4 combined sub-models, their performance values of **T** look slightly different because of the least imbalanced ratio among 4 defined question models. Only sub-space training seem to be enough to yield effective sub-models.

Table 4.6: The selected parameter values and distribution ratios of class 1 and non-class 1.

Sub-data	i^{th}fold	2D SOM neurons		Clusters		k	FFNN hidden units	Pre-training data set		Testing data set		Nearly balanced training data set	
		(x_-,y_-)	(x_+,y_+)	c_-	c_+			Total	Ratio of class 1 to non-class 1	Total	Ratio of class 1 to non-class 1	Total	Ratio of class 1 to non-class 1
	1			60	41		1	5,485	0.20 : 99.80	2,743	0.22 : 99.78	1,856	52.05 : 47.95
1	2	(10,10)	(10,10)	68	39	10	1	5,486	0.22 : 99.78	2,742	0.18 : 99.82	2,276	51.32 : 48.68
	3			63	36		1	5,485	0.20 : 99.80	2,743	0.22 : 99.78	2,514	56.96 : 43.04
	1			22	36		1	1,739	1.44 : 98.56	870	1.38 : 98.62	2,588	47.30 : 52.70
2	2	(7,7)	(7,7)	31	21	10	1	1,740	1.44 : 98.56	869	1.38 : 98.62	1,514	52.05 : 47.95
	3			23	27		8	1,739	1.38 : 98.62	870	1.49 : 98.51	1,903	46.24 : 53.76
	1			46	42		1	4,858	1.32 : 98.68	2,428	1.32 : 98.68	6,053	54.57 : 45.43
3	2	(7,7)	(7,7)	48	35	10	1	4,857	1.32 : 98.68	2,429	1.32 : 98.68	5,429	52.22 : 47.78
	3			47	43		2	4,857	1.32 : 98.68	2,429	1.32 : 98.68	6,557	52.11 : 47.89
	1			49	32		1	3,461	2.02 : 97.98	1,729	1.97 : 98.03	4,210	54.44 : 45.56
4	2	(7,7)	(7,7)	49	29	3	1	3,460	1.99 : 98.01	1,730	2.02 : 97.98	4,414	51.93 : 48.07
	3			49	30		1	3,459	1.99 : 98.01	1,731	2.02 : 97.98	4,202	55.02 : 44.98
	1			35	21		2	2,948	0.78 : 99.22	1,473	0.81 : 99.19	1,592	51.63 : 48.37
5	2	(7,7)	(7,7)	36	20	10	1	2,946	0.78 : 99.22	1,475	0.81 : 99.19	1,574	51.97 : 48.03
	3			24	24		64	2,948	0.81 : 99.19	1,473	0.75 : 99.25	1,787	42.08 : 57.92
	1			27	16		1	10,878	0.10 : 99.90	5,436	0.09 : 99.91	1,206	56.38 : 43.62
6	2	(7,7)	(7,7)	26	19	5	16	10,876	0.10 : 99.90	5,438	0.09 : 99.91	1,090	51.74 : 48.26
	3			31	19		2	10,874	0.09 : 99.91	5,440	0.11 : 99.89	1,004	52.59 : 47.41

Table 4.7: The selected parameter values and distribution ratios of class 2 and non-class 2.

Sub-data	i^{th}fold	2D SOM neurons		Clusters		k	FFNN hidden units	Pre-training data set		Testing data set		Nearly balanced training data set	
		(x_-,y_-)	(x_+,y_+)	c_-	c_+			Total	Ratio of class 2 to non-class 2	Total	Ratio of class 2 to non-class 2	Total	Ratio of class 2 to non-class 2
	1			71	16		2	5,485	0.47 : 99.53	2,743	0.47 : 99.53	1,657	50.27 : 49.73
1	2	(10,10)	(10,10)	70	16	0	4	5,486	0.47 : 99.53	2,742	0.47 : 99.53	1,688	50.59 : 49.41
	3			65	15		2	5,485	0.47 : 99.53	2,743	0.47 : 99.53	1,418	51.34 : 48.66
	1			58	57		64	1,739	10.18 : 89.82	870	10.23 : 89.77	2,560	51.25 : 48.75
2	2	(10,10)	(10,10)	58	44	0	4	1,740	10.17 : 89.83	869	10.24 : 89.76	3,196	54.82 : 45.18
	3			61	53		8	1,739	10.24 : 89.76	870	10.11 : 89.89	3,225	59.44 : 40.56
	1			100	69		4	4,858	5.48 : 94.52	2,428	5.48 : 94.52	7,770	55.39 : 44.61
3	2	(10,10)	(10,10)	99	63	0	8	4,857	5.48 : 94.52	2,429	5.48 : 94.52	7,916	54.83 : 45.17
	3			99	63		2	4,857	5.48 : 94.52	2,429	5.48 : 94.52	7,350	51.37 : 48.63
	1			96	53		4	3,461	5.20 : 94.80	1,729	5.21 : 94.79	6,056	54.16 : 45.84
4	2	(10,10)	(10,10)	97	59	0	2	3,460	5.20 : 94.80	1,730	5.20 : 94.80	5,834	54.51 : 45.49
	3			98	61		16	3,459	5.20 : 94.80	1,731	5.20 : 94.80	5,710	52.75 : 47.25
	1			63	41		1	2,948	4.99 : 95.01	1,473	4.96 : 95.04	2,508	52.15 : 47.85
5	2	(10,10)	(10,10)	61	42	0	2	2,946	4.96 : 95.04	1,475	5.02 : 94.98	2,234	51.30 : 48.70
	3			67	37		1	2,948	4.99 : 95.01	1,473	4.96 : 95.04	2,466	51.34 : 48.66
	1			71	25		1	10,878	0.57 : 99.43	5,436	0.55 : 99.45	2,352	50.77 : 49.23
6	2	(10,10)	(10,10)	68	21	0	4	10,876	0.56 : 99.44	5,438	0.57 : 99.43	2,040	50.49 : 49.51
	3			72	25		2	10,874	0.56 : 99.44	5,440	0.57 : 99.43	2,172	50.00 : 50.00

Table 4.8: The selected parameter values and distribution ratios of class 3 and non-class 3.

Sub-data	i^{th}fold	2D SOM neurons		Clusters		k	FFNN hidden units	Pre-training data set		Testing data set		Nearly balanced training data set	
		(x_-,y_-)	(x_+,y_+)	c_-	c_+			Total	Ratio of class 3 to non-class 3	Total	Ratio of class 3 to non-class 3	Total	Ratio of class 3 to non-class 3
	1			68	19		1	5,485	1.11 : 98.89	2743	1.09 : 98.91	1690	53.02 : 46.98
1	2	(10,10)	(10,10)	68	20	0	4	5,486	1.09 : 98.91	2742	1.13 : 98.87	1822	50.71 : 49.29
	3			70	20		2	5,485	1.11 : 98.89	2743	1.09 : 98.91	1864	50.54 : 49.46
	1			60	48		16	1,739	15.30 : 84.70	870	15.29 : 84.71	2960	50.95 : 49.05
2	2	(10,10)	(10,10)	55	54	0	32	1,740	15.29 : 84.71	869	15.30 : 84.70	3458	58.76 : 41.24
	3			56	62		8	1,739	15.30 : 84.70	870	15.29 : 84.71	3823	45.30 : 54.70
	1			99	84		8	4,858	11.53 : 88.47	2428	11.53 : 88.47	9996	54.20 : 45.80
3	2	(10,10)	(10,10)	99	78	0	16	4,857	11.53 : 88.47	2429	11.53 : 88.47	10659	54.80 : 45.20
	3			98	82		8	4,857	11.53 : 88.47	2429	11.53 : 88.47	9560	52.97 : 47.03
	1			95	76		8	3,461	12.28 : 87.72	1729	12.26 : 87.74	9779	53.16 : 46.84
4	2	(10,10)	(10,10)	99	76	0	32	3,460	12.28 : 87.72	1730	12.25 : 87.75	10099	53.08 : 46.92
	3			98	76		8	3,459	12.26 : 87.74	1731	12.31 : 87.69	10282	53.74 : 46.26
	1			60	43		16	2,948	6.78 : 93.22	1473	6.72 : 93.28	3037	55.22 : 44.78
5	2	(10,10)	(10,10)	64	45	0	4	2,946	6.75 : 93.25	1475	6.78 : 93.22	2823	52.60 : 47.40
	3			65	47		8	2,948	6.75 : 93.25	1473	6.79 : 93.21	2845	53.46 : 46.54
	1			63	24		2	10,878	0.91 : 99.09	5436	0.90 : 99.10	2080	53.65 : 46.35
6	2	(10,10)	(10,10)	68	23	0	4	10,876	0.91 : 99.09	5438	0.90 : 99.10	1978	51.26 : 48.74
	3			68	18		1	10,874	0.90 : 99.10	5440	0.92 : 99.08	2162	52.45 : 47.55

Table 4.9: The selected parameter values and distribution ratios of class 4 and non-class 4.

Sub-data	i^{th}fold	2D SOM neurons		Clusters		k	FFNN hidden units	Pre-training data set		Testing data set		Nearly balanced training data set	
		(x_-,y_-)	(x_+,y_+)	c_-	c_+			Total	Ratio of non-class 4 to class 4	Total	Ratio of non-class 4 to class 4	Total	Ratio of non-class 4 to class 4
1	1	(10,10)	(10,10)	72	27	0	2	5,485	1.79 : 98.21	2,743	1.79 : 98.21	2,505	52.10 : 47.90
	2			64	28		2	5,486	1.79 : 98.21	2,742	1.79 : 98.21	2,325	52.69 : 47.31
	3			71	23		2	5,485	1.79 : 98.21	2,743	1.79 : 98.21	2,452	54.24 : 45.76
2	1	(10,10)	(10,10)	52	63	0	16	1,739	26.91 : 73.09	870	26.90 : 73.10	3,991	49.11 : 50.89
	2			50	62		8	1,740	26.90 : 73.10	869	26.93 : 73.07	4,026	49.03 : 50.97
	3			58	64		4	1,739	26.91 : 73.09	870	26.90 : 73.10	4,123	48.56 : 51.44
3	1	(10,10)	(10,10)	99	89	0	64	4,858	18.32 : 81.68	2,428	18.33 : 81.67	12,154	55.83 : 44.17
	2			99	91		32	4,857	18.32 : 81.68	2,429	18.32 : 81.68	12,055	56.32 : 43.68
	3			98	95		32	4,857	18.32 : 81.68	2,429	18.32 : 81.68	11,837	55.88 : 44.12
4	1	(10,10)	(10,10)	94	87	0	32	3,461	19.50 : 80.50	1,729	19.43 : 80.57	11,352	56.29 : 43.71
	2			97	91		64	3,460	19.48 : 80.52	1,730	19.48 : 80.52	11,137	55.36 : 44.64
	3			96	92		64	3,459	19.46 : 80.54	1,731	19.53 : 80.47	12,211	56.78 : 43.22
5	1	(10,10)	(10,10)	63	58	0	32	2,948	12.55 : 87.45	1,473	12.49 : 87.51	3,485	58.79 : 41.21
	2			59	54		32	2,946	12.49 : 87.51	1,475	12.61 : 87.39	3,455	56.87 : 43.13
	3			60	52		32	2,948	12.55 : 87.45	1,473	12.49 : 87.51	2,494	52.21 : 47.79
6	1	(10,10)	(10,10)	65	30	0	16	10,878	1.58 : 98.42	5,436	1.55 : 98.45	2,916	52.26 : 47.74
	2			64	26		4	10,876	1.57 : 98.43	5,438	1.56 : 98.44	2,788	52.51 : 47.49
	3			70	31		4	10,874	1.55 : 98.45	5,440	1.60 : 98.40	2,964	52.23 : 47.77

4.2 Comparative Study

To point out pros and cons of this methods, SCL value (Zhou and Nakhleh, 2011), the conserved chemical content between two aligned compounds as stated by information from KEGG RPAIR database (Kotera et al., 2004) divided by the maximum chemical content of these two compounds, was calculated for each metabolite input query. To illustrate SCL value calculation, two biochemical reactions R1 and R2 (in Figure 3.1a) are demonstrated as the following equations: R1) $A+B \rightleftarrows C+D$ and R2) $C \rightleftarrows E+F$. A metabolite input query $h_i = \{A, C\}$ is a compound A and a compound C situated on each side of the first reaction. The originally defined SCL for $h_i = \{A, C\}$ when ignoring product and substrate information in the reaction is as follows:

$$SCL_{\{A,C\}} = \frac{|Cnt(A) \cap Cnt(C)|}{\max(|Cnt(A)|, |Cnt(C)|)},$$

where $Cnt(\cdot)$ is chemical content e.g. $Cnt(A)$ is calculated by counting non-hydrogen atoms of A. For SCL of a metabolite input query $h_j = \{A, E, C\}$, originally defined SCL is simply extended as follows:

$$SCL_{\{A,E,C\}} = \frac{|Cnt(A) \cap Cnt(E) \cap Cnt(C)|}{\max(|Cnt(A)|, |Cnt(E)|, |Cnt(C)|)}.$$

After calculating each SCL value for each metabolite input query, the comparison between the SCL method and the proposed method can be arranged. In addition, due to the pre−process of the original SCL value calculation, in the case of no information from KEGG RPAIR database, the SCL is set to zero. In this work, SIMCOMP (Hattori et al., 2010) was used for computing alignment of chemical contents.

For comparative tasks (Sections 4.2.1 to 4.2.3), besides the output values yielded from the proposed method called our result I, the adjusted output values called our$^{\#}$ result I is one method which adds the $2D$ structural compound alignment pre-process of SCL method as the post-process applied to the output values. Apart from that, the mean of our result I from four models is denoted as our result II which each mean value associated with its input pattern is the average of 4 output values from all 4 predictive models. The mean of our$^{\#}$ result I from four models is stated as our$^{\#}$ result II in a similar way. The improved performance result II from result I are expected.

The paired t-test at 5% significant level (Section 3.5) was explored (Section 4.2.2) for the

Table 4.10: *AUC* performance of the combined 3-fold testing data of the total sub-models when the number of sub-models, g, in the data splitting step is 6,8 and 9 for separating class q and non-class q for $q = 1,2,3,4$.

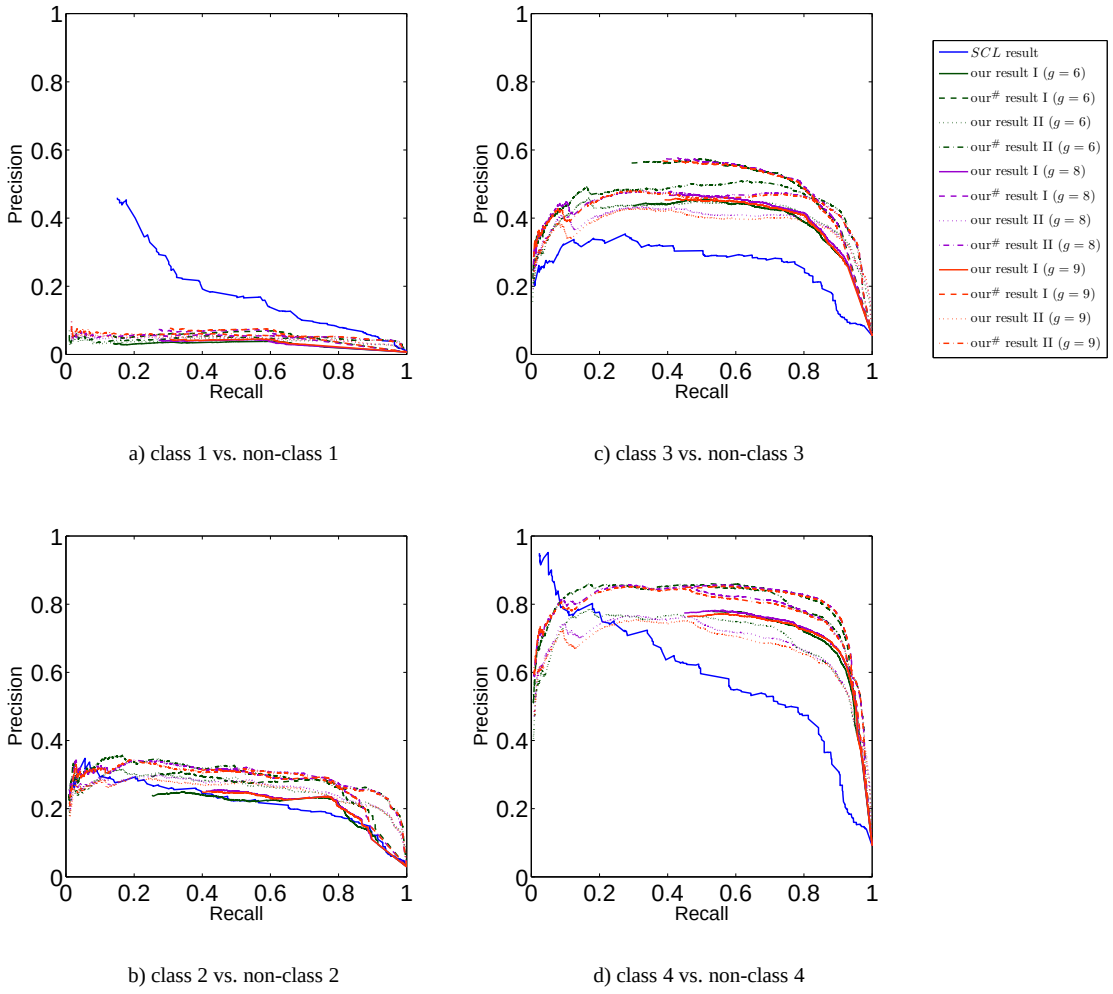
q	No.of sub-models	SCL	our result I	our result II	our$^{\#}$result I	our$^{\#}$result II
	6		0.7952	0.8980	0.8376	0.9209
1	8	**0.9548**	0.7759	0.9065	0.8055	0.9256
	9		0.7860	0.9067	0.8136	0.9264
	6		0.8975	0.9480	0.9115	**0.9553**
2	8	0.9113	0.8945	0.9474	0.9060	0.9544
	9		0.8947	0.9467	0.9065	0.9542
	6		0.9226	**0.9496**	0.9281	**0.9496**
3	8	0.8764	0.9249	0.9473	0.9276	0.9475
	9		0.9226	0.9461	0.9262	0.9470
	6		0.9614	0.9685	0.9625	**0.9729**
4	8	0.9147	0.9637	0.9675	0.9647	0.9716
	9		0.9634	0.9666	0.9650	0.9714

Each bold *AUC* value of each class q and non-class q model is the highest one.

paired performance value sets from two comparing methods. Each set is computed by the same performance evaluation metric. The 5% significance test at several cut-off values was done. Before doing the paired t-test, the effect of the various numbers of sub-models was observed (Section 4.2.1). Later, the unseen data prediction was analyzed (Section 4.2.3) in terms of sub-model and pathway viewpoints by using non-cut-off value metric like *AUC* (Section 4.2.3.1) as well as compound and pathway viewpoints by using correctness measurement at a chosen cut-off value (section 4.2.3.2 and 4.2.3.3). Additionally, they were visualized in forms of pathway maps (Section 4.2.3.4).

4.2.1 Sub-pre−training data size impact

At first, the pre-training data were separated into $g = 6$ sub-data according to a sub-data division step in the proposed method (Section 3.2.3). Then, overall acceptable predictive sub-models for questions 1 to 4 were trained and selected. To increase effectiveness of predictive class models, some of sub-data were recursively divided into $g = 8$ and $g = 9$ sub-data. The various score outcomes as above mentioned including SCL scores were compared by *AUC* values (Table 4.10). The very slightly different performance between $g = 6$, 8 and 9 in each type of scores indicates that $g = 6$ pre−training sub-data is enough to receive overall acceptable predictive sub-models for questions 1 to 4. Increasing numbers of pre-training sub-data is slightly increasing performance for overall predictive class 1 sub-models, but other class models are not



a) class 1 vs. non-class 1

c) class 3 vs. non-class 3

b) class 2 vs. non-class 2

d) class 4 vs. non-class 4

Figure 4.7: Precision-Recall graphs of the combined 3-fold testing data of the total sub-models when the numbers of sub-models, g, in the data splitting step are 6,8 and 9 for separating class q and non-class q for $q = 1, 2, 3, 4$.

in the same trends. Because the more sub-data may make the looseness of global information in training each predictive class model, but, in class 1 sub-models, they may reduce the complex of separation hyper-planes according to the neural network model building. However, the more amounts of sub-data certainly sacrifices more time in training and selecting satisfactory sub-models. Besides, the precision-recall graphs were plots in Figure 4.7 which each area under its curve also showed the resemble results as *AUC* values in Table 4.10. Both *AUC* values and precision-recall graphs from our$^{\#}$ result I/II mostly improve those from our result I/II. In the next experiments, $g = 6$ sub-models were selected in building each predictive class q model where $q = 1, 2, 3, 4$ and measured performance characteristics in many viewpoints.

4.2.2 Cut−off value variation and significance test

Both SCL values and output values, called the combined model result I, from an output layer of the combined neural network sub-models are in the range from 0 to 1, therefore various cut-off values were set from 0.05 to 0.95 increasing by 0.05 to separating both SCL scores and output scores into two classes. Apart from that, the same various cut-off values were applied to the mean output scores, called the combined model result II, as well. Then, to measure and compare the performance of SCL values, our output values and our$^{\#}$ output values in two forms of the combined model result I and II, each performance evaluation metric as in Figure 3.4 was computed at each cut-off value for different testing data and pre-training data from all 3 trails. The average performance values ware plotted as in Figures 4.8 to 4.11. Clearly, considering the whole performance results, the average output values denoted as the combined model result I from both our method and our$^{\#}$ method (see (a) in Figures 4.8 to 4.11) as well as the average mean output values denoted as the combined model result II from both our method and our$^{\#}$ method (see (b) in Figures 4.8 to 4.11) are comparative at almost every cut-off values. When focusing on only a cut-off value that yielded the highest performance value from each metric, the average SCL values seems to mostly loss other two comparing values (Figures 4.9 to 4.11) excepting the performance values of the predictive models for question 1 (Figure 4.8). Next, the cut-off values with the highest performance values between the combined model result I and II (Figures 4.9 to 4.11) were considered. The highest performance values of each metric in the combined model result II mainly belongs to our method or our$^{\#}$ method whereas such highest values only some of them belongs to our method or our$^{\#}$ method in the combined model result I. In contrast, the half of highest performance values in the combined model result I and II of the predictive models for question 1 (figure 4.8) belongs to SCL method while the half of them look inconclusive from the plotted graphs. Additionally, most of the performance outcomes in Figures 4.8 to 4.11 indicate that the models rather little over fit to the pre-training data sets than the testing data sets.

The next plots aim to show each performance at the selected cut-off values with significance test from diffrent values, comparatively. SCL values as well as the overall output values of each predictive class sub-model when the associated i^{th} fold testing sub-data was applied. The default cut-off value was 0.4 for evaluating performances of SCL values (Zhou and Nakhleh, 2011). Hence, to comparatively measure the performance of SCL values, our output values and our$^{\#}$ output values in two forms of the combined model result I and II, each performance evalu-

ation metric (as in Figure 3.4) was computed for all 3 folds of the testing data. Furthermore, the small, medium and high cut−off values were set as 0.3, 0.5 and 0.8, respectively, to separating both SCL scores and output scores into two classes. The additional metrics apart from those 5 metrics for selecting sub-models are as follows:

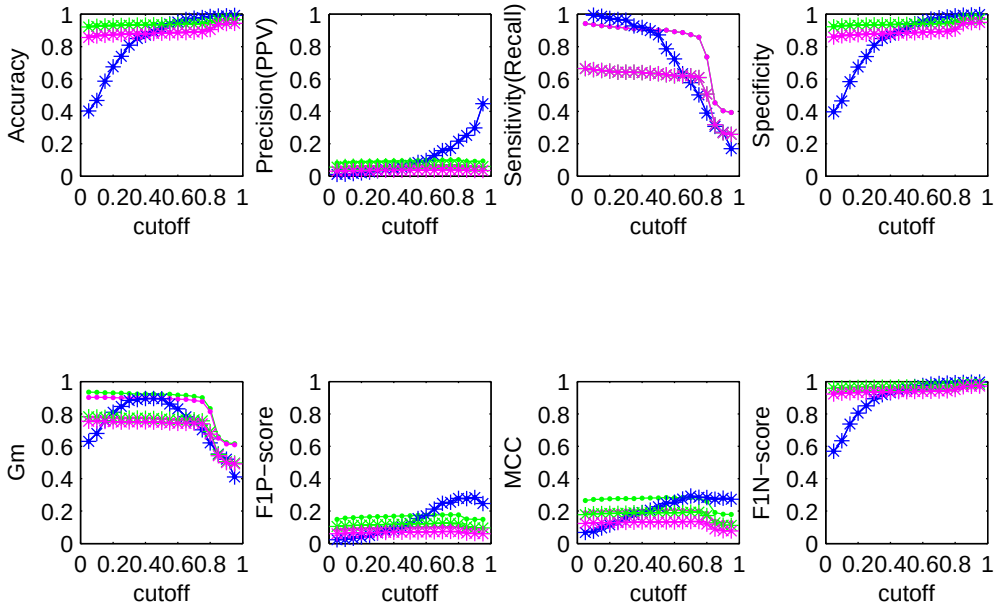
1) Precision (PPV) measures the fraction of truly predicted minority class samples in all samples that are predicted as minority class samples,

2) Positive class F_1-measure ($F1P$)is harmonic mean of precision and recall,

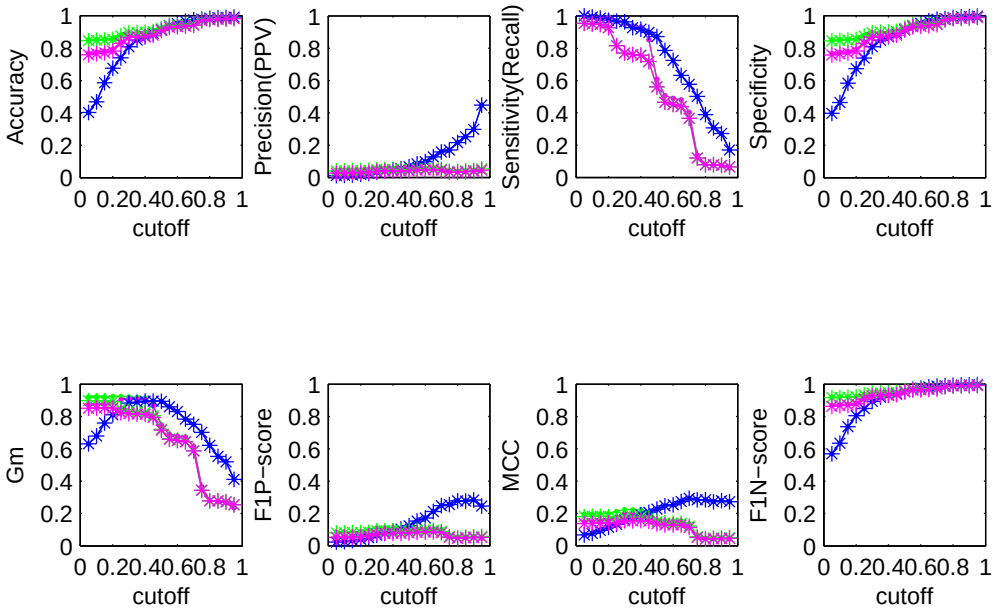
3) Negative class F_1-measure ($F1N$) is harmonic mean of TNR noting that negative predictive value (NPV) where NPV measures the fraction of truly predicted majority class samples in all samples that are predicted as majority class samples, and

4) Matthews correlation coefficient (MCC) measures the superiority of binary class classification which the high truly predicted both majority and minority class samples and the low wrongly predicted both majority and minority class samples produce the MCC values close to ideal value, 1.

Later, to compare each performance value from two different method results at each cut−off value, the paired *t*-test with 5% significant level was performed for every possible pair in the same score type i.e. I or II. The average performance values ware plotted (Figures 4.12 to 4.15). In addition, the average performance values are shown with their standard deviation if each of them significantly overcomes its compared performance value from another method result. Because of an assumption that the differences of each paired value in the process of the paired t-test must be normally distributed, Kolmogorov-Smirnov test for the normal distribution test (Gibbons and Chakraborti, 2003) was also performed in every comparing result pair of each performance metric. The null hypothesis is defined as the differences of each paired value follow the standard normal distribution whereas the alternative hypothesis is defined as such differences do not follow the standard normal distribution. There are three pair types (six pairs in total) for each performance evaluation e.g. result I/II of SCL method vs. our method, result I/II of SCL method vs. our$^{\#}$ method and result I/II of our method vs. our$^{\#}$ method. Every result pair of all performance evaluation failed to reject null hypothesis which implied that the differences of every result pair are normally distributed at 1% significant level.

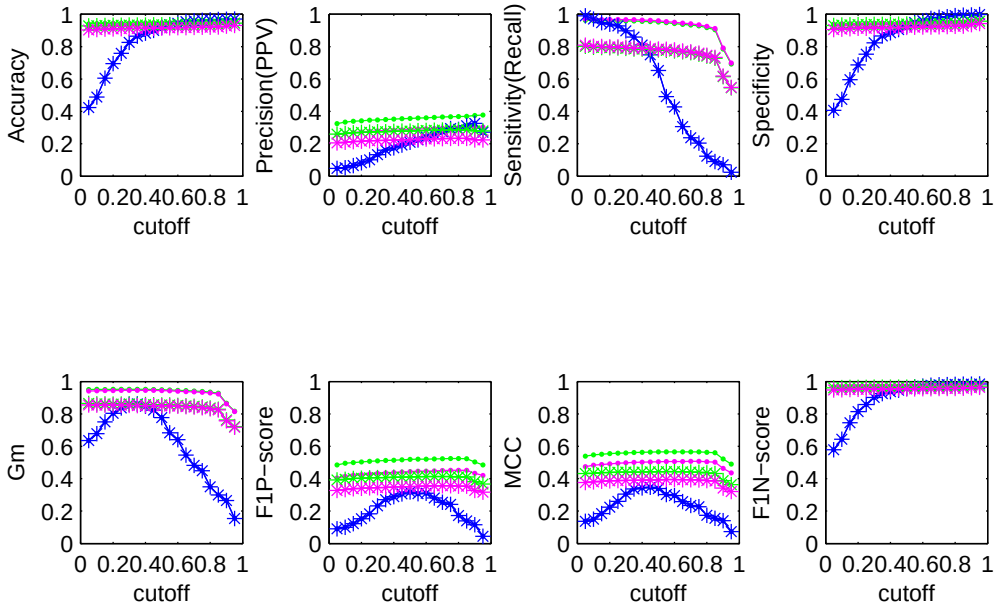


(a)Cut-off values versus its evaluation values on the combined model result I

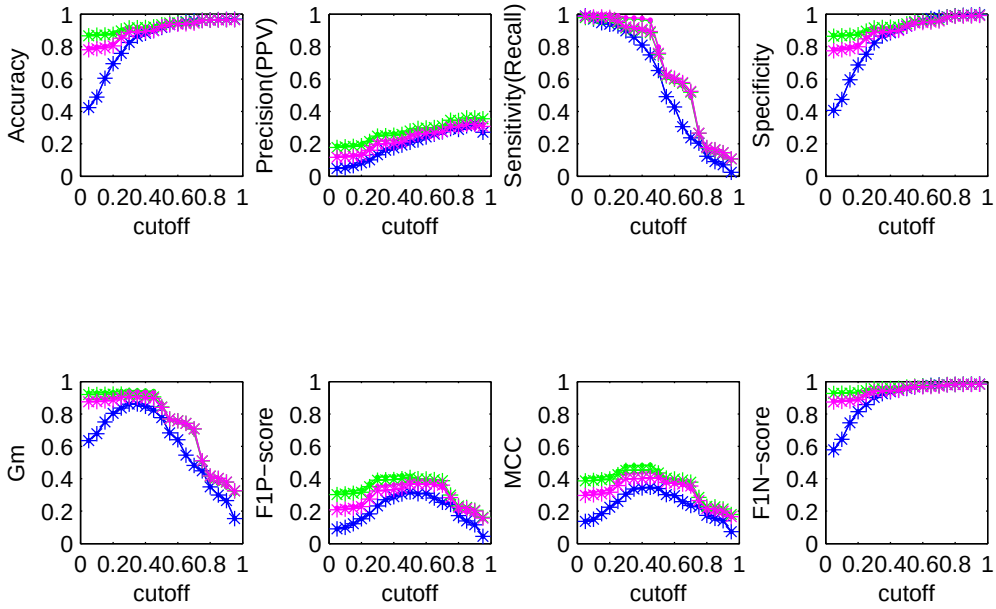


(b)Cut-off values versus its evaluation values on the combined model result II

Figure 4.8: Performance evaluation of class 1 vs. non-class 1 prediction model at cut-off value = $0.05, 0.1, 0.15, ..., 0.95$. Dot and asterisk denote the pre-training data sets and the testing data sets, respectively, while blue, magenta and green represent each average performance value resulting from applying the 3-fold pre-training/testing data sets to SCL methods, our methods and our$^{\#}$ methods, respectively.

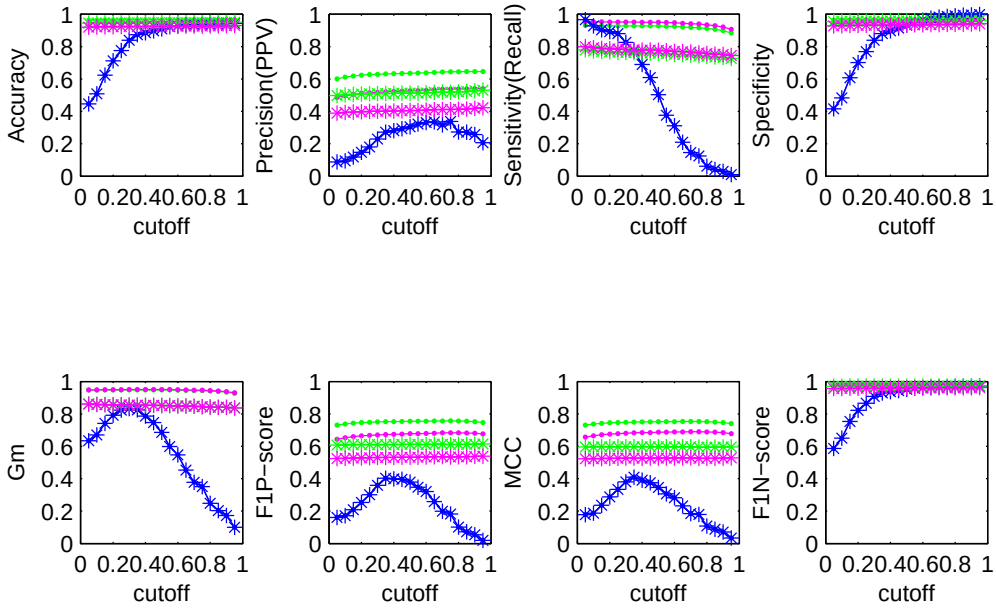


(a)Cut-off values versus its evaluation values on the combined model result I

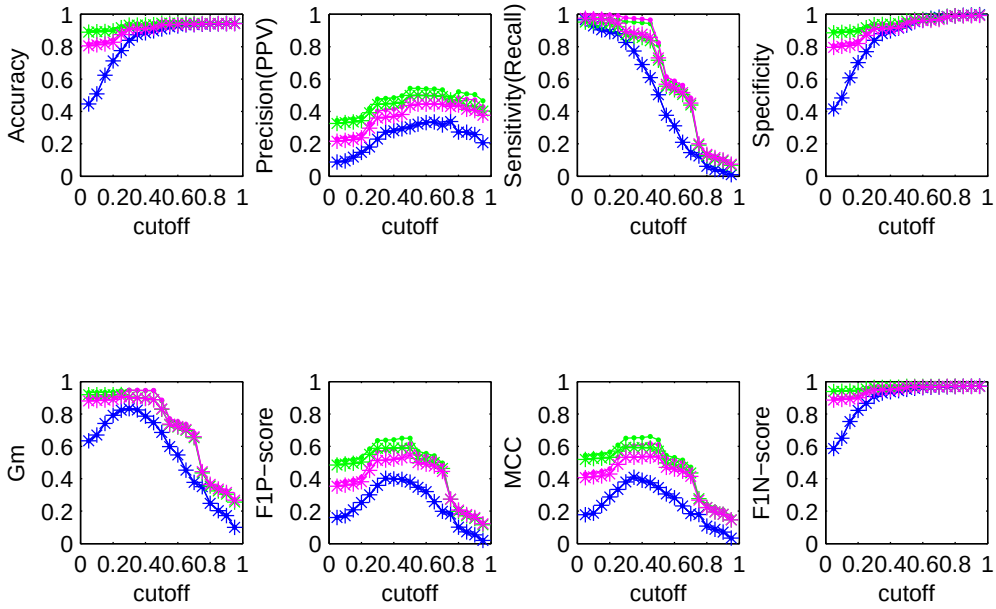


(b)Cut-off values versus its evaluation values on the combined model result II

Figure 4.9: Performance evaluation of class 2 vs. non-class 2 prediction model at cut-off value = $0.05, 0.1, 0.15, ..., 0.95$. Dot and asterisk denote the pre-training data sets and the testing data sets, respectively, while blue, magenta and green represent each average performance value resulting from applying the 3-fold pre-training/testing data sets to SCL methods, our methods and our$^{\#}$ methods, respectively.

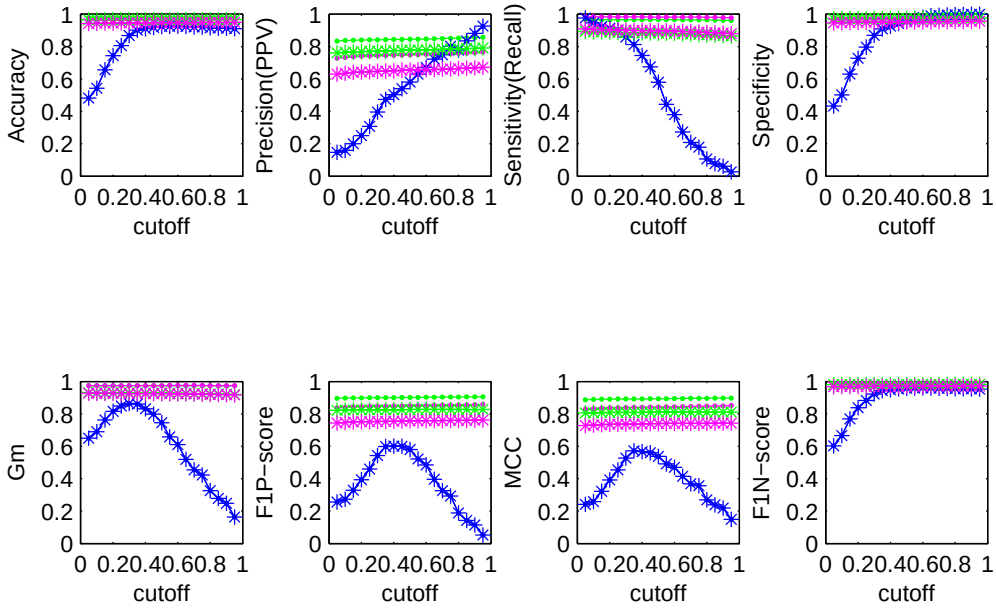


(a)Cut-off values versus its evaluation values on the combined model result I

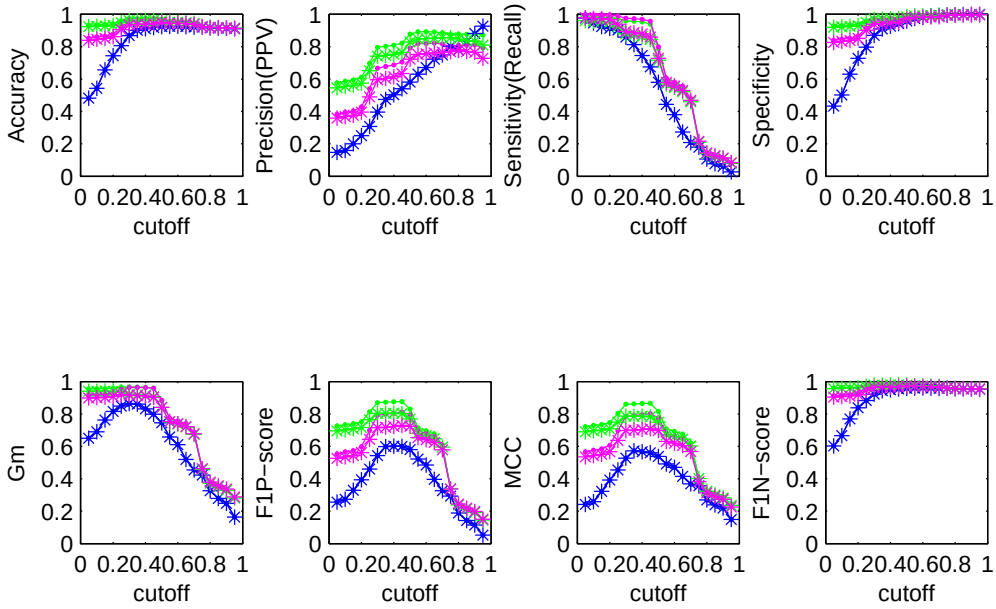


(b)Cut-off values versus its evaluation values on the combined model result II

Figure 4.10: Performance evaluation of class 3 vs. non-class 3 prediction model at cut-off value $= 0.05, 0.1, 0.15, ..., 0.95$. Dot and asterisk denote the pre-training data sets and the testing data sets, respectively, while blue, magenta and green represent each average performance value resulting from applying the 3-fold pre-training/testing data sets to SCL methods, our methods and our$^{\#}$ methods, respectively.



(a)Cut-off values versus its evaluation values on the combined model result I



(b)Cut-off values versus its evaluation values on the combined model result II

Figure 4.11: Performance evaluation of class 4 vs. non-class 4 prediction model at cut-off value $= 0.05, 0.1, 0.15, ..., 0.95$. Dot and asterisk denote the pre-training data sets and the testing data sets, respectively, while blue, magenta and green represent each average performance value resulting from applying the 3-fold pre-training/testing data sets to SCL methods, our methods and our$^{\#}$ methods, respectively.

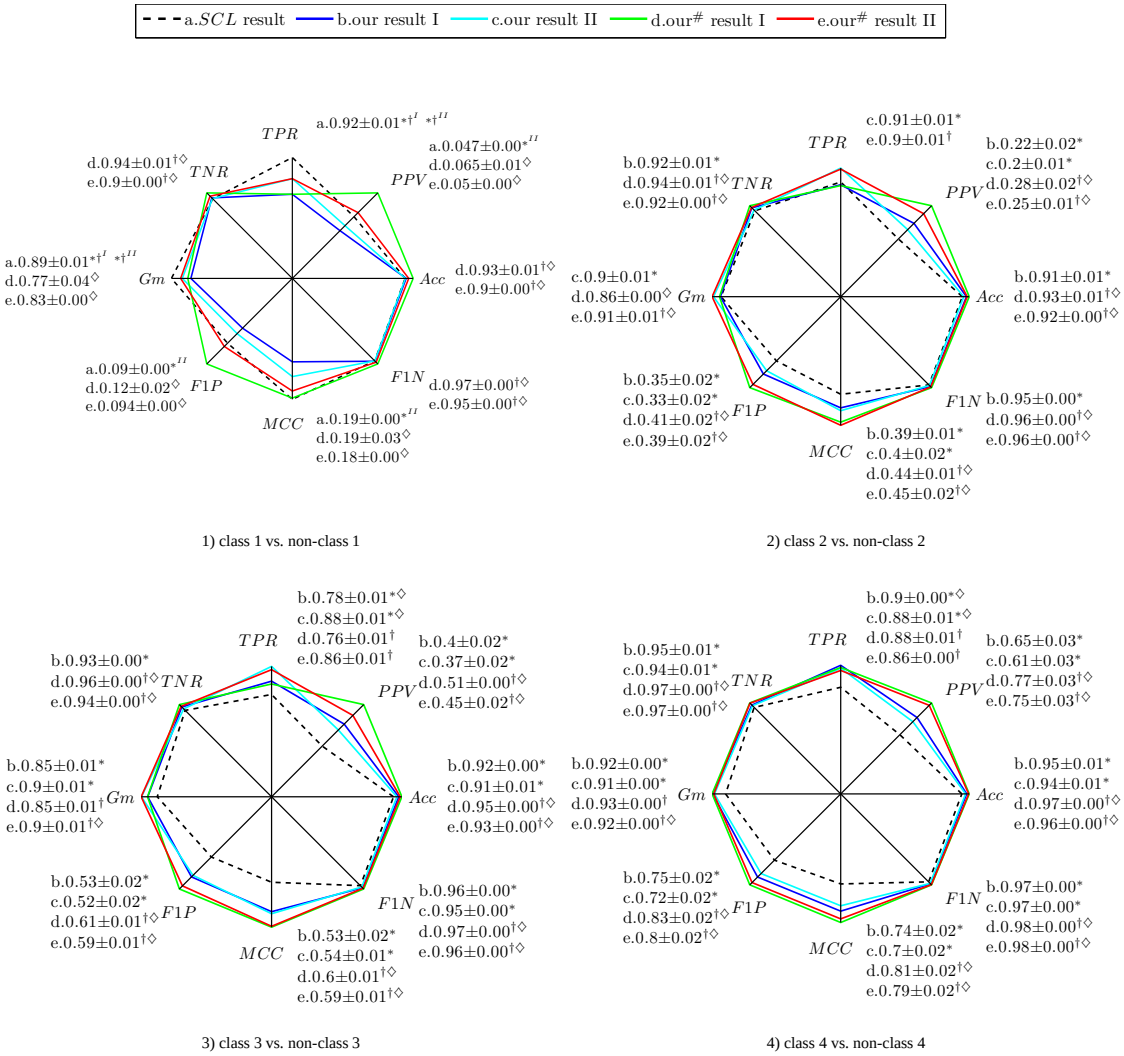


Figure 4.12: Performance evaluation of five different scores (a-e) is comparatively depicted. Each performance evaluation result at cut-off value = 0.4 of every class q vs. non-class q model, $q = 1, 2, 3, 4$, was plotted by each average performance value of each score type (a-e) resulting from applying the 3-fold testing data.In each axis, all five average values were comparatively scaled into values between 0 to 1. A maximum one was scaled to one. Additionally, at 5% level of significance, the average results with standard deviation of the significantly outperforming performance analysed by the paired t-test significance test are shown. Denote that *, †and $^\diamond$ are a significantly better performance results in the types of scores, I or II, from the following paired method: SCL method vs. our method, SCL method vs. our$^\#$ method, and our method vs. our$^\#$ method, respectively.

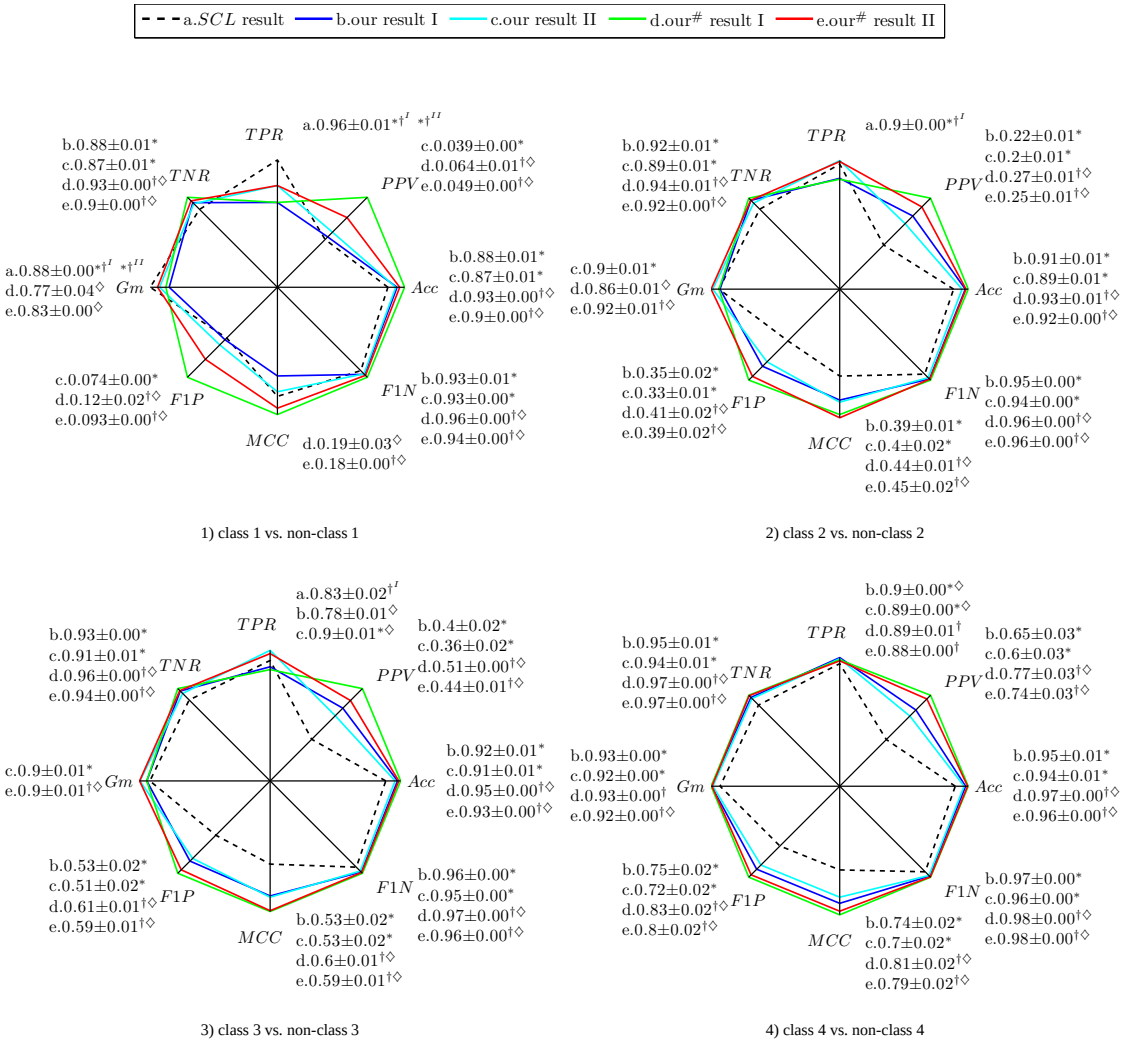


Figure 4.13: Performance evaluation of five different scores (a-e) is comparatively depicted. Each performance evaluation result at cut-off value = 0.3 of every class q vs. non-class q model, $q = 1,2,3,4$, was plotted by each average performance value of each score type (a-e) resulting from applying the 3-fold testing data.In each axis, all five average values were comparatively scaled into values between 0 to 1. A maximum one was scaled to one. Additionally, at 5% level of significance, the average results with standard deviation of the significantly outperforming performance analysed by the paired t-test significance test are shown. Denote that *, †and $^\diamond$ are a significantly better performance results in the types of scores, I or II, from the following paired method: SCL method vs. our method, SCL method vs. our$^\#$ method, and our method vs. our$^\#$ method, respectively.

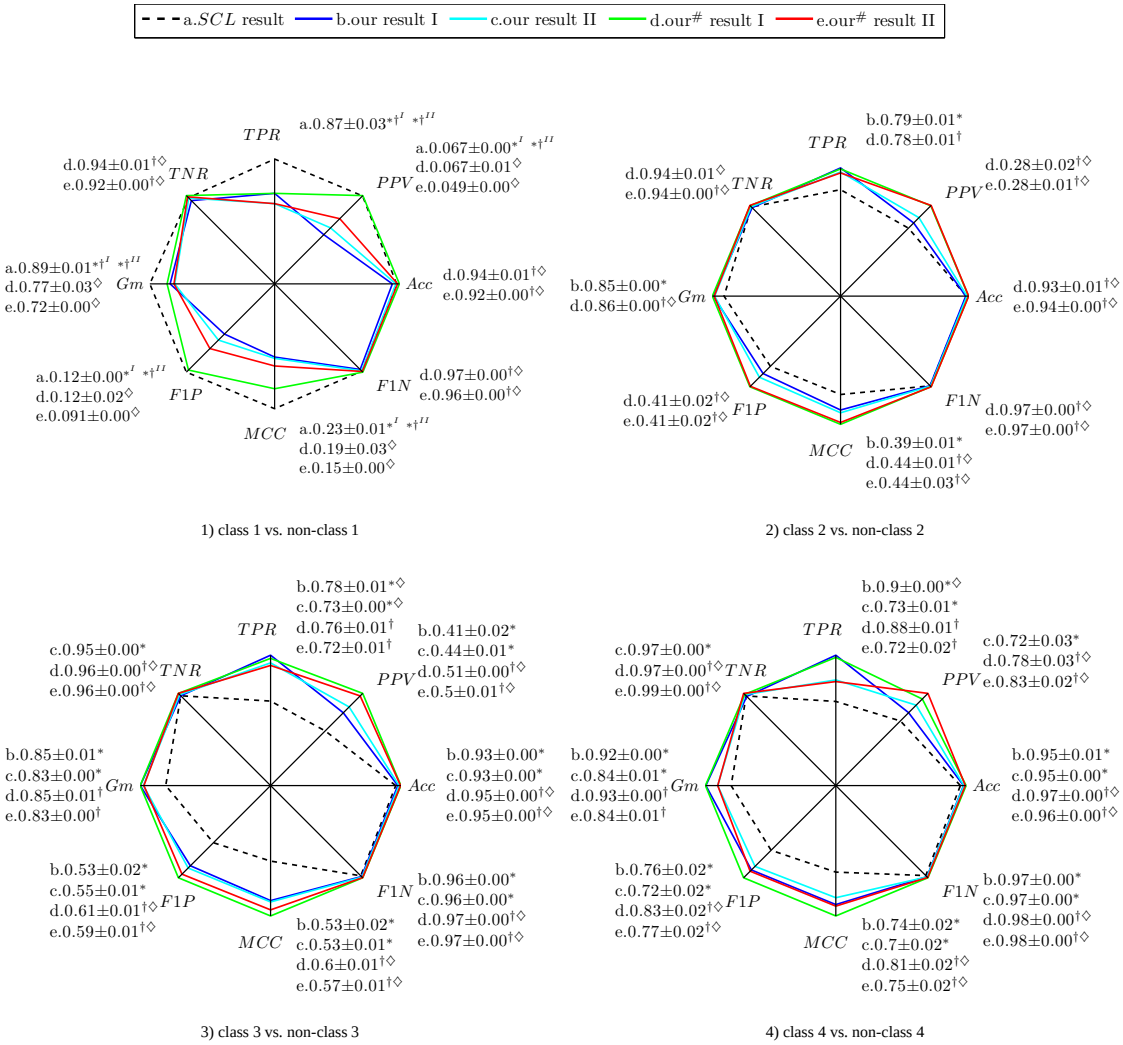


Figure 4.14: Performance evaluation of five different scores (a-e) is comparatively depicted. Each performance evaluation result at cut-off value = 0.5 of every class q vs. non-class q model, $q = 1, 2, 3, 4$, was plotted by each average performance value of each score type (a-e) resulting from applying the 3-fold testing data.In each axis, all five average values were comparatively scaled into values between 0 to 1. A maximum one was scaled to one. Additionally, at 5% level of significance, the average results with standard deviation of the significantly outperforming performance analysed by the paired t-test significance test are shown. Denote that *, †and $^\diamond$ are a significantly better performance results in the types of scores, I or II, from the following paired method: SCL method vs. our method, SCL method vs. our$^\#$ method, and our method vs. our$^\#$ method, respectively.

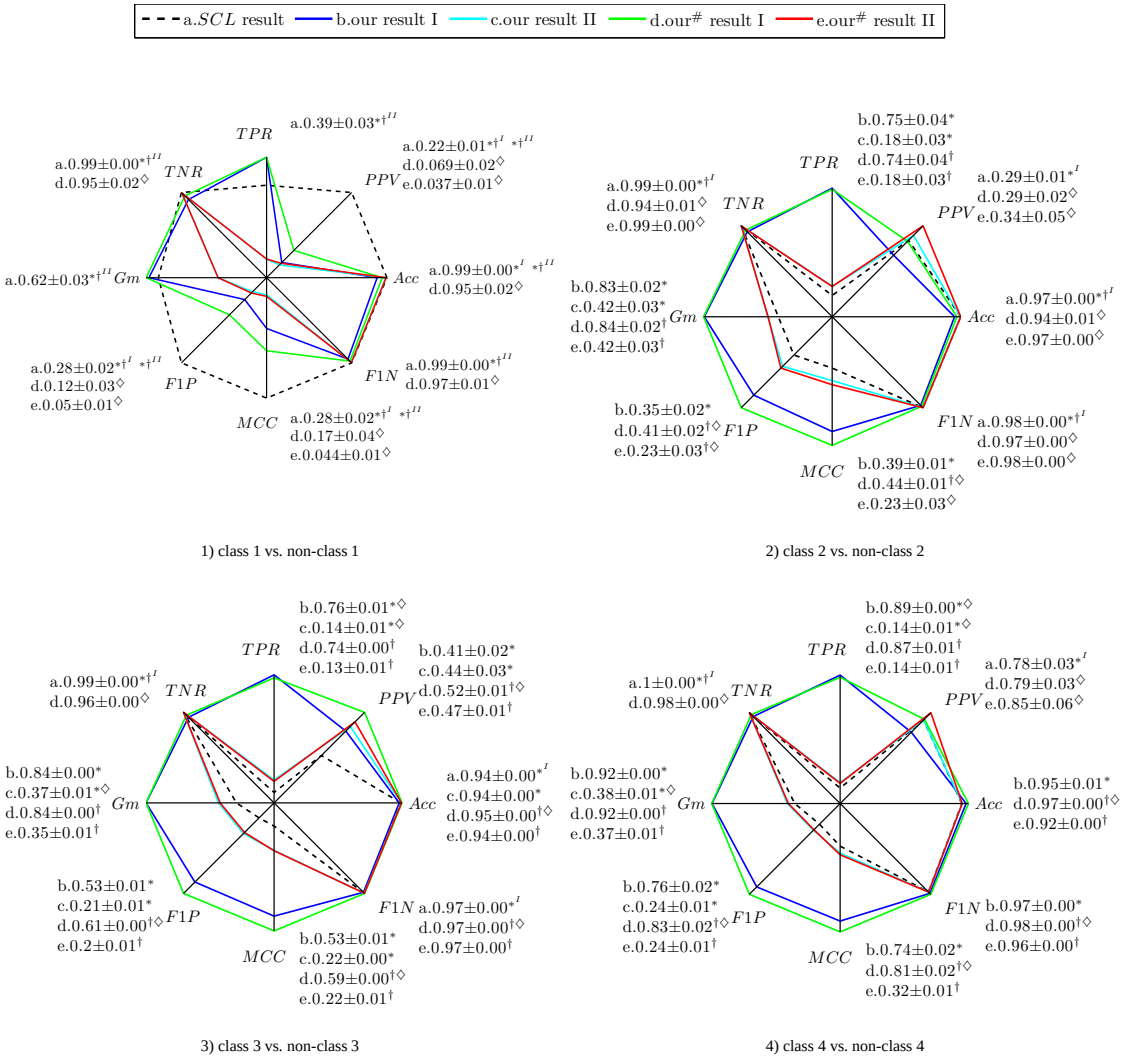


Figure 4.15: Performance evaluation of five different scores (a-e) is comparatively depicted. Each performance evaluation result at cut-off value = 0.8 of every class q vs. non-class q model, $q = 1, 2, 3, 4$, was plotted by each average performance value of each score type (a-e) resulting from applying the 3-fold testing data.In each axis, all five average values were comparatively scaled into values between 0 to 1. A maximum one was scaled to one. Additionally, at 5% level of significance, the average results with standard deviation of the significantly outperforming performance analysed by the paired t-test significance test are shown. Denote that *, †and $^\diamond$ are a significantly better performance results in the types of scores, I or II, from the following paired method: SCL method vs. our method, SCL method vs. our$^\#$ method, and our method vs. our$^\#$ method, respectively.

Comparing performance values at cut-off value = 0.4 (Figure 4.12), the mostly common significant performance results come from our# method in both I and II (line d-e) for all 4 predictive models. These suggest that the combination of the proposed method and irrelevant compound filtering using $SCL = 0$ or the existent compound pairs in RPAIR database works well for predicting various defined degrees of relevant compounds (Section 3.2.1). In addition, the general performance values of models that the balanced training data originated from the pre-training data with the less degrees of the imbalanced situation tend to be high. Moreover, the small, medium and high cut−off values were set as 0.3, 0.5 and 0.8, respectively, therefore, the same representation of the performance values was shown (Figures 4.13 to 4.15). The typical results are also similar to Figure 4.12 except the class 1 vs. non-class 1 model. The outcomes of cut-off variation and significance test point out that cut-off variation obviously affects the general performance results of all compared methods in the class 1 vs. non-class 1 model which it contains the highest degrees of imbalanced pre-training data. However, in the cases of the lesser degrees of imbalanced pre-training data, the predictive model performance can be improved by the proposed methods. Especially, our# method in both I and II.

4.2.3 Unseen data prediction and comparison

In this experiment, the four models from combining each of six selected sub-models corresponding to each class q model for $q = 1, 2, 3, 4$ were used in the tasks of predicting unseen data patterns prepared by the 7-pathway examples. Each unseen input pattern was predicted by three models derived by the previous 3-fold cross validation procedure, then a max output value was a predicted output value. Theses following 7 *E.coli* pathways were selected from 7 different pathway functions:

1) purine metabolism involved in a process of nucleotides,

2) valine leucine and isoleucine biosynthesis involved in a process of proteins,

3) streptomycin biosynthesis involved in a process of secondary metabolites,

4) methane metabolism involved in a process of energy metabolism,

5) nicotinate and nicotinamide metabolism involved in a process of cofactors and vitamins,

6) phospholipid biosynthesis involved in a process of lipids, and

7) pyruvate oxidation pathway involved in a process of carbohydrates.

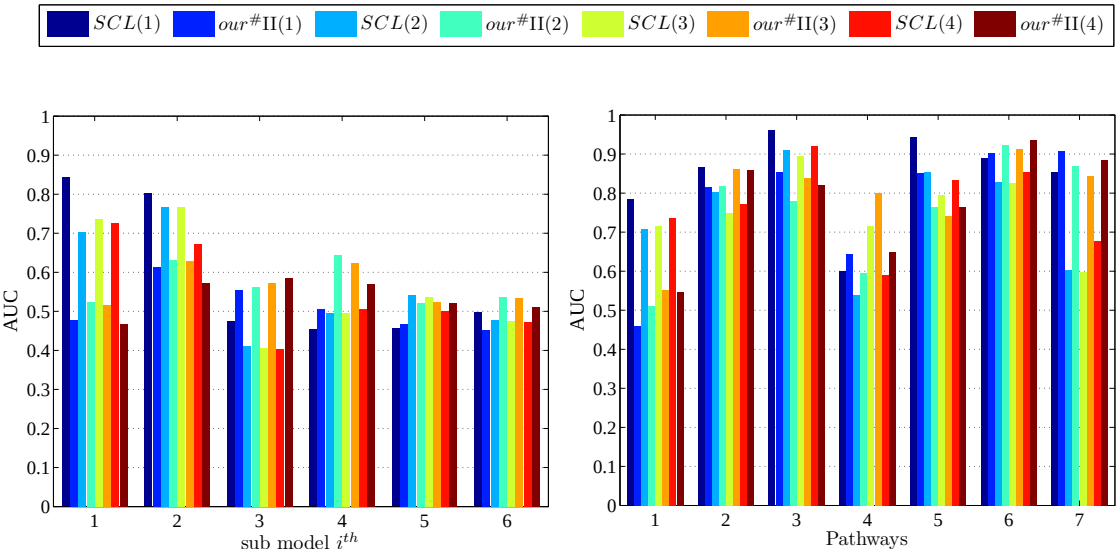
These pathways were downloaded from KEGG Pathway as the same version of KEGG Ligand database used for preparing relevant compound features except the last two pathways came from aMAZE database as the same database we used in model building (see Section 3.4). Their relationship according to our four defined questions was extracted into a data pattern set with four target sets.

4.2.3.1 *AUC* performance: sub-model vs. pathway perspective

A data pattern set with four targets was divided into 6 sub-data according to a key feature calculated in a sub-data division step (Section 3.2.3). First one, the output scores associated with each unseen data pattern were predicted by each sub-model of classes 1 to 4. Then, based on the results of the previous section, *AUC* performance evaluation of all 6 sub-data was calculated by applying scores from our$^{\#}$ method II. Another one, the output scores yielded by each sub model of class q was gathered and re-divided according to each pathway example they are associated with. Then, *AUC* values of scores from our$^{\#}$ method II was computed. Afterwards, both *AUC* values were compared with *AUC* values of SCL method.

Focusing on every class q model in Figure 4.16, *AUC* results of scores from our$^{\#}$ method II in the 3^{rd}-4^{th} and the 5^{th}-6^{th} sub-models are clearly better and slightly better than results

of scores from SCL method, respectively, while, AUC results of scores from SCL method in the 1^{st}-2^{nd} sub-models obviously outperform results of another method. AUC values of scores from SCL method in purine metabolism, streptomycin biosynthesis, and nicotinate and nicotinamide metabolism are greater than results of scores from our$^{\#}$ method II. However, for the most parts, AUC results of scores from our$^{\#}$ method II in the rest four pathway examples are higher than AUC values of scores from SCL method. In the sub-model view, the large parts of overall sub-models yield output scores from our$^{\#}$ method II with the better AUC performances. In the pathway perspective, both methods seem to complement each other with the different efficient AUC performances. Interestingly, some drawbacks of SCL score is ineffective in some compounds involved lipid pathways and a pyruvate oxidation pathway like acetyl-CoA and acetate (Zhou and Nakhleh, 2011), but, the greater AUC results of scores from our$^{\#}$ method II in these two cases were found.



a) AUC performance evaluation according to each selected predictive sub-model

b) AUC performance evaluation according to each pathway example

Figure 4.16: AUC performance comparison between the combine model result II by our$^{\#}$ method and those results by SCL method for all 4 defined question models. Note that the new unseen input data were derived from the 7-pathway examples (see Section 4.2.3)

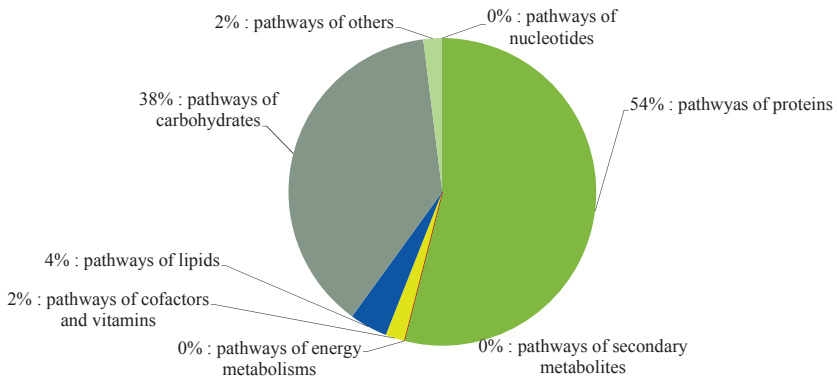


Figure 4.17: Distribution amount of pathways according to metabolite types of 50 *Ecoli* pathways from aMAZE (Lemer et al., 2004) that their involed reaction and metabolite sets were used in model training processes (Section 3.4).

4.2.3.2 Correctness performance: compound perspective

In the form of correctness measurement, accuracy *Acc* measures how good the method correctly predict in both binary classes in a data pattern set is. However, in the imbalanced binary class data, the majority class accurateness affects *Acc*. Then, G-mean *Gm* would be suitable to represent the reasonable correctness of each binary classes since it is the geometric mean of TPR and TNR (Figure 3.4). In this task, the 172 input-target sets were prepared due to all available 2D compound structures from the 7-pathway examples. For each set, every metabolite query in such set has one distinct compound in common. After that, to measure how accurate they are in the compound viewpoint, correctness performance denoted as C based on Gm, TPR and TNR were offered as the following: a) if that set contains binary targets, Gm performance was calculated as C of the output scores at cut-off=0.5 from our method I (C_{our}), those from our$^{\#}$ method I ($C_{our^{\#}}$), and those from SCL method (C_{SCL}) in comparison; and b) if that set contains only either a positive or a negative target, either TPR or TNR performance was calculated as C of the output scores at cut-off=0.5, namely, C_{our}, $C_{our^{\#}}$, and C_{SCL}.

C_{our}, $C_{our^{\#}}$, and C_{SCL} of the 172 input-target sets were calculated for all four defined questions. Each C performance value is in $[0,1]$. To visualize and simplify all C results (see figure 4.18), C values were categorized into four levels as the following: 1) $C \in [0, 0.25]$ are the low values denoted as white color; 2) $C \in (0.25, 0.50]$ are the medium-low values denoted as green yellow color; 3) $C \in (0.50, 0.75]$ are the medium-high values denoted as brown color; and 4) $C \in (0.75, 1]$ are the high values denoted as blue color.

In Figure 4.18, C_{our}, $C_{our^{\#}}$, and C_{SCL} of the 121 input-target sets for all four defined questions has been shown. The rest C results of the 51 input-target sets are omitted because none of them are the high C values. For the 121 input-target sets which each of them is associated with a compound, they are displayed as different colors as the following:

1) some of them are dark green color labeled as the positive seen metabolites if they are involved in the training data and they also appear on the reference maps,

2) some of them are italic and light green color labeled as the negative seen metabolites if they are involved in the training data and they do not appear on the reference maps,

3) some of them are dark red color labeled as the positive unseen metabolites if they are not involved in the training data and they also appear on the reference maps, and

4) some of them are italic and light red color labeled as the egative unseen metabolites if

they are not involved in the training data and they do not appear on the reference maps.

Additionally, 66 in 121 distinct common compounds in each set are the seen metabolites. Apart from that, for each input-target set associated with a compound, such compounds participated in one or more than one of 7-pathway examples (see Section 4.2.3) were also identified.

The groups of C results can be analyzed across different defined questions and compared methods. First, almost all of C results of input-target set id 1-26 associated with each compound have high C values across all defined questions and compared methods. More than half of them are the input-target sets of the negative seen metabolites. Second, C results of input-target set id 27-41 associated with each compound were explored. In question 1 model, it was found that set id 27-32 yielded results from three methods with high C values whereas set id 33-41 yielded results from our and our$^{\#}$ methods with high C values. In question 2-4 models, they were found that a third of them, id 27-32, which are the positive unseen metabolites have the high C_{SCL} values as well as around a third of them, id 33-36, which are the positive seen metabolites have the high C_{our} and $C_{our^{\#}}$ values. Focusing on set id 27-32, most of them are from example pathways concerning nucleotides, secondary metabolites and energy metabolisms which are known as the rare trained data in the training processes (Figure 4.17). Almost all of set id 33-41 are the positive seen metabolites that they failed to achieve high C_{SCL} values across four defined questions. When considering each compound involved in set id 27-41, the most compounds are phospholipids, i.e. Glycerone phosphate, 1,2-Diacyl-sn-glycerol, CDP-diacylglycerol, Phosphatidylglycerophosphate, Phosphatidylserine. Such compounds with long chain shapes and their route characteristics of lipid transformations cause no high SCL scores (Zhou and Nakhleh, 2011). The second groups are compounds carrying formyl or acetyl groups for attaching to other compounds by its roles i.e. Acetyl-CoA, Methanol, Formaldehyde(Methanal), and Formate. An input-target set associated with Glycine yields high correctness in questions 1 and 4 models since Glycine is an amino acid with simple structure and also the positive seen metabolites as the trained data. It is involved in many pairs in class 1 like a hub of transformation, so this may be a reason that yielded high C values in question 1 and 4 models. The rest of them are myo-Inositol, 1D-myo-Inositol 3-phosphate, Urate, Nicotinamide and Oxalureate which all positive unseen metabolites excepting Nicotinamide. Because they are not obvious to discuss about their high C_{our} values via only visualization of resemble structures or their route characteristics, these may be implied that the trained neural network models can effectively predict these kinds of them. Third, the most of input-target set

id 42-60 associated with each compound are from as Valine Leucine and Isoleucine biosynthesis which is protein pathways and such pathways are the main parts in the trained data (see Figure 4.17). In question 1 model, all $C_{our^{\#}}$ values are high while a half of C_{SCL} values are high. In other question models, $C_{our^{\#}}$ values are still high in a large group of sets. Fourth, *C* results of input-target set id 61-74 associated with each compound were investigated. In question 1 and 3 models, the high C_{SCL} values are the main results, but in question 2 and 4 models, the high $C_{our^{\#}}$ values are the main results. Same as input-target set id 42-60, the most of compounds are from Valine Leucine and Isoleucine biosynthesis. The last part, *C* results of input-target set id 75-121 were considered, only in question 1 model can yield mainly high correctness from SCL method. Nearly all sets are associated with compounds from Purine metabolism which is nucleotide pathways and such pathways are not participated in the trained data (Figure 4.17).

In conclusion from the compound perspective, filtering irrelevant 2D structure pair of compounds as our$^{\#}$ method can improve the output values from the proposed method because it helps to eliminate noise results in some cases e.g. set id 42-60. Apart from that, almost all cases of high C_{our} values are also high $C_{our^{\#}}$ values. Besides, input-target sets that yielded either high C_{our} values or $C_{our^{\#}}$ values in at least one question model are mainly associated with the seen metabolites involved in the trained data.

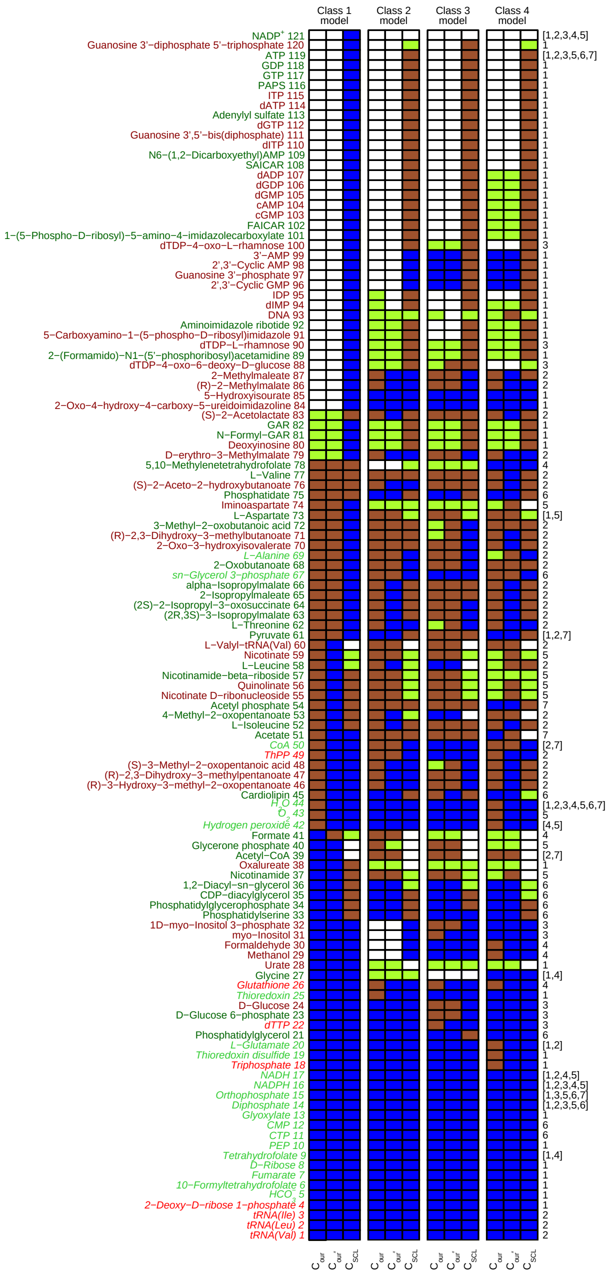


Figure 4.18: The correctness performance, C, evaluation of four defined models in compound perspectives. Each class model, the output scores were obtained by SCL method, our method I, and our$^{\#}$ method I. From 174 compounds obtained by the 7-pathway examples, there are 172 available 2D compound structures. These are 121 results with at least one high C value of 172 input-target sets such that every metabolite query in a set contains one compound in common. The right side shows pathway example id (see Section 4.2.3) which each compound is participated. C values were categorized into four levels as the following: 1) $C \in [0, 0.25]$ denoted as white color; 2) $C \in (0.25, 0.50]$ denoted as green yellow color; 3) $C \in (0.50, 0.75]$ denoted as brown color; and 4) $C \in (0.75, 1]$ denoted as blue color. Each set involved each compound is displayed as different colors as the following: 1) dark green color labelled as positive seen metabolites; 2) italic and light green color labelled as negative seen metabolites; 3) dark red color labelled as positive unseen metabolites; and 4) italic and light red color labelled as negative unseen metabolites. Note that the details of seen/unseen metabolites and C computation are in Section 4.2.3.2.

4.2.3.3 Correctness performance: pathway perspective

In this task, the metabolite queries prepared from the 7-pathway examples were gathered into each input-target set associated with each pathway example (see Section 4.2.3). In one input-target set, every metabolite query contains at least a metabolite that is participated in such considered pathway. Then, the output results of each set for all four defined question were evaluated the correctness performance, C, in three compared method as Section 4.2.3.2. In addition, the way to define class target becomes an issue in comparison. In the definition called *metabolite transformation network* (Section 3.1), every existing compound pairs in the reference maps can be transformable in one step when there exist at least a reaction to change one to another. This was defined because of the following two reasons. First, each metabolite acting as a main or side compound did not be defined in networks since the results of trained model would express and define them. Second, based on SCL method, it also prefers this definition, so it would be a reason for comparison. However, C performance when the transformation obtained from original reference maps like xml files from KEGG pathway database was also shown. In Table 4.11, the four defined class targets obtained from both metabolite transformation network and original reference map yielded the same trends of correctness performance, C, in each of the 7-pathway examples. Both Purine metabolism and Streptomycin biosynthesis are the kind of the rare data in the training data set, so C_{SCL} values are maximum in all four questions comparing to C_{our} and $C_{our^{\#}}$ values. However, Methane metabolism is also the kind of the rare data in the training data set, but maximum values are mainly obtained from C_{our} values. Apart from that maximum C_{our} values still be found in main results of Pyruvate oxidation pathway which is carbohydrate pathways. These kinds of pathways are not tiny parts of the training data set. All maximum values across four defined questions are obtained from $C_{our^{\#}}$ values in an input-target set of Phospholipid biosynthesis which is lipid pathways although lipid pathways are small parts of the training data set. All maximum $C_{our^{\#}}$ values were also shown from results of Nicotinate and Nicotinamide metabolism which is also the kind of very small data in the training data set. The output results of valine Leucine and Isoleucine biosynthesis yielded the maximum C values from $our^{\#}$ or SCL method in different defined questions.

In brief, the correctness performance at a chosen cut-off value in both pathway and compound perspectives indicate that the trained data must be sufficient to cover considered kinds of pathways such that the required accurate prediction was achieved by the trained model (Tables 4.11 to 4.12).

Table 4.11: The C performance evaluation of four defined cases in pathway perspectives. Each class model, the output scores were obtained by SCL method, our method I, and our$^{\#}$ method I. There are 7 input sets with 2 target sets according to 7 metabolite query sets of the 7-pathway examples. The two target sets for comparison were obtained by definition called metabolite transformation network and original reference maps (see Section 4.2.3.3).

The 7-pathway examples of *E.coli*	C	Metabolite transformation network				Original reference maps			
		Class 1	Class 2	Class 3	Class 4	Class 1	Class 2	Class 3	Class 4
Purine Metabolism	C_{SCL}	**0.731**	**0.590**	**0.518**	**0.538**	**0.773**	**0.656**	**0.581**	**0.594**
	C_{our}	0.227	0.237	0.336	0.275	0.328	0.244	0.336	0.272
	$C_{our^{\#}}$	0.194	0.225	0.336	0.267	0.328	0.241	0.342	0.274
Valine Leucine and Isoleucine Biosynthesis	C_{SCL}	**0.773**	**0.722**	0.673	0.691	**0.774**	**0.718**	0.676	0.691
	C_{our}	0.644	0.588	0.757	0.761	0.659	0.591	0.757	0.761
	$C_{our^{\#}}$	0.692	0.618	**0.820**	**0.803**	0.708	0.621	**0.821**	**0.803**
Streptomycin Biosynthesis	C_{SCL}	**0.838**	**0.634**	**0.548**	**0.657**	**0.838**	**0.634**	**0.548**	**0.657**
	C_{our}	0.449	0.477	0.247	0.363	0.449	0.477	0.247	0.363
	$C_{our^{\#}}$	0.464	0.517	0.253	0.371	0.464	0.517	0.253	0.371
Methane Metabolism	C_{SCL}	0.509	0.322	**0.378**	0.391	0.623	0.000	0.000	0.329
	C_{our}	**0.680**	**0.441**	0.251	**0.516**	**0.745**	**0.457**	**0.573**	**0.704**
	$C_{our^{\#}}$	0.557	0.397	0.268	0.428	0.683	0.424	0.433	0.593
Nicotinate and Nicotinamide Metabolism	C_{SCL}	0.652	0.374	0.265	0.325	0.652	0.374	0.265	0.325
	C_{our}	0.636	0.489	0.430	0.483	0.636	0.489	0.430	0.483
	$C_{our^{\#}}$	**0.704**	**0.500**	**0.466**	**0.505**	**0.704**	**0.500**	**0.466**	**0.505**
Phospholipid Biosynthesis	C_{SCL}	0.739	0.614	0.632	0.650	0.739	0.614	0.632	0.650
	C_{our}	0.812	0.867	0.862	0.885	0.812	0.867	0.862	0.885
	$C_{our^{\#}}$	**0.877**	**0.883**	**0.882**	**0.911**	**0.877**	**0.883**	**0.882**	**0.911**
Pyruvate oxidation Pathway	C_{SCL}	0.750	0.483	0.432	0.561	0.750	0.483	0.432	0.561
	C_{our}	0.722	**0.781**	**0.896**	**0.771**	0.722	**0.781**	**0.896**	**0.771**
	$C_{our^{\#}}$	**0.827**	0.763	0.859	0.727	**0.827**	0.763	0.859	0.727

Denote that bold values represent the maximum value in each class model of each input-target set involved in each 7-pathway example.

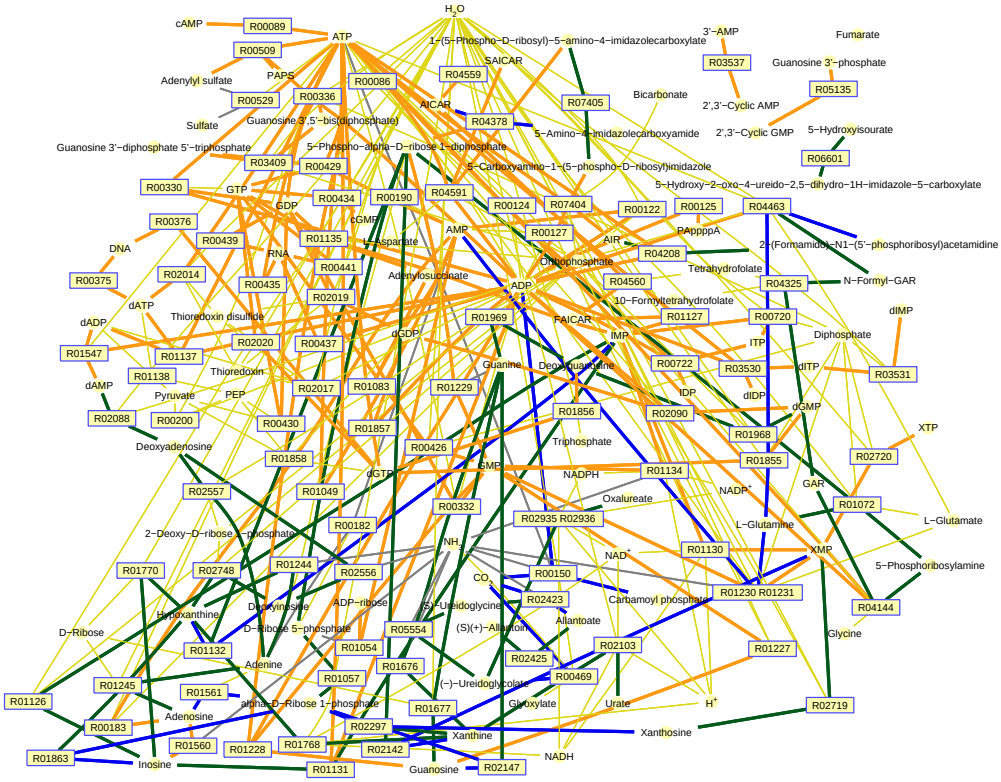
Table 4.12: Pie charts represent amount of the seen/unseen compounds participated in each metabolite query set in the *C* performance evaluation of all four defined cases in the pathway perspectives. Each class model, the output scores were obtained by SCL method, our method I, and our$^{\#}$ method I. There are 7 input sets according to 7 metabolite query sets such that each input set was evaluated by the 2 target sets of the 7-pathway examples. The two target sets for comparison were obtained by definition called metabolite transformation network and original reference maps (see Section 4.2.3.3).

The 7-pathway examples of *E.coli*	No. of compounds with available structures	No. of metabolite queries	Metabolite transformation network	Original reference maps
Purine Metabolism	96	433,200		
Valine Leucine and Isoleucine Biosynthesis	46	46,575		
Streptomycin Biosynthesis	18	2,601		
Methane Metabolism	19	3,078		
Nicotinate and Nicotinamide Metabolism	28	10,206		
Phospholipid Biosynthesis	16	1,800		
Pyruvate oxidation Pathway	10	405		

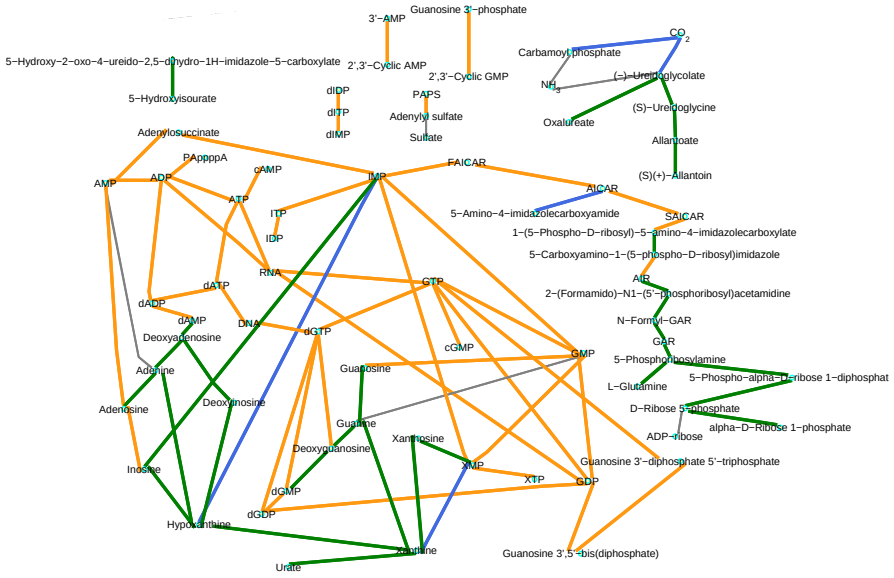
Denote that dark green, light green, dark red and light red represent amount of the positive seen metabolites, the negative seen metabolites, the positive unseen metabolites and the negative unseen metabolites, respectively (see Section 4.2.3.2 and Figure 4.18).

4.2.3.4 Traditional map visualization: comparison with original reference maps and metabolite transformation network

The comparison with references of 7-pathway examples in the form of two types of positive target maps as analyzing in Section 4.11 are visualized (Figures 4.19 to 4.25). These maps were demonstrated to depict how good both compared methods in the positive class 1 prediction are, in other words, one or two steps of compound transformation. All thick links are the combined routes that are minority(positive) target class 1 and also the results of predicted links from SCL method and our$^{\#}$ method I at cut-off value = 0.5 in comparison. The vitualization of class 1 C results are nearly same trend as AUC performance in the pathway perspective. Both compared methods can predict both same and different links. In a case of valine leucine and isoleucine biosynthesis such that protein routes are mainly trained data. All target links ($FN = 0$) of class 1 data patterns our$^{\#}$ method I at cut-off value = 0.5 can be recovered as shown in Figure 4.20.

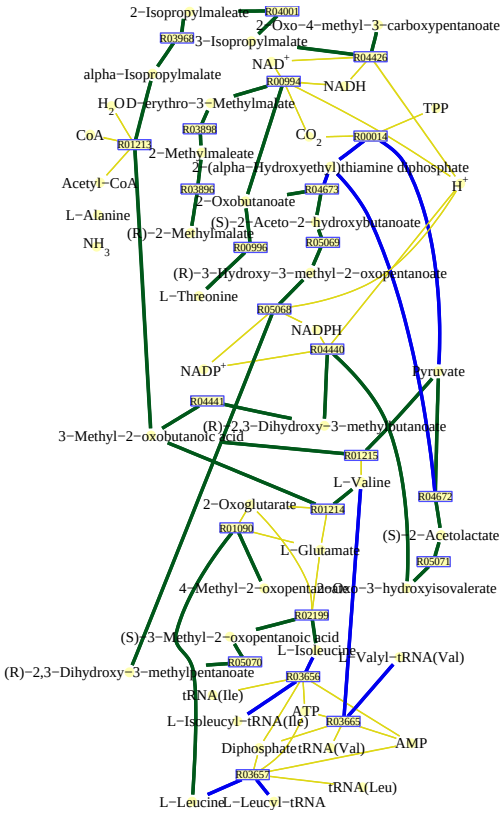


(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is an xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

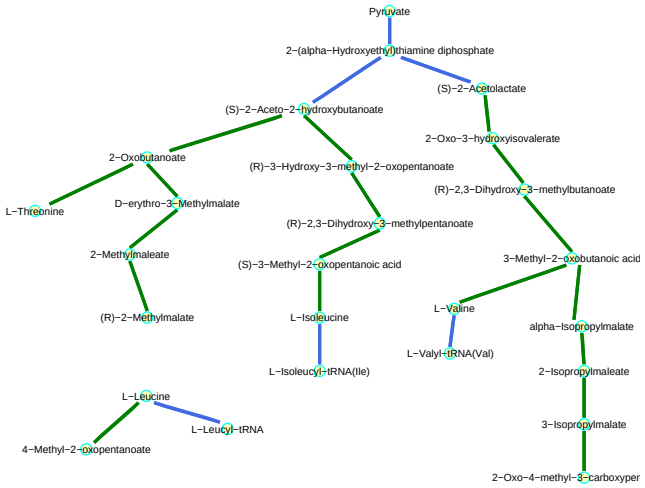


(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

Figure 4.19: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* purine metabolism. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively. In addition, the false negative links are grey.

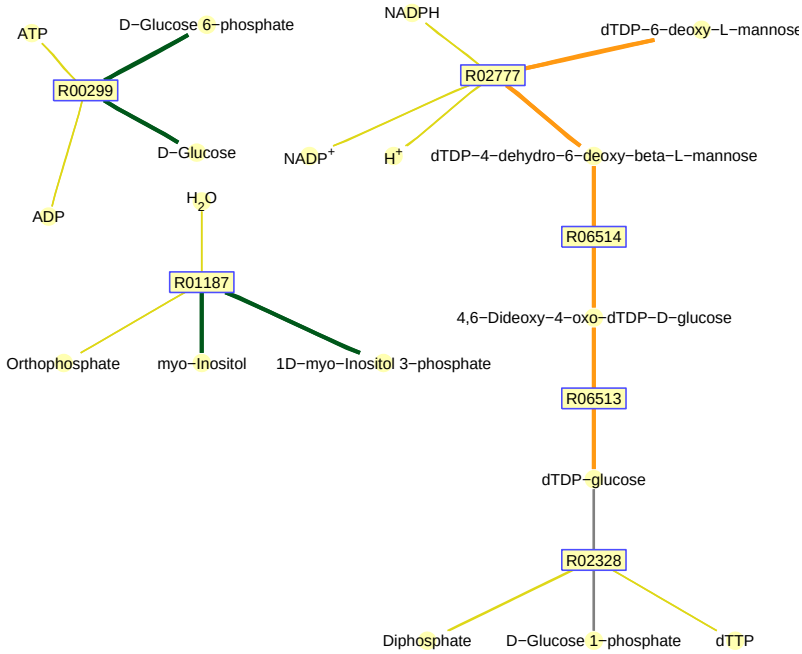


(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is an xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

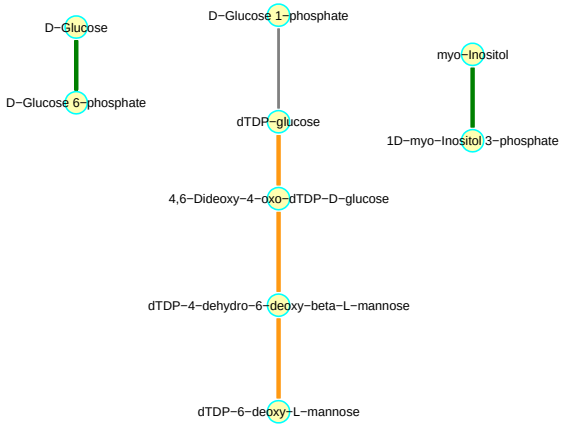


(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

Figure 4.20: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* valine leucine and isoleucine biosynthesis. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively.

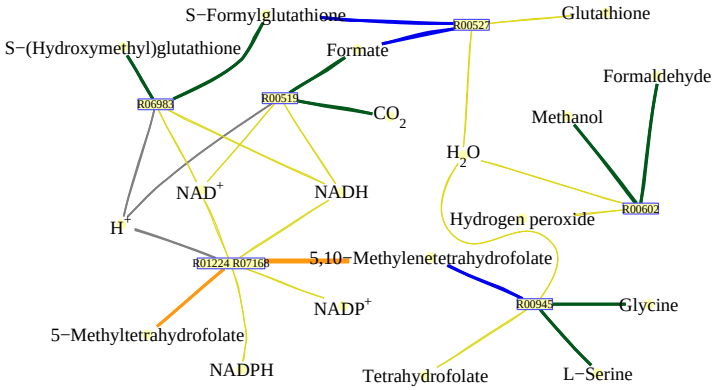


(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is an xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

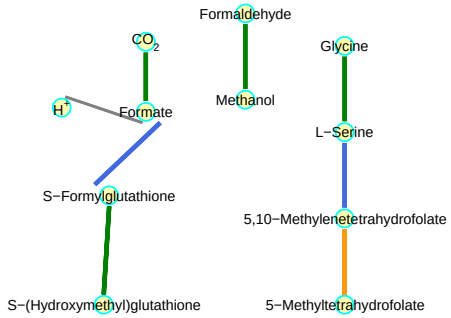


(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

Figure 4.21: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* streptomycin biosynthesis. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively. In addition, the false negative links are grey.

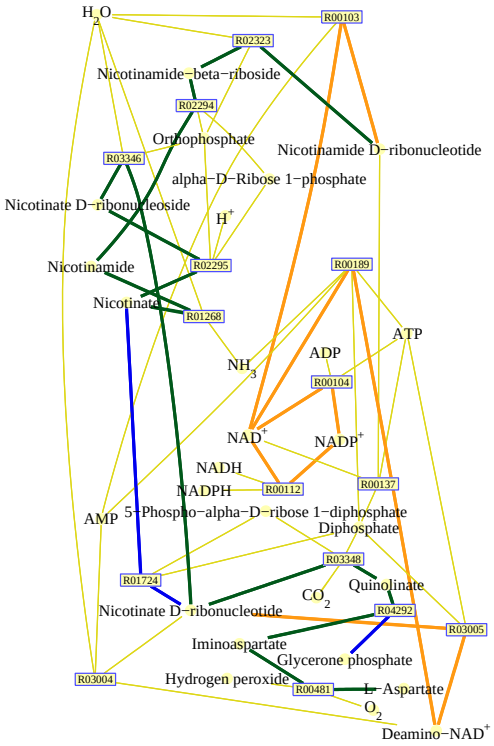


(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is an xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

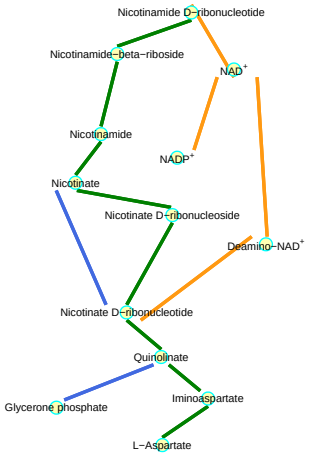


(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

Figure 4.22: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* methane metabolism. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively.

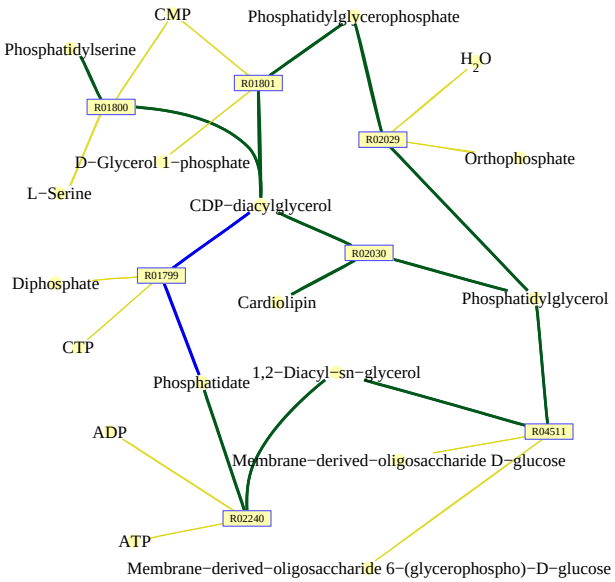


(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is an xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

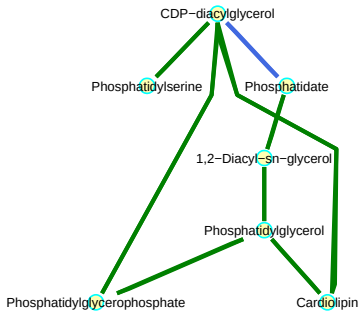


(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce xml file from KEGG pathway database as same version as KEGG ligand database used in model training.

Figure 4.23: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* nicotinate and nicotinamide metabolism. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively.

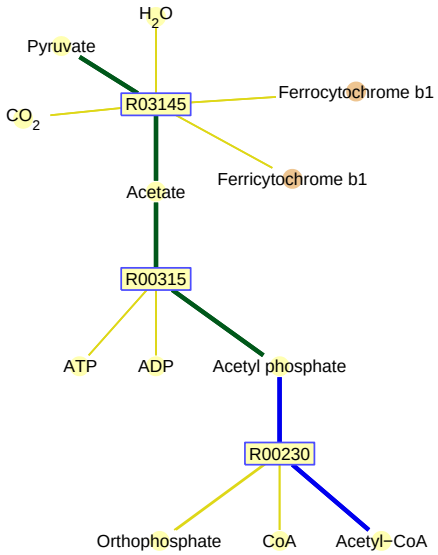


(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is a file from aMAZE database as same version as used in model training.

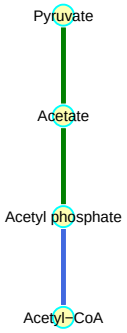


(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce file from aMAZE database as same version as used in model training.

Figure 4.24: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* phospholipid biosynthesis. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively.



(a) A traditional compound-reaction map acording to definition called metabolite transformation network including information from a referecnce map which is a file from aMAZE database as same version as used in model training.



(b) A compound-compound map with only appearing metabolites on a reference map acording to only information from a referecnce file from aMAZE database as same version as used in model training.

Figure 4.25: Comparison of class 1 *vs* non−class 1 true positive(TP) samples at cut-off value= 0.5 in the illustration as a traditional map, (a) and (b), of *E.coli* pyruvate oxidation pathway. Denote that the green, blue and orange links are true positive(TP) links predicted by both SCL method and our$^{\#}$ method I, only our$^{\#}$ method I, and only SCL method, respectively. The yellow links are drawn to fulfill a conventional compound and reaction maps. The light yellow circles and the rectangles are compounds and reactions, respectively. The dark yellow compounds have no structure data. In addition, the false negative links are grey.

CHAPTER V

DISCUSSION

5.1 Atomically convertibility of each considered metabolite input query in biochemical transformation routes from a predefined graph

When two compounds given as a beginning metabolite and a terminal metabolite or three compounds given as a beginning metabolite, an intermediate metabolite and a terminal metabolite, the following basic questions which can be answered by information obtained from the reconstructed biochemical transformation networks like metabolic networks. Whether there is a route that transforms a beginning metabolite via an intermediate metabolite (if it is given) to a terminal metabolite, where a route or a path can be simply defined as the sequences of feasible biochemical transformation steps or reactions. To discover a path, it concerns about whether there exists possible biochemical transformation steps for the interested compounds. In real life, many factors are involved in systematic and dynamic ways to transform one compound to another. But, to construct such ideal relationship of a metabolic system, it requires a vast of data with the big tasks.

To study some specific questions as the above mentioned question or discovering knowledge from metabolic networks, partial data with some necessary factors to construct question-specific network can be enough to achieve satisfied answer. One of such network is a metabolic reaction network model which is a graph-based model that a set of interested metabolites are connected by the relation of them in the feasible biochemical transformation steps as a network. The algorithm used in graph theory problems such as shortest path with some conditions caused by the predefined question is widely applied to give required information from the model. Thus, the correctness of answers depends on the accurately defined transformation steps or each relation between two metabolites as well as some defined conditions to help in the elimination of irrelevant relations. The accurately defined transformation steps are stored by many databases according the objective of each database. However, in some specific data or objective, information from databases still be lacking for reconstructing a specific graph model to answer specific question.

5.2 The four defined supervised classification problems and their corresponding binary answers

In this work, instead of another different alternative or a new graph algorithm with some conditions to answer an above mentioned question, a graph model was predefined and reconstructed (see Sections 3.1 and 3.4), then this graph is ready to convert to the supervised classification problem which is solved by supervised learning methods like the feed-forward neural network model to make the predictive model that can be predict the unseen data. In the experiment, a graph was built up from a finite set of pathways from *E.coli* based on the collected data from reliable database including the well-defined transformation steps.

An above mentioned question was divided into four specifically defined questions according to degrees and types of metabolite relation (see Section 3.2.1 and Figure 3.1). The most specific question, question 1 for the class 1 vs. non-class 1 model is about whether it is the one or two steps of consecutive reactions. Another more specific question, question 2 for the class 2 vs. non-class 2 model is about whether metabolites in a query are related in any paths but only one of the interested compound pairs defined from metabolites in a query exists one step of the biochemical transformation. The little more specific question of metabolite relation, question 3 for the class 3 vs. non-class 3 model is about whether metabolites in a query are related in any paths but none of the interested compound pairs defined from metabolites in a query exists one step of the biochemical transformation. The less specific metabolite relation question, question 4 which it implies the combination of classes 1 to 3 vs. otherwise. Later, the routine graph algorithm based on the bread-first search algorithm with necessary conditions to properly discover paths in this graph was used in order to search each answer of each metabolite query in prepared query sets from a graph for all four defined questions. An answer stands for a target class (class q or non-class q) of each question q predictive model. A metabolite query stands for a set of a beginning, an intermediate (if it is assigned), and a target metabolites used for preparing an input feature pattern. From each whole input data set and their target class set, some of them were used for building each predictive model referred to the pre-training data sets and the some of them were used for evaluating the correctness of the built predictive models referred to the testing data sets. Chemical properties e.g. molecular properties of each metabolite in a metabolite query from its calculated optimized 3D structure by using 2D coordinate were indirectly used as an input feature pattern, because the calculated molecular properties for each metabolite query were transformed by numerical methods to represent new characteristics as an

input feature pattern with the lesser number of properties. Henceforth, the pre-training data sets could be used for the model building.

5.3 The algorithm to handle the seriously imbalanced binary class data before constructing four feed-forward neural network predictive models

Unfortunately, the nature of the metabolite transformation network graph including the defined metabolite query sets results in the unequal proportion of the binary class data. When observing each binary class data distribution and the size before dividing the pre-training and the testing data sets for each model building, they are huge and very unbalancing. The too big training data set may cause time-consuming in training the feed-forward neural network models. It would take a long time until reaching the the sum square error threshold or desired number of epochs. So, the whole data was divided by using a key feature into more adequate size and easy to be trained by the neural network method with suitable parameter values in practical time as each the predictive sub-model which they was finally combined. Besides, the sub-models can be trained at the same time. In an imbalanced data situation, it is commonly known that impacts the typical sum square error in the weight updating procedure of the feed-forward neural network. Therefore, the nearly balanced training data sets created by using the pre-training data sets with the proposed methods were alternatively trained.

The proposed methods rely on two main criteria for managing and fixing the imbalanced data for each predictive model building. First, sub-groups of each binary class data was discovered by clustering methods in order to help to handle the complex data space that the binary class data could be hard to be separated by activation function of each hidden unit in the feed-forward neural network algorithm. Second, due to the high unbalancing ratio of data, the minority class data are very small comparing with the majority class data. Hence, there are three procedures to handle the imbalanced data and turn it into the nearly balanced data. First procedure (if it is necessary) before doing each class data clustering process, at most new k data positioned between each minority data and every k minority neighbor data are created by the proposed rules in order to oversample the minority data as well as expand the minority class data space, appropriately. Second procedure, additional created data from resampling method are combined to each sub-data group before finding two border data sets of each sub-cluster pair with each one belonging to different binary class. This procedure aims to not only find the standard deviation difference of before and after adding the created data as the suitably expand-

ing breadth for more data generation in the next step but also provide the possibility that the created data become one point in border data sets which infers that such new point locates new possible space beneficial to model building for unseen data prediction. Third procedure, two border data sets are found for each pair of minority and majority sub-data groups in a term of under sampling. Then, they are used as initial data for synthetizing more data into the nearly balance data with sufficient size in a term of over sampling.

To explore the general ability of the new characteristics as input feature patterns with the defined targets trained by the feed-forward neural network methods in all four defined problems, all trained sub-models with adequate parameter values were sought out in the limited rounds of sub-model building with the acceptable evaluation values from some metrics applied to the testing data sets, the nearly balanced training data sets, and also the pre-training data set so that each metric result value can be comparatively explored.

In searching appropriate parameter values, even though the experiments to observe effects of difference of them in detail were not formed, it may be useful to discuss some issues. First, the numbers of neurons in 2D SOM method should be enough to roughly cluster each binary class data, because insufficient numbers of neurons lead to bad capturing local groups resulting in unsuccessful predictive model built by the nearly balanced training data sets. However, too many numbers of neuron in 2D SOM methods cause SOM process to slowly reach the desired stopping criteria and also make the lack of information in minority sub-data with very tiny size. Additionally, the big occurring clusters according to the class distribution nature of that sub-data bring about the large minority-majority sub-data pairs with the large numbers of border data sets which produce the huge size of the corresponding nearly balanced training data set. Second, the new generated data according to the *k* nearest minority data in each minority data would be unnecessary if the data space is not too complicated and/or not too less informative to cluster each binary class into sub-groups using 2D SOM method. Besides, the obtaining sub-groups still provide needed information in a term of suitable extended minority data space in order to effectively generate the corresponding nearly balanced training data set for successful sub-model building. The last one, the numbers of hidden neurons in each feed-forward neural network model were systemically varied. Since the difference in high dimensional sub-data location and space, the feed-forward neural network models with various numbers of hidden neurons were simultaneously performed.

5.4 The experimental outcomes and comparison

The comparative study is to compare the strength of chemical linkage (SCL) values with our output values as various score types i.e. our/our$^{\#}$ result I/ II in classifying binary class data. Our$^{\#}$ output values are the output values from each our predictive model, but the pre-process of SCL which the obviously irrelevant metabolite queries are set to zero. Also, the corresponding output values are filtered to zero for fair comparison. Apart from that, there are two types of the score results, namely, the model result I which all output values from every selected sub-model built for that single question are combined and the model result II which four output values associated with one input pattern are average from all four predictive combined sub-model. The results of these in detail are in the previous section. In this section, the crucial general findings were discussed.

First, the appropriate numbers of sub-models building in the sub-data division step depends on training time and the desired performance of the built sub-models for each particular defined question. If it is too big, then it would take a long time and the chance of the missing global information. But, if it is too small, then some sub-data with so complicated space would take too long time to yield good performance. In this work, the total 6 pre-training sub-data were appropriate.

Second, at the small(0.3), medium(0.5) and large(0.8) including default(0.4) cut-off values, their performance results in all eight evaluation metrics were observed and the pair *t*-test significance test at 0.05 degree of significance was also performed. In all cut-off values, the common performance results of the models from our/our$^{\#}$ score I/ II tend to reverse the degree of the imbalanced data situation except those of the models from SCL scores. The least to the most specific questions for asking route-relevance of each metabolite query are the questions 4,3,2, and 1 with the least to the most degrees of the imbalanced sub-data sets. The different cut-off values effect the performance results. The very effective general performance results of four models were received by our$^{\#}$ score I/ II at cut-off values = 0.3 and 0.4. The most effective SCL scores appeared at cut-off value = 0.5, but they did not significantly outperformed in every metric. These above results tell us that the proposed input feature patterns in a form of the sub-space data can be classified by the supervised feed-forward neural network techniques. The more improved output scores is our $^{\#}$ score which combined the good point of SCL calculation such that our $^{\#}$ score are set to zero same as SCL score if an input pattern query contains

no common compound alignments. In addition, our # result I and II showed the comparable performance noting that the significance test for the result pairs of them was not computed.

Third, from the efficient performance results of our # score I and II, more two tasks of comparison were done. One is non-cut-off value measurement. *AUC* values of our # score II were calculated and compared with those of SCL scores for predicting the new data from the 7-pathway examples for the four defined questions in the form of each pathway evaluation and each sub-model evaluation. The results showed that to predict the answers of questions about route relevance of a metabolite input query set according to the sub-model perspective as the proposed method or the traditional pathway perspective, both our # method II and SCL method contain its different advantages. However, the proposed method is more flexible, theoretically, since the size of sub-data for training sub-model in a defined question can be tuned to increase a chance of yielding the satisfied performance results.

Another one is the correctness measurement at a chosen cut-off value. First experiment, each metabolite input query set which one set is associated with each distinct compound from the 7-pathway examples was prepared. *C* values of our score I and our # score I were calculated and compared with those of SCL scores. This experiment explored the correctness of the compared methods in comparison for predicting each metabolite input query set when one particular compound exists in every query of a whole set, in other words, the correctness in the compound perspective was measured. This tells us that which one of the compared methods is adequate for predicting an interested question about relevant route associated with a certain compound. The results depict that some compounds can be accurately predicted the related general paths by one of the compared methods in the different questions about route-relevance. There are compounds that their high *C* values in the various questions can be received by our # method I, but in the large numbers of compounds, their results suggest that the our # score I and SCL scores are competitive when predicting metabolite queries of the seen compounds that are involved in the trained data. In the rare pattern input cases of the trained data, unsuccessful prediction of the unseen data with the same rare cases were shown by our # method. However, they were unsuccessful SCL prediction in some sets like lipid routes and compounds carrying formyl/acetyl groups, our # method can effectively predicted them. Besides, results of our # method seem to improve results of our method excepting in some small amount of sets. Another experiment, the 7 metabolite input query sets and their input data sets were prepared as same as the non-cut-off value measurement, so this is the correctness evaluation in the pathway

perspective. But, there are two target types that were prepared by original reference maps and definition called metabolite transformation network for comparison. Generally, for both target types, some pathways can be predicted with high *C* results from different compared methods.

From both experiments with the correctness measurement at a chosen cut-off value as well as the non-cut-off value measurement, the necessary training data patterns should exist for building supervised models with the desired abilities.

5.5 Using the four predictive models in unseen data prediction

Various experiments were represented in Section 4.2, however, they were illustrated in each binary class prediction. So, our result I and our$^{\#}$ result I earned by trained models of the nearly balanced training data were depicted in a form of the 4×4 confusion matrices as the combined four class prediction (appendix A) in the 7-pathways examples. Besides, the 4×4 confusion matrices of the 7 additional other pathways (Table A.15) concerning metabolism of terpenoids and polyketides, metabolism of other amino acids, biosynthesis of other secondary metabolites, and xenobiotics biodegradation and metabolism were also provided. All of them were considered as the unseen data in the pathway types/roles that never involved in the trained data. Each value at position (i, j) in confusion matrix is amount of data patterns with their target class i such that a model j yields maximum output value and predict them as class j. Denote that there are two types of targets obtained by definition, metabolite transformation network, and original reference maps from KEGG pathway database as the same version as the trained data. Confusion matrices show overall results of model prediction, since amount of true positive patterns of all patterns in a set across four models are directly displayed. In each testing data where their metabolite queries associated with a single pathway, the small size of the testing data seems to have identical two types of targets and predicted outcomes. In addition, the amount of positive target class 2 is more than those of positive target class 3. Confusion matrices of our$^{\#}$ result I were always improved from those of our result I when considering misclassified fraction of patterns (confusion value). Because adding information of irrelevant 2D structures alignment of metabolite input queries can reduce misclassified amount in positive class 4. Focusing on overall confusion matrices of the 7 additional other pathways, the combined predictive models captured some simple in-route relations in a case of one or two consecutive steps (positive class 1). These indicate that, in a case of the rare trained pattern types such as the pathway types/roles that never involved in the trained data, the simplest unseen

patterns of in-route relations were able to detected by the trained models. Moreover, the large amount of correctly classified majority class like positive class 4 came from our# result I. The positive data patterns in classes 2 and 3 were hardly accurately predicted due to complication of unseen in-route relation and the rare trained pattern types. In conclusion, these confusion matrices tell us that decreasing misclassified outcomes still be further challenged. First, the process of the nearly balanced training data preparation should be further developed to make the effective class separation of neural network models. Second, other features, especially, 2D structure similarity still be necessary and should be added. Otherwise features should be further considered, since the large amount of features would take a long time in model training process.

5.6 Combined four predictive models versus each predictive binary class model in unseen data prediction

In combined four predictive models from four binary predictive models, there are more than one class models that predicted positive outcomes for one pattern. This situation occurred in not small amount of patterns when predicting only one class for one pattern (combined four predictive models). These should be explored in many viewpoints. First view point is the shortest path criteria in assigning one of four defined class targets when given one considered metabolite query. The shortest path concept is widely used in the previous works of path searching in metabolic pathways, since it is easy to cope with a simple graph model of metabolic networks and the existing graph algorithms. Furthermore, every metabolite query from a considered metabolite transformation network can be categorized into one class based on our four defined classes. However, the total routes in the form of metabolite transformation networks are beyond just the combination of multiple shortest steps into a network (Figure 5.1). One weak point of shortest path concept in the proposed definitions and other previous works is the lack of information about alternative routes which may be necessary, especially, in the new pathway design for metabolic engineering applications. So, some previous work based on graph algorithms tried to extend shortest path conditions by gathering shortest paths not exceeding a setting step/weight (Faust et al., 2011). Therefore, the extension of metabolite query can be designed for further handling this issue. Anyway, the effective binary class models can preliminary offer class target in each binary class prediction for each predicted pattern. Second view point is the combining four binary class models into one multi-class predictive model. The simplest output combining scheme were used such as a maximum output value becomes the predicted class outcome for such considered input pattern associated with a metabolite query. Apart from

discussion in Section 5.5, the effective output scheme for multiple binary class combination still be challenged (Galar et al., 2011) and should be further improved.

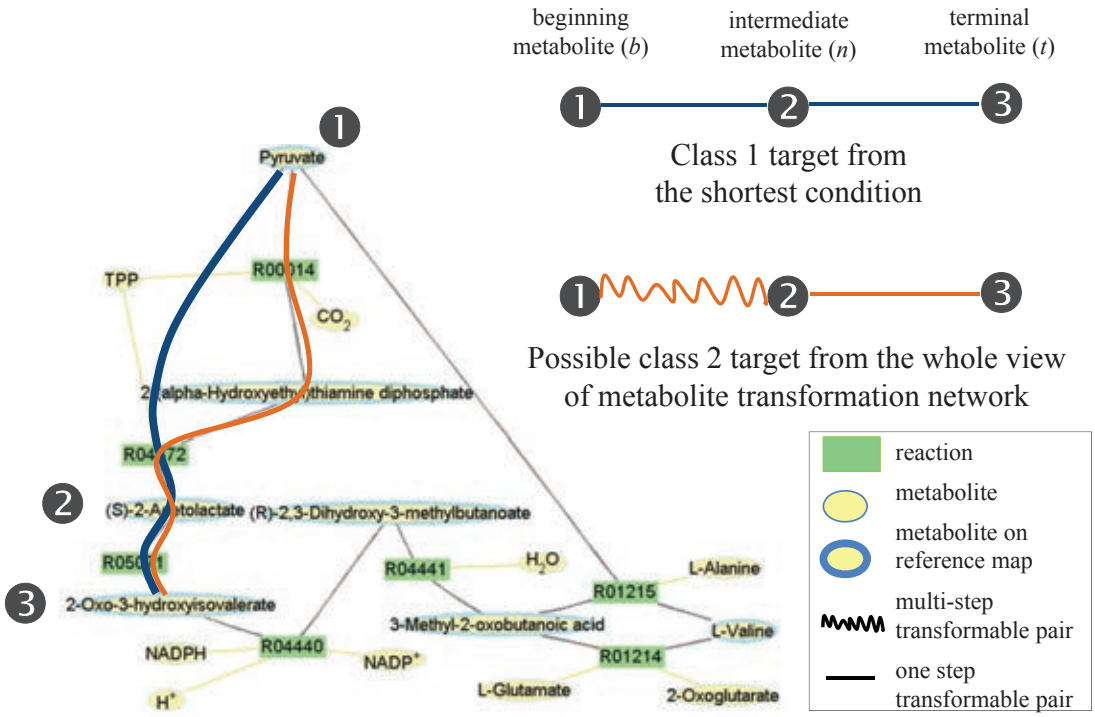


Figure 5.1: An example of the shortest path criteria issue as discussed in Section 5.6. A graph was defined connection by metabolite transformation network (Section 3.1). These example data were from a part of eco00290.xml in KEGG Pathway Database(14-7-2010). Denote that the grey links are the main routes. An example of a metabolite input query (1,2,3) : (Pyruvate,(S)-2-Acetolactate,2-Oxo-3-hydroxyisovalerate) with class target 1 according to a shortest path condition is shown. A four bit vector of Our$^{\#}$ result I at cut-off value= 0.5 from four class models is $(1,0,0,1)$. In addition, almost predicted outcomes for metabolite queries in this picture mostly yielded more than one class prediction at cut-off value= 0.5.

5.7 Other discussion

Other aspects bring to be discussed. The common benefit of supervised learning algorithms is the flexibility to build model in any defined questions(input) and answers(class). In the problem of identifying the type of relevance in the routes of each interested metabolite query. Instead of these four questions, it can be variously defined. For instance, a metabolite query consists of only in-route metabolites. Its input feature pattern can be any finite sets of properties that considerably contain the association with its defined target class. However, if the ratio of class distribution in the training data is imbalanced, then, the pre-process should be added to fix it to balance the training data. In addition, although it is time-consuming in the hard separating data, the sub-data division method will help to simplify the complex location of the huge data. Aside from that, there are many existing ideas to further improve predictive performance, for example, the committee scheme training processes.

CHAPTER VI

CONCLUSION

To discover the meaningful paths from the metabolic reaction network model which is a graph-based model, the partial data with some necessary factors to construct question-specific network can be enough to achieve satisfied answer. In this work, the supervised learning schemes such as the feed-forward neural network were alternatively offered in order to construct the predictive models learned by prior reference data. Initially, a graph model was predefined and reconstructed, then it was demonstrated by an *E.coli* finite pathway set with the well-defined transformation steps from the reliable databases. Later, this graph was ready to convert to the supervised classification problems which were solved by the feed-forward neural network model to make the predictive models that can predict the unseen data.

These above results tell us that the nearly balanced training data from the proposed input feature patterns can be satisfactorily classified by supervised feed-forward neural network techniques. Especially, the pre-training data contain enough necessary information like in the defined questions 2 to 4 model building which they are asking about metabolite relation beyond two step changing. In case of no more than two step metabolite transformation (a defined question 1) model building, the enough necessary information of positive class is further required in order to obviously obtain superior performance results from the proposed methods to those from SCL method. The numerically transformed input feature patterns resulting from the computed $3D$ molecular properties of every metabolite in each considered query are suitable for training their binary classes of transformation by supervised learning methods if the binary proportion is quite equal. Apart from that, the $2D$ co-ordinate compound alignment as the useful output filter from SCL method, such as our $^{\#}$ scores, is the reasonable combination which additionally helps to yield the better performance results. Moreover, the input patterns as the enough representatives for the whole considered routes of compound types are crucial for the effectiveness of such trained models.

References

Adam, A., et al. (2010). A modified artificial neural network learning algorithm for imbalanced data set problem. In Second International Conference on Computational Intelligence, Communication Systems and Networks(CICSyN'10) , pp. 44–48.

Aittokallio, T. and Schwikowski, B. (2006). Graph-based methods for analysing networks in cell biology. Briefings in Bioinformatics, 7(3): 243–255.

Akutsu, T. (2004). Efficient extraction of mapping rules of atoms from enzymatic reaction data. Journal of Computational Biology, 11(2-3): 449–462.

Almeida, J. S. and Voit, E. O. (2003). Neural-network-based parameter estimation in S-system models of biological networks. Genome Informatics, 14: 114–123.

Arita, M. (2000). Metabolic reconstruction using shortest paths. Simulation Practice and Theory, 8: 109–125.

Arita, M. (2004). The metabolic world of Escherichia coli is not small. Proceedings of the National Academy of Sciences of the United States of America, 101(6): 1543–1547.

Barua, S., Islam, M., and Murase, K. (2011). A novel synthetic minority oversampling technique for imbalanced data set learning. In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), Neural Information Processing, volume 7063 of Lecture Notes in Computer Science , pp. 735–744. Berlin, Heidelberg: Springer-Verlag.

Baths, V., Roy, U., and Singh, T. (2011). Disruption of cell wall fatty acid biosynthesis in Mycobacterium tuberculosis using a graph theoretic approach. Theoretical Biology and Medical Modelling, 8(1): 5.

Batuwita, R. and Palade, V. (2009). A new performance measure for class imbalance learning. application to bioinformatics problems. In International Conference on Machine Learning and Applications(ICMLA '09) , pp. 545–550.

Batuwita, R. and Palade, V. (2010). FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Transactions on Fuzzy Systems, 18(3): 558–571.

Bishop, C.M. (2006). Pattern recognition and machine learning. USA: Springer New York.

Blum, T. and Kohlbacher, O. (2008a). MetaRoute: fast search for relevant metabolic routes for interactive network navigation and visualization. Bioinformatics, 24(18): 2108–2109.

Blum, T. and Kohlbacher, O. (2008b). Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. Journal of Computational Biology, 15(6): 565–576.

Browne, A., Hudson, B. D., Whitley, D. C., Ford, M. G., and Picton, P. (2004). Biological data mining with neural networks: implementation and application of a flexible decision tree extraction algorithm to genomic problem domains. Neurocomputing, 57: 275 – 293.

Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2012). DBSMOTE: density-based synthetic minority over-sampling technique. Applied Intelligence, 36(3): 664–684.

Caspi, R., et al. (2010). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Research, 38(suppl 1): D473–D479.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16(1): 321–357.

Chen, S., He, H., and Garcia, E. (2010). RAMOBoost: ranked minority oversampling in boosting. IEEE Transactions on Neural Networks, 21(10): 1624–1642.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). Introduction to Algorithms. USA: MIT Press, 2nd edition.

Cottret, L. and Jourdan, F. (2010). Graph methods for the investigation of metabolic networks in parasitology. Parasitology, 137: 1393–1407.

Crabtree, J. D. and Mehta, D. P. (2009). Automated reaction mapping. ACM Journal on Experimental Algorithmics, 13: 15:1.15–15:1.29.

Croes, D., Couche, F., Wodak, S., and van Helden, J. (2006). Inferring meaningful pathways in weighted metabolic networks. Journal of Molecular Biology, 356: 222–236.

Croes, D., Couche, F., Wodak, S. J., and van Helden, J. (2005). Metabolic pathfinding: inferring relevant pathways in biochemical networks. Nucleic Acids Research, 33(suppl 2): W326–W330.

de Figueiredo, L. F., et al. (2009). Computing the shortest elementary flux modes in genome-scale metabolic networks. Bioinformatics, 25(23): 3158–3165.

Deville, Y., Gilbert, D., van Helden, J., and Wodak, S. J. (2003). An overview of data models for the analysis of biochemical pathways. Briefings in Bioinformatics, 4(3): 246–259.

Ertekin, S., Huang, J., and Giles, C. L. (2007). Active learning for class imbalance problem. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval(SIGIR '07) , pp. 823–824.

Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. Computational Intelligence, 20(1): 18–36.

Faust, K., Croes, D., and van Helden, J. (2009). Metabolic pathfinding using RPAIR annotation. Journal of Molecular Biology, 388(2): 390 – 414.

Faust, K., Croes, D., and van Helden, J. (2011). Prediction of metabolic pathways from genome-scale metabolic networks. Biosystems, 105(2): 109 – 121.

Faust, K., Dupont, P., Callut, J., and van Helden, J. (2010). Pathway discovery in metabolic networks by subgraph extraction. Bioinformatics, 26(9): 1211–1218.

Félix, L. and Valiente, G. (2007). Validation of metabolic pathway databases based on chemical substructure search. Biomolecular Engineering, 24(3): 327 – 335.

Finley, S. D., Broadbelt, L. J., and Hatzimanikatis, V. (2009). Computational framework for predictive biodegradation. Biotechnology and Bioengineering, 104(6): 1086–1097.

Fu, X., Wang, L., Chua, K. S., and Chu, F. (2002). Training RBF neural networks on unbalanced data. In Proceedings of the 9th International Conference on Neural Information Processing(ICONIP'02), volume 2 , pp. 1016–1020.

Galar, M., Fernndez, A., Barrenechea, E., Bustince, H., and Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition, 44(8): 1761 – 1776.

Gibbons, J. and Chakraborti, S. (2003). Nonparametric Statistical Inference, Fourth Edition: Revised and Expanded. Statistics: A Series of Textbooks and Monographs. USA: Marcel Dekker.

Goto, S., Okuno, Y., Hattori, M., Nishioka, T., and Kanehisa, M. (2002). LIGAND: database of chemical compounds and reactions in biological pathways. Nucleic Acids Research, 30(1): 402–404.

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. In Fourth International Conference on Natural Computation (ICNC '08), volume 4 , pp. 192–201.

Hattori, M., Tanaka, N., Kanehisa, M., and Goto, S. (2010). SIMCOMP/SUBCOMP: chemical structure search servers for network analyses. Nucleic Acids Research, 38(suppl 2): W652–W656.

Hatzimanikatis, V., Li, C., Ionita, J. A., and Broadbelt, L. J. (2004). Metabolic networks: enzyme function and metabolite structure. Current Opinion in Structural Biology, 14(3): 300 – 306.

Haykin, S. (1998). Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd edition.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9): 1263–1284.

Heath, A. P., Bennett, G. N., and Kavraki, L. E. (2010). Finding metabolic pathways using atom tracking. Bioinformatics, 26(12): 1548–1555.

Heath, A. P., Bennett, G. N., and Kavraki, L. E. (2011). An algorithm for efficient identification of branched metabolic pathways. Journal of Computational Biology, 18(11): 1575–1597.

Heinonen, M., Lappalainen, S., Mielikäinen, T., and Rousu, J. (2011). Computing atom mappings for biochemical reactions without subgraph isomorphism. Journal of Computational Biology, 18(1): 43–58.

Hong, X., Chen, S., and Harris, C. (2007). A kernel-based two-class classifier for imbalanced data sets. IEEE Transactions on Neural Networks, 18(1): 28–41.

Hou, B., Ellis, L., and Wackett, L. P. (2004). Encoding microbial metabolic logic: predicting biodegradation. Journal of Industrial Microbiology and Biotechnology, 31(6): 261–272.

Huss, M. and Holme, P. (2007). Currency and commodity metabolites: their identification and relation to the modularity of metabolic networks. IET Systems Biology, 1(5): 280–285.

Jeatrakul, P. and Wong, K.-W. (2012). Enhancing classification performance of multi-class imbalanced data using the OAA-DB algorithm. In The International Joint Conference on Neural Networks(IJCNN '12) , pp. 1–8.

JOELib (2004). JOELib 2004-08-27. http://sourceforge.net/projects/joelib/. Accessed: 2011-08-28.

Kaleta, C., de Figueiredo, L. F., and Schuster, S. (2009). Can the whole be less than the sum of its parts? pathway analysis in genome-scale metabolic networks using elementary flux patterns. Genome Research, 19(10): 1872–1883.

Kanehisa, M., et al. (2008). KEGG for linking genomes to life and the environment. Nucleic Acids Research, 36: D480–D484.

Klamt, S. and Stelling, J. (2003). Two approaches for metabolic pathway analysis? Trends in Biotechnology, 21(2): 64 – 69.

Kotera, M., et al. (2004). RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. Genome Informatics, 15(2): P062.

Kotera, M., Tabei, Y., Yamanishi, Y., Tokimatsu, T., and Goto, S. (2013). Supervised de novo reconstruction of metabolic pathways from metabolome-scale compound sets. Bioinformatics, 29(13): i135–i144.

Kraipeerapun, P. and Fung, C. C. (2009). Binary classification using ensemble neural networks and interval neutrosophic sets. Neurocomputing, 72(13-15): 2845 – 2856.

Küffner, R., Zimmer, R., and Lengauer, T. (2000). Pathway analysis in metabolic databases via differential metabolic display (DMD). Bioinformatics, 16(9): 825–836.

Lacroix, V., Cottret, L., Thebault, P., and Sagot, M. (2008). An introduction to metabolic networks and their structural analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 5(4): 594–617.

Le, S. Q., Ho, T. B., and Phan, T. T. H. (2004). A novel graph-based similarity measure for 2D chemical structures. Genome Informatics, 15(2): 82–91.

Lemer, C., et al. (2004). The aMAZE LightBench: a web interface to a relational database of cellular processes. Nucleic Acids Research, 32(Database-Issue): 443–448.

Li, C., et al. (2004). Computational discovery of biochemical routes to specialty chemicals. Chemical Engineering Science, 59(22-23): 5051 – 5060.

Liu, J.-F. and Yu, D.-R. (2007). A weighted rough set method to address the class imbalance problem. In International Conference on Machine Learning and Cybernetics, volume 7 , pp. 3693–3698.

Liu, X.-Y., Wu, J., and Zhou, Z.-H. (2006). Exploratory under-sampling for class-imbalance learning. In Sixth International Conference on Data Mining(ICDM '06) , pp. 965–969.

Ma, H. and Zeng, A.-P. (2003). Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. Bioinformatics, 19(2): 270–277.

Marvin (2011). Molconvertor in marvin 5.5.1. http://www.chemaxon.com/products/marvin/molconverter/. Accessed: 2011-12-09.

McClymont, K. and Soyer, O. S. (2013). Metabolic tinker: an online tool for guiding the design of synthetic metabolic pathways. Nucleic Acids Research, 41(11): e113.

Mithani, A., Preston, G. M., and Hein, J. (2009). Rahnuma: hypergraph-based tool for metabolic pathway prediction and network comparison. Bioinformatics, 25(14): 1831–1832.

Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, 6(4): 525–533.

Mu, F., Unkefer, C. J., Unkefer, P. J., and Hlavacek, W. S. (2011). Prediction of metabolic reactions based on atomic and molecular properties of small-molecule compounds. Bioinformatics, 27(11): 1537–1545.

Oberhardt, M. A., Chavali, A. K., and Papin, J. A. (2009). Flux balance analysis: interrogating genome-scale metabolic networks. In I. V. Maly (Ed.), Systems Biology, volume 500 of Methods in Molecular Biology , pp. 61–80. Humana Press.

Pey, J., Prada, J., Beasley, J., and Planes, F. (2011). Path finding methods accounting for stoichiometry in metabolic networks. Genome Biology, 12(5): R49.

Pitkänen, E., Jouhten, P., and Rousu, J. (2009). Inferring branching pathways in genome-scale metabolic networks. BMC Systems Biology, 3(1): 103.

Pitkänen, E., Rantanen, A., Rousu, J., and Ukkonen, E. (2005). Finding feasible pathways in metabolic networks. In P. Bozanis & E. Houstis (Eds.), Advances in Informatics, volume 3746 of Lecture Notes in Computer Science , pp. 123–133. Springer Berlin Heidelberg.

Pitkänen, E., Rousu, J., and Ukkonen, E. (2010). Computational methods for metabolic reconstruction. Current Opinion in Biotechnology, 21: 70 – 77.

Plaimas, K., Lursinsap, C., and Suratanee, A. (2005). High performance of artificial neural network for resolving ambiguous nucleotide problem. In Proceedings 19th IEEE International Parallel and Distributed Processing Symposium(IPDPS 2005) , pp. 7pages.

Planes, F. and Beasley, J. (2008). A critical examination of stoichiometric and path-finding approaches to metabolic pathways. Briefings in Bioinformatics, 9(5): 422–436.

Rahman, S. A., Advani, P., Schunk, R., Schrader, R., and Schomburg, D. (2005). Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). Bioinformatics, 21(7): 1189–1193.

Raman, K. and Chandra, N. (2009). Flux balance analysis of biological systems: applications and challenges. Briefings in Bioinformatics, 10(4): 435–449.

Rantanen, A., et al. (2008). An analytic and systematic framework for estimating metabolic flux ratios from 13 C tracer experiments. BMC Bioinformatics, 9: 266.

Raymond, J. W., Gardiner, E. J., and Willett, P. (2002). Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm. Journal of Chemical Information and Computer Sciences, 42(2): 305–316.

Raymond, J. W. and Willett, P. (2002). Maximum common subgraph isomorphism algorithms for the matching of chemical structures. Journal of Computer-Aided Molecular Design, 16: 2002.

Schilling, C. H. and Palsson, B. O. (1998). The underlying pathway structure of biochemical reaction networks. Proceedings of the National Academy of Sciences, 95(8): 4193–4198.

Schuster, S., Fell, D. A., and Dandekar, T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. Nature Biotechnology, 18(3): 326–332.

Steinbeck, C., et al. (2003). The Chemistry Development Kit (CDK): an open-source java library for chemo- and bioinformatics. Journal of Chemical Information and Computer Sciences, 43(2): 493–500.

Stewart, J. J. P. (2009). Mopac2009. `http://openmopac.net/`. Accessed: 2011-09-04.

Sun, Y., Wong, A. K. C., and Kamel, M. S. (2009). Classification of imbalanced data: a review. International Journal of Pattern Recognition and Artificial Intelligence, 23(04): 687–719.

Thanathamathee, P. and Lursinsap, C. (2013). Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and AdaBoost techniques. Pattern Recognition Letters, 34(12): 1339–1347.

van Helden, J., Wernisch, L., Gilbert, D., and Wodak, S. J. (2002). Graph-based analysis of metabolic networks. In H.-W. Mewes, H. Seidel, & B. Weiss (Eds.), Bioinformatics and Genome Analysis, volume 38 of Ernst Schering Research Foundation Workshop , pp. 245–274. Springer Berlin Heidelberg.

Visa, S. (2005). Issues in mining imbalanced data sets - a review paper. In Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference , pp. 67–73.

Walpole, R. E., Myers, R. H., Myers, S. L., and Ye, K. (2011). Probability and Statistics for Engineers and Scientists. USA: Pearson Education, 9th edition.

Wood, M. J. and Hirst, J. D. (2005). Recent applications of neural networks in bioinformatics. In B. Apolloni, M. Marinaro, & R. Tagliaferri (Eds.), Biological and Artificial Intelligence Environments , pp. 91–97. Springer Netherlands.

Xu, Y., et al. (2002). Artificial neural networks and gene filtering distinguish between global gene expression profiles of barrett's esophagus and esophageal cancer. Cancer Research, 62(12): 3493–3497.

Yan, R., Liu, Y., Jin, R., and Hauptmann, A. (2003). On predicting rare classes with SVM ensembles in scene classification. In Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03), volume 3 , pp. III–21–4 vol.3.

Zhang, G. (2000). Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 30(4): 451–462.

Zhao, J., Yu, H., Luo, J., Cao, Z., and Li, Y. (2006). Complex networks theory for analyzing metabolic networks. Chinese Science Bulletin, 51(13): 1529–1537.

Zheng, Z., Wu, X., and Srihari, R. (2004). Feature selection for text categorization on imbalanced data. ACM SIGKDD Exploration Newsletter, 6(1): 80–89.

Zhou, W. and Nakhleh, L. (2011). The strength of chemical linkage as a criterion for pruning metabolic graphs. Bioinformatics, 27(14): 1957–1963.

Appendix

Appendix A

CONFUSION MATRICES OF UNSEEN DATA PREDICTION

Table A.1: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Purine metabolism in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	56	26	42	1,111	1,235
Positive class 2	893	486	801	13,915	16,095
Positive class 3	4,561	3,252	5,883	83,871	97,567
Positive class 4	18,535	14,245	19,811	265,712	318,303
Total	24,045	18,009	26,537	364,609	433,200

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.37.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	28	14	23	226	291
Positive class 2	460	234	407	6,132	7,233
Positive class 3	2,738	1,982	3,824	55,191	63,735
Positive class 4	20,819	15,779	22,283	303,060	361,941
Total	24,045	18,009	26,537	364,609	433,200

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.29.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	45	13	27	1,150	1,235
Positive class 2	750	417	688	14,240	16,095
Positive class 3	4,203	2,982	5,567	84,815	97,567
Positive class 4	12,312	10,594	12,845	282,552	318,303
Total	17,310	14,006	19,127	382,757	433,200

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.33.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	27	12	22	230	291
Positive class 2	455	220	385	6,173	7,233
Positive class 3	2,720	1,883	3,760	55,372	63,735
Positive class 4	14,108	11,891	14,960	320,982	361,941
Total	17,310	14,006	19,127	382,757	433,200

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.25.

Table A.2: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Valine leucine and isoleucine biosynthesis in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	23	12	16	8	59
Positive class 2	279	114	290	20	703
Positive class 3	1,185	359	768	25	2,337
Positive class 4	7,503	5,107	7,051	23,815	43,476
Total	8,990	5,592	8,125	23,868	46,575

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.47.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	22	10	12	5	49
Positive class 2	271	103	270	11	655
Positive class 3	1,194	372	792	37	2,395
Positive class 4	7,503	5,107	7,051	23,815	43,476
Total	8,990	5,592	8,125	23,868	46,575

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.47.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	23	12	16	8	59
Positive class 2	274	111	289	29	703
Positive class 3	1,146	346	764	81	2,337
Positive class 4	5,323	4,140	4,378	29,635	43,476
Total	6,766	4,609	5,447	29,753	46,575

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.34.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	22	10	12	5	49
Positive class 2	266	100	269	20	655
Positive class 3	1,155	359	788	93	2,395
Positive class 4	5,323	4,140	4,378	29,635	43,476
Total	6,766	4,609	5,447	29,753	46,575

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.34.

Table A.3: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Streptomycin biosynthesis in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	4	0	3	9
Positive class 2	0	6	0	12	18
Positive class 3	0	8	1	6	15
Positive class 4	173	651	83	1,652	2,559
Total	175	669	84	1,673	2,601

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.36.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	4	0	3	9
Positive class 2	0	6	0	12	18
Positive class 3	0	8	1	6	15
Positive class 4	173	651	83	1,652	2,559
Total	175	669	84	1,673	2,601

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.36.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	4	0	3	9
Positive class 2	0	6	0	12	18
Positive class 3	0	8	1	6	15
Positive class 4	70	413	27	2,049	2,559
Total	72	431	28	2,070	2,601

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.21.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	4	0	3	9
Positive class 2	0	6	0	12	18
Positive class 3	0	8	1	6	15
Positive class 4	70	413	27	2,049	2,559
Total	72	431	28	2,070	2,601

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.21.

Table A.4: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Methane metabolism in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	1	9	5	18
Positive class 2	1	6	5	24	36
Positive class 3	5	2	0	6	13
Positive class 4	176	205	354	2,276	3,011
Total	185	214	368	2,311	3,078

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.26.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	6	3	12
Positive class 2	0	5	8	8	21
Positive class 3	5	0	2	3	10
Positive class 4	178	208	352	2,297	3,035
Total	185	214	368	2,311	3,078

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.25.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	4	11	18
Positive class 2	1	4	3	28	36
Positive class 3	5	1	0	7	13
Positive class 4	83	100	114	2,714	3,011
Total	91	106	121	2,760	3,078

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.12.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	4	5	12
Positive class 2	0	4	4	13	21
Positive class 3	4	0	1	5	10
Positive class 4	85	101	112	2,737	3,035
Total	91	106	121	2,760	3,078

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.11.

Table A.5: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our$^{\#}$ result I (c-d) using four models for predicting Nicotinate and nicotinamide metabolism in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	12	0	6	16	34
Positive class 2	65	10	29	159	263
Positive class 3	114	17	52	394	577
Positive class 4	1,193	71	1,050	7,018	9,332
Total	1,384	98	1,137	7,587	10,206

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.31.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	12	0	6	16	34
Positive class 2	65	10	29	159	263
Positive class 3	114	17	52	394	577
Positive class 4	1,193	71	1,050	7,018	9,332
Total	1,384	98	1,137	7,587	10,206

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.31.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	12	0	6	16	34
Positive class 2	62	10	29	162	263
Positive class 3	109	17	52	399	577
Positive class 4	307	8	233	8,784	9,332
Total	490	35	320	9,361	10,206

c) Predicted outcomes from our$^{\#}$ result I and the known targets defined by metabolite transformation network. Confusion value= 0.13.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	12	0	6	16	34
Positive class 2	62	10	29	162	263
Positive class 3	109	17	52	399	577
Positive class 4	307	8	233	8,784	9,332
Total	490	35	320	9,361	10,206

d) Predicted outcomes from our$^{\#}$ result I and the known targets obtained from an original reference map. Confusion value= 0.13.

Table A.6: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our$^{\#}$ result I (c-d) using four models for predicting Phospholipid biosynthesis in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	18	1	2	0	21
Positive class 2	40	13	1	0	54
Positive class 3	36	7	6	2	51
Positive class 4	394	118	107	1,055	1,674
Total	488	139	116	1,057	1,800

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.40.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	18	1	2	0	21
Positive class 2	40	13	1	0	54
Positive class 3	36	7	6	2	51
Positive class 4	394	118	107	1,055	1,674
Total	488	139	116	1,057	1,800

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.40.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	18	1	2	0	21
Positive class 2	39	13	1	1	54
Positive class 3	35	7	6	3	51
Positive class 4	260	97	59	1,258	1,674
Total	352	118	68	1,262	1,800

c) Predicted outcomes from our$^{\#}$ result I and the known targets defined by metabolite transformation network. Confusion value= 0.28.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	18	1	2	0	21
Positive class 2	39	13	1	1	54
Positive class 3	35	7	6	3	51
Positive class 4	260	97	59	1,258	1,674
Total	352	118	68	1,262	1,800

d) Predicted outcomes from our$^{\#}$ result I and the known targets obtained from an original reference map. Confusion value= 0.28.

Table A.7: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Pyruvate oxidation pathway in the 7-pathway examples (section 4.2.3)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	0	1	2	5
Positive class 2	4	2	0	2	8
Positive class 3	3	0	2	0	5
Positive class 4	74	7	35	271	387
Total	83	9	38	275	405

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.32.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	0	1	2	5
Positive class 2	4	2	0	2	8
Positive class 3	3	0	2	0	5
Positive class 4	74	7	35	271	387
Total	83	9	38	275	405

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.32.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	0	1	2	5
Positive class 2	3	2	0	3	8
Positive class 3	3	0	1	1	5
Positive class 4	29	4	14	340	387
Total	37	6	16	346	405

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.15.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	0	1	2	5
Positive class 2	3	2	0	3	8
Positive class 3	3	0	1	1	5
Positive class 4	29	4	14	340	387
Total	37	6	16	346	405

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.15.

Table A.8: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Fluorobenzoate degradation in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	1	1	0	3
Positive class 2	1	1	0	0	2
Positive class 3	0	0	1	0	1
Positive class 4	2	5	4	1	12
Total	4	7	6	1	18

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.78.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	1	1	0	3
Positive class 2	1	1	0	0	2
Positive class 3	0	0	1	0	1
Positive class 4	2	5	4	1	12
Total	4	7	6	1	18

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.78.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	1	1	0	3
Positive class 2	1	1	0	0	2
Positive class 3	0	0	1	0	1
Positive class 4	0	0	0	12	12
Total	2	2	2	12	18

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.17.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	1	1	0	3
Positive class 2	1	1	0	0	2
Positive class 3	0	0	1	0	1
Positive class 4	0	0	0	12	12
Total	2	2	2	12	18

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.17.

Table A.9: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our$^{\#}$ result I (c-d) using four models for predicting Novobiocin biosynthesis in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	2	0	3
Positive class 2	1	1	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	26	3	79	174	282
Total	29	4	81	174	288

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.39.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	2	0	3
Positive class 2	1	1	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	26	3	79	174	282
Total	29	4	81	174	288

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.39.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	2	0	3
Positive class 2	1	1	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	19	1	42	220	282
Total	22	2	44	220	288

c) Predicted outcomes from our$^{\#}$ result I and the known targets defined by metabolite transformation network. Confusion value= 0.23.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	2	0	3
Positive class 2	1	1	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	19	1	42	220	282
Total	22	2	44	220	288

d) Predicted outcomes from our$^{\#}$ result I and the known targets obtained from an original reference map. Confusion value= 0.23.

Table A.10: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our$^{\#}$ result I (c-d) using four models for predicting Phosphonate and phosphinate metabolism in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	0	1
Positive class 2	0	0	0	0	0
Positive class 3	0	0	0	0	0
Positive class 4	5	0	2	10	17
Total	6	0	2	10	18

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.39.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	0	1
Positive class 2	0	0	0	0	0
Positive class 3	0	0	0	0	0
Positive class 4	5	0	2	10	17
Total	6	0	2	10	18

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.39.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	0	1
Positive class 2	0	0	0	0	0
Positive class 3	0	0	0	0	0
Positive class 4	1	0	0	16	17
Total	2	0	0	16	18

c) Predicted outcomes from our$^{\#}$ result I and the known targets defined by metabolite transformation network. Confusion value= 0.06.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	0	1
Positive class 2	0	0	0	0	0
Positive class 3	0	0	0	0	0
Positive class 4	1	0	0	16	17
Total	2	0	0	16	18

d) Predicted outcomes from our$^{\#}$ result I and the known targets obtained from an original reference map. Confusion value= 0.06.

Table A.11: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Naphthalene degradation in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	2	0	5
Positive class 2	0	0	2	0	2
Positive class 3	0	1	0	0	1
Positive class 4	251	40	306	121	718
Total	253	42	310	121	726

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.83.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	2	0	5
Positive class 2	0	0	2	0	2
Positive class 3	0	1	0	0	1
Positive class 4	251	40	306	121	718
Total	253	42	310	121	726

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.83.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	2	0	5
Positive class 2	0	0	2	0	2
Positive class 3	0	1	0	0	1
Positive class 4	28	8	49	633	718
Total	30	10	53	633	726

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.13.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	2	1	2	0	5
Positive class 2	0	0	2	0	2
Positive class 3	0	1	0	0	1
Positive class 4	28	8	49	633	718
Total	30	10	53	633	726

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.13.

Table A.12: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our# result I (c-d) using four models for predicting Nitrotoluene degradation in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	3	4
Positive class 2	2	0	0	0	2
Positive class 3	0	1	0	0	1
Positive class 4	31	52	11	449	543
Total	34	53	11	452	550

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.18.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	3	4
Positive class 2	2	0	0	0	2
Positive class 3	0	1	0	0	1
Positive class 4	31	52	11	449	543
Total	34	53	11	452	550

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.18.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	3	4
Positive class 2	2	0	0	0	2
Positive class 3	0	1	0	0	1
Positive class 4	6	23	3	511	543
Total	9	24	3	514	550

c) Predicted outcomes from our# result I and the known targets defined by metabolite transformation network. Confusion value= 0.07.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	1	0	0	3	4
Positive class 2	2	0	0	0	2
Positive class 3	0	1	0	0	1
Positive class 4	6	23	3	511	543
Total	9	24	3	514	550

d) Predicted outcomes from our# result I and the known targets obtained from an original reference map. Confusion value= 0.07.

Table A.13: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our$^{\#}$ result I (c-d) using four models for predicting Caprolactam degradation in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	0	0	3
Positive class 2	2	0	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	53	14	3	50	120
Total	59	14	3	50	126

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.58.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	0	0	3
Positive class 2	2	0	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	53	14	3	50	120
Total	59	14	3	50	126

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.58.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	0	0	3
Positive class 2	2	0	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	9	10	0	101	120
Total	15	10	0	101	126

c) Predicted outcomes from our$^{\#}$ result I and the known targets defined by metabolite transformation network. Confusion value= 0.17.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	0	0	3
Positive class 2	2	0	0	0	2
Positive class 3	1	0	0	0	1
Positive class 4	9	10	0	101	120
Total	15	10	0	101	126

d) Predicted outcomes from our$^{\#}$ result I and the known targets obtained from an original reference map. Confusion value= 0.17.

Table A.14: The 4×4 confusion matrix of four positive classes yielded by our result I (a-b) and our$^{\#}$ result I (c-d) using four models for predicting Biosynthesis of siderophore group nonribosomal peptides in the additional 7 other pathways (table A.15)

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	1	1	5
Positive class 2	2	2	3	1	8
Positive class 3	2	0	3	0	5
Positive class 4	17	6	56	191	270
Total	24	8	63	193	288

a) Predicted outcomes from our result I and the known targets defined by metabolite transformation network. Confusion value= 0.31.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	1	1	5
Positive class 2	2	2	3	1	8
Positive class 3	2	0	3	0	5
Positive class 4	17	6	56	191	270
Total	24	8	63	193	288

b) Predicted outcomes from our result I and the known targets obtained from an original reference map. Confusion value= 0.31.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	1	1	5
Positive class 2	2	2	3	1	8
Positive class 3	2	0	3	0	5
Positive class 4	4	0	14	252	270
Total	11	2	21	254	288

c) Predicted outcomes from our$^{\#}$ result I and the known targets defined by metabolite transformation network. Confusion value= 0.10.

Predicted outcomes	1	2	3	4	Total of each class
Positive class 1	3	0	1	1	5
Positive class 2	2	2	3	1	8
Positive class 3	2	0	3	0	5
Positive class 4	4	0	14	252	270
Total	11	2	21	254	288

d) Predicted outcomes from our$^{\#}$ result I and the known targets obtained from an original reference map. Confusion value= 0.10.

Table A.15: The additional 7 other pathways from KEGG Pathway database as the same version as trained data (section 3.4)

Reference i.d.	Name	Pathway types/roles
eco00364	Fluorobenzoate degradation	Xenobiotics biodegradation and metabolism
eco00401	Novobiocin biosynthesis	Biosynthesis of other secondary metabolites
eco00440	Phosphonate and phosphinate metabolism	Metabolism of other amino acids
eco00626	Naphthalene degradation	Xenobiotics biodegradation and metabolism
eco00633	Nitrotoluene degradation	Xenobiotics biodegradation and metabolism
eco00930	Caprolactam degradation	Xenobiotics biodegradation and metabolism
eco01053	Biosynthesis of siderophore group nonribosomal peptides	Metabolism of terpenoids and polyketides

Biography

Sasiporn Tongman was born on November, 1983, in Prachinburi, Thailand. She received bachelor degree of science (honours) in biochemistry, from faculty of science, Chulalongkorn university, Thailand, in 2005. Her bachelor degree was supervised by Asst.Prof.Dr. Suchart Chanama, Ph.D. Her philosophy degree has been under the supervision of Prof.Dr. Chidchanok Lursinsap, Ph.D. and the co-supervision of Asst.Prof.Dr.Suchart Chanama, Ph.D. She was granted a Ph.D. scholarship from the strategic scholarships fellowships frontier research networks by higher education commission, Thailand, in 2006. Her interested research fields are applying *in silico* methods, especially machine learning approaches, to scientific problems in system biology, computational biology, and bioinformatics.